NASA/CR—2004-212958

LSPRAY–II: A Lagrangian Spray Module

M.S. Raju
QSS Group, Inc., Cleveland, Ohio

March 2004

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at 301–621–0134

- Telephone the NASA Access Help Desk at 301–621–0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076

NASA/CR—2004-212958

LSPRAY–II: A Lagrangian Spray Module

M.S. Raju
QSS Group, Inc., Cleveland, Ohio

Prepared under Contract NAS3–00145

National Aeronautics and
Space Administration

Glenn Research Center

March 2004

Acknowledgments

The research funding for this work was provided by NASA Glenn Research Center with Dr. N.-S. Liu acting as the technical monitor.

Available from

NASA Center for Aerospace Information
7121 Standard Drive
Hanover, MD 21076

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22100

Available electronically at http://gltrs.grc.nasa.gov

LSPRAY–II: A Lagrangian Spray Module

M.S. Raju
QSS Group, Inc.
Cleveland, Ohio 44135

1 ABSTRACT

LSPRAY-II is a Lagrangian spray solver developed for application with parallel computing and unstructured grids. It is designed to be massively parallel and could easily be coupled with any existing gas-phase flow and/or Monte Carlo Probability Density Function (PDF) solvers. The solver accommodates the use of an unstructured mesh with mixed elements of either triangular, quadrilateral, and/or tetrahedral type for the gas flow grid representation. It is mainly designed to predict the flow, thermal and transport properties of a rapidly vaporizing spray because of its importance in aerospace application. The manual provides the user with an understanding of various models involved in the spray formulation, its code structure and solution algorithm, and various other issues related to parallelization and its coupling with other solvers. With the development of LSPRAY-II, we have advanced the state-of-the-art in spray computations in several important ways. Some of the major features of the spray module are:

- It facilitates the use of both unstructured grids and parallel computing and, thereby, facilitates large-scale combustor computations involving complex geometrical configurations.
- In order to deal with modern gas-turbine fuels that are mixtures of many compounds, we have extended the spray formulation to the modeling of multicomponent liquid fuels.
- In order to extend the applicability of the spray computations over a wide range of low-pressure conditions, we have completed the implementation of the liquid-phase variable thermo-transport properties into our spray formulation.
- The spray module is designed in such a way so that it could easily be coupled with other CFD codes.
- It can be used in the calculation of both steady as well as unsteady computations.
- We have developed and implemented a time-averaging method into the calculation of the liquid-phase source contributions in order to accelerate convergence towards a steady state solution.
- The spray module has a multi-liquid and multi-injection capability.

With the aim of improving the overall solution procedure involving sprays, we have made several relevant contributions to the gas side of the computations:

- In order to demonstrate the importance of chemistry/turbulence interactions in the modeling of reacting sprays, we have extended the joint scalar Monte Carlo PDF (Probability Density Function) approach to the modeling of spray flames.
- In order to account for the nonideal gas behavior under critical and supercritical conditions, we have completed the integration of a high-pressure EOS (Equation-of-State) into our CFD module and, also, the implementation of a high-pressure correction into the calculation of the gas-phase transport properties.

This modeling approach provided favorable results when applied to several different spray flames representative of those encountered in both gas-turbine combustors as well as stratified-charge rotary combustion (Wankel) engines.

2 NOMENCLATURE

$\underline{a}_n$	outward area normal vector of the nth surface, m^2
B_k	Spalding transfer number
C_D	drag coefficient
C_p	specific heat, J/(kg K)
D	turbulent diffusion coefficient, m^2/s
d	droplet diameter, m
h	specific enthalpy, J/kg
k	turbulence kinetic energy, m^2/s^2
k_{ij}	binary interaction parameter
l_k	mixture latent heat of evaporation, J/kg
$l_{k,eff}$	effective latent heat of evaporation, J/kg (defined in Eq. (35))
l_{kin}	heat of vaporization at normal boiling point, J/kg
M_i	molecular weight of ith species, kg/kg-mole
m_k	droplet vaporization rate, kg/s
m_{ko}	initial mass flow rate associated with kth droplet group, kg/s
N_f	number of surfaces contained in a given computational cell
N_p	total number of computational cells
n_k	number of droplets in kth group
P	pressure, N/m^2
P_c	critical pressure, N/m^2
P_r	Prandtl number
R_u	gas constant, J/(kg K)
Re	Reynolds number
r_k	droplet radius, m
r_{ko}	initial droplet radial location, m
s_k	droplet radius-squared ($= r_k^2$), m^2
s_{mlc}	liquid source contribution of the gas-phase continuity equation
s_{mle}	liquid source contribution of the gas-phase energy equation
s_{mlm}	liquid source contribution of the gas-phase momentum equations
s_{mls}	liquid source contribution of the gas-phase species equations
T	temperature, K
T_{nb}	normal boiling point, K
T_c	critical temperature, K
t	time, s
u_i	ith velocity component, m/s
u_{ik}	ith velocity component of kth droplet group, m/s
V_c	volume of the computational cell, m^3 or critical molar volume, m^3/kg-mole
V_n	molar volume at normal pressure, m^3/kg-mole
$\dot{w}_j$	gas-phase chemical reaction rate, 1/s
x_i	Cartesian coordinate in the ith direction, m
y_j	mass fraction of jth species
$\underline{x}$	spatial vector
α	represents a coordinate related to a Hill's Vortex or a spray cone angle, deg.
β	a spray cone angle, deg.
χ	mole fraction
Δt_f	local time step used in the flow solver, s
Δt_{gl}	global time step used in the spray solver, s
Δt_{il}	fuel injection time step, s
Δt_{ml}	allowable time step in the spray solver, s
ΔV	computational cell volume, m^3
δ	Dirac-delta function
ϵ	rate of turbulence dissipation, m^2/s^3
ϵ_j	fractional mass evaporating rate of species at the droplet surface
Γ_ϕ	turbulent diffusion coefficient, kg/ms
λ	thermal conductivity, J/(ms K)
μ	dynamic viscosity, kg/ms
ω	turbulence frequency, 1/s or acentric factor in Table 1
ρ	density, kg/m^3
ρ_{ln}	liquid density (at 1 bar, 273.15 K), kg/m^3
σ	characteristic diameter of a molecule in Table 1
τ	stress tensor term, kg/ms^2
θ	void fraction or a spray cone angle, deg.

Subscripts

f	represents conditions associated with fuel
g	global or gas-phase
i	index for the coordinate or species components
j	index for the species component
k	droplet group or liquid-phase conditions
l	liquid-phase
n	nth-face of the computational cell
o	initial conditions or oxidizer
p	conditions associated with the properties of a grid cell
s	represents conditions at the droplet surface or adjacent computational cell
t	conditions associated with time
,	partial differentiation with respect to the variable followed by it

Superscripts

$''$	fluctuations

3 INTRODUCTION

There are many occurrences of sprays in a variety of industrial and power applications and materials processing [1]. A liquid spray is a two phase flow with the gas as the continuous phase and the liquid as the dispersed phase in the form of droplets or ligaments [1]. The interaction between the two phases, which are coupled through exchanges of mass, momentum, and energy, can occur in different ways at disparate time and length scales involving various thermal, mass, and fluid dynamic factors.

A number of finite-difference formulations have been advanced over the years for predicting the flow (mass and momentum) and thermal properties of a rapidly vaporizing spray. Some of the pros and cons of various formulations can be found in [1-5]. Depending on the nature of the spray, an appropriate selection could be made from the choice of various well-known spray formulations (multicontinua, discrete-particle, or probabilistic) based on either a Lagrangian or an Eulerian representation for the liquid-phase equations by incorporating appropriately chosen droplet sub-grid models. In this manual, we only summarize the salient aspects of the spray formulation adopted from our previous work [6-15] without attempting to provide an in-depth review on the transport and fluid dynamic behavior of reacting sprays. The present solution procedure could be used within the context of both the multicontinua and probabilistic spray formulations, and it allows for resolution on a scale greater than the average spacing between two neighboring droplets [1]. An Eulerian scheme is assumed for the gas phase equations as it is an adopted choice for NCC (National Combustion Code). The liquid-phase equations form a system of hyperbolic equations and they could be solved by means of either an Eulerian or a Lagrangian representation. A Lagrangian scheme is chosen as it reduces the errors associated with numerical diffusion. The droplet sub-grid models are based on well established models for droplet drag; the vaporization models of a polydisperse spray take into account the transient effects associated with the droplet internal heating and the forced convection effects associated with droplet internal circulation; and it employs models for gas-film valid over a wide range of low to intermediate droplet Reynolds numbers [7]. The present formulation is based on a stochastic particle tracking method applicable for flows with a dilute spray approximation where the droplet loading is low. The numerical method could be used within the context of both steady and unsteady calculations [8-13]. Not considered in the present release of the code are the effects associated with the droplet breakup, droplet/shock interaction, and dense spray effects.

In order to facilitate large-scale combustor computations, we have extended our previous work on sprays to parallel computing [10-14]. But it is also well known that considerable effort usually goes into generating computational grids of practical combustor geometries. In order to allow representation of complex geometries with relative ease, we have extended our previous work on sprays to unstructured meshes [11-14] as it is well known that the grid generation time could be improved considerably with the use of widely available grid generation packages such as GRIDGEN. With the aim of advancing the current multi-dimensional computational tools used in the design of advanced technology combustors, two new computer codes, LSPRAY - a Lagrangian spray solver [14] and EUPDF - an Eulerian Monte Carlo PDF solver [15], were developed, thereby extending our previous work on the Monte Carlo PDF and sprays [10] to unstructured grids as a part of NCC development. The combined unstructured 3D CFD/Spray/Monte-Carlo-PDF solver is designed to be massively parallel and accommodates the use of an unstructured mesh with mixed elements comprised of either triangular, quadrilateral, and/or tetrahedral type.

A current status of the use of the parallel computing in turbulent reacting flows involving sprays, scalar Monte Carlo PDF, and unstructured grids was described in Ref. [11]. It outlined several numerical techniques developed for overcoming some of the high computer CPU-time and memory-storage requirements associated with the use of Monte Carlo solution methods. The parallel performance of both the PDF and CFD modules was found to be excellent but the results were mixed for the spray computations showing reasonable performance on massively parallel computers like Cray T3D; but its performance was poor on the workstation clusters [10]. In order to improve the parallel performance of the spray module, two different domain decomposition strategies were developed and the results from both strategies were summarized [10-14].

Currently, most of the finite-difference techniques used for predicting the two-phase flows make use of the physics derived from the single-component liquid droplet studies with constant properties. How-

ever, it is well known that most of the gas-turbine fuels are multicomponent mixtures of many compounds with a wide distillation curve [7,18]. The multicomponent nature of the liquid sprays is becoming evident with the increasing need to use jet fuels derived from heavier petroleum compounds. The gasification behavior of a multicomponent fuel droplet may differ significantly over that of a pure single component fuel droplet [18]. Also, the calculation of the variable thermo-transport properties of the liquid-mixtures becomes more important at high pressures. The flame ignition characteristics such as the flame blow-off and extinction conditions could also be influenced by the nonuniform concentration of the fuels with different volatilities. However, the importance of the multicomponent liquid fuels with variable properties received little attention in the modeling of comprehensive gas-turbine combustor spray computations. With this in mind, we have made the following improvements:

(a) In order to deal with the gas turbine fuels that are mixtures of many compounds, we have extended the spray formulation to multicomponent liquid sprays. This implementation also takes into account the effect of variable liquid properties.

(b) In order to account for the nonideal gas behavior under critical and supercritical conditions, we have completed the integration of the Peng-Robinson EOS into our CFD module and also, the implementation of a high-pressure correction into the calculation of the transport properties in the gas phase. These modifications would enable the calculation of high-pressure flow properties in the gas-phase regardless of sprays.

In this manual, we concentrate only on providing the details of the spray module along with the high pressure corrections made to the gas-phase calculations. However, a discussion on the application of the joint scalar Monte Carlo PDF method to two-phase flows could be found elsewhere in [10-15]. Some of the salient features of the spray module are summarized below:

- An efficient particle tracking algorithm was developed and implemented into the Lagrangian spray solver in order to facilitate particle movement in an unstructured grid of mixed elements.
- LSPRAY-II is currently coupled with an unstructured flow (CFD) solver of NCC [21-23], and an Eulerian-based Monte Carlo probability density function solver - EUPDF [15], which were selected to be as the integral components of the NCC cluster of modules. EUPDF was developed for application with sprays, combustion, unstructured grids and parallel computing.
- The spray solver receives the mean velocity and turbulence fields from the flow solver. The solution for the scalar (energy and species) fields could be provided by means of either a conventional CFD solver or a Monte Carlo PDF solver depending on the choice of the solver.
- The spray solver supplies the spray source-term contributions arising from the exchanges of mass, momentum and energy with the liquid-phase to the flow solvers (CFD and/or Monte Carlo PDF). This information could be used in either conservative or non-conservative finite-difference formulations of the gas phase equations.

The furnished code demonstrates the successful methods used for parallelization and coupling of the spray to the flow code. First, complete details of the spray solution procedure is presented along with several other numerical issues related to the coupling between the CFD, LSPRAY-II, and EUPDF solvers. It is followed by a brief description of the combined parallel performance of the three solvers (CFD, EUPDF, and LSPRAY-II) along with a brief summary of the validation cases.

4 GOVERNING EQUATIONS FOR THE GAS PHASE

Here, we summarize the conservation equations for the gas-phase in Eulerian coordinates [1]. These equations are valid for a dilute spray with a void fraction of the gas, θ, close to unity. The void fraction is defined as the ratio of the equivalent volume of gas to a given volume of a gas and liquid mixture. This is done for the purpose of identifying the interphase source terms arising from the exchanges of mass, momentum, and energy with the liquid-phase.

The conservation of the mass leads to:

$$[\bar{\rho} V_c]_{,t} + [\bar{\rho} V_c u_i]_{,x_i} = s_{mlc} = \sum_k n_k \, m_k \qquad (1)$$

For mass conservation, the source term is given as a summation over different classes of droplets. Each class represents the average properties of a different polydisperse group of droplets. Here, n_k represents

the number of droplets in a given class and m_k represents the corresponding mass vaporization rate.

For the conservation of the jth species, we have:

$$[\bar{\rho}V_c y_j]_{,t} + [\bar{\rho}V_c u_i y_j]_{,x_i} - [\bar{\rho}V_c D y_{j,x_i}]_{,x_i} - \bar{\rho}V_c \dot{w}_j =$$

$$s_{mls} = \sum_k \epsilon_j \, n_k \, m_k \qquad (2)$$

where

$$\sum_j \dot{w}_j = 0 \; and \; \sum_j \epsilon_j = 1$$

For the species conservation, the source term contains an additional variable, ϵ_j, which is defined as the fractional vaporization rate for species j.

For the momentum conservation, we have:

$$[\bar{\rho}V_c u_i]_{,t} + [\bar{\rho}V_c u_i u_j]_{,x_j} + [pV_c]_{,x_i} -$$

$$[\theta V_c \tau_{ij}]_{,x_j} - [(1-\theta)V_c \tau_{lij}]_{,x_j} = s_{mlm} =$$

$$\sum_k n_k \, m_k \, u_{ki} - \sum_k \frac{4\pi}{3} \rho_k \, r_k^3 \, n_k \, u_{ki,t} \qquad (3)$$

where the shear stress τ_{ij} in Eq. (3) is given by:

$$\tau_{ij} = \mu[u_{i,x_j} + u_{j,x_i}] - \frac{2}{3}\mu\delta_{ij}u_{i,x_j}$$

For the momentum conservation, the first source term represents the momentum associated with liquid fuel vapor and the second represents the momentum change associated with droplet drag.

For the energy conservation, we have:

$$[\bar{\rho}V_c h]_{,t} + [\bar{\rho}V_c u_i h]_{,x_i} - [\theta V_c \lambda T_{,x_i}]_{,x_i} - [(1-\theta)V_c \lambda_l T_{,x_i}]_{,x_i}$$

$$-[\theta V_c p]_{,t} = s_{mle} = \sum_k n_k \, m_k \Big(h_s - l_{k,eff}\Big) \qquad (4)$$

Similarly, the energy conservation has a first source term associated with liquid fuel vapor and the second represents the heat loss associated with the latent heat of vaporization and it also contains an additional heat flux (loss or gain) to the droplet interior from the ambient.

The main purpose of the spray solver is to calculate the source terms arising from the exchanges of the mass, momentum, and energy and, then, feed that appropriate information to the gas-phase solver. In the case of NCC, it suplies the source terms to the CFD solver and, also, to the Monte Carlo PDF solver if needed.

The polynomial fits for the variable thermodynamic properties at low pressures are taken from the data set compiled by McBride et. el. [24]. The transport properties involving the thermal conductivity and molecular viscosity for individual species is estimated based on the Chapman-Enskog collision theory [25-27]. And Wilke's formulae is used to determine the properties of mixture [25-27]. The binary-diffusion coefficients are determined based on the Chapman-Enskog theory and the Lennard-Jones potential [25-27].

5 HIGH PRESSURE EQUATION OF STATE

In order to calculate the high pressure gas behavior, the Peng-Robinson EOS (Equation-Of-State) is employed for a multi-component mixture in the following form [16-17,28]:

$$P = \frac{RT}{V - b_m} - \frac{a_m}{V^2 + 2b_m V - b_m^2} \qquad (5)$$

where

$a_m = \sum_i \sum_j y_i y_j (a_i a_j)^{1/2}(1 - k_{ij})$,

$b_m = \sum_i y_i b_i$,

$b_i = \frac{0.07780 R T_{ic}}{P_{ic}}$,

$a_i = \frac{0.45724 R^2 T_{ic}^2}{P_{ic}}[1 + f_{i\omega}(1 - T_{ir}^{1/2})]^2$,

$T_{ir} = T/T_{ic}$,

$f_{i\omega} = 0.37464 + 1.54226\omega_i - 0.26992{\omega_i}^2$,

ω_i is known as the acentric factor of the molecules which is a measure of non-sphericity of the molecules, and k_{ij} is known as the binary interaction coefficient. However, the Peng-Robinson EOS is rewritten as a cubic EOS in terms of the compressibility factor, Z ($= \frac{PV}{RT}$), before it is solved:

$$Z^3 - (1 - B^\star)Z^2 + (A^\star - 2B^\star - 3B^{\star 2})Z$$

$$-A^\star B^\star + B^{\star 2} + B^{\star 3} = 0 \qquad (6)$$

where

$A^\star = \frac{a_m P}{R^2 T^2}$, and $B^\star = \frac{b_m P}{RT}$.

We have chosen the Peng-Robinson EOS because of its simplicity and, more importantly, it proved to be very useful in the supercritical droplet vaporization studies of [19-20].

Table 1 lists various physical constants for some of the species that we found to be of interest in our spray computations. It contains values for the boiling temperature at normal pressure, the critical temperature, the critical pressure, the critical density, the latent heat of vaporization at normal pressure, the critical molar volume, the molar volume at normal pressure, and the acentric factor of the molecules. Most of this data is collected from [16-18] and is useful in both the evaluation of the Peng-Robinson EOS and the calculation of various other variable properties. And Table 2 lists the binary parameters, k_{ij}, for a mixture of n-heptane, O_2, N_2, CO_2, & H_2O used in a calculation involving the Peng-Robinson EOS. While the data for most of the binary pairs was obtained from various reference books, some of the missing data, however, is replaced with known data found for other binary pairs of molecules with similar molecular weights.

6 HIGH PRESSURE CORRECTIONS FOR GAS-PHASE TRANSPORT PROPERTIES

The effect of pressure on the viscosity of pure gases is determined by making use of the Reichenberg method [16,19,29]:

$$\frac{\mu}{\mu_n} = 1 + Q_v \frac{A_v P_r^{3/2}}{B_v P_r + (1 + C_v P_r^{D_v})^{-1}} \quad (7)$$

where

$$A_v = \frac{\alpha_1}{T_r} exp(\alpha_2 T_r^a) \quad B_v = A_v(\beta_1 T_r - \beta_2)$$
$$C_v = \frac{\gamma_1}{T_r} exp(\gamma_1 T_r^c) \quad D_v = \frac{\lambda_1}{T_r} exp(\lambda_2 T_r^d)$$

$$\alpha_1 = 1.982510^{-03} \quad \alpha_2 = 5.2683 \quad a = -0.5767$$
$$\beta_1 = 1.6552 \quad \beta_2 = 1.2760$$
$$\gamma_1 = 0.1319 \quad \gamma_2 = 3.7035 \quad c = -79.8678$$
$$\lambda_1 = 2.9496 \quad \lambda_2 = 2.9190 \quad d = -16.6169$$

and $Q_v = 1.0$ for non-polar molecules.

The effect of pressure on the thermal conductivity of pure gases is determined by making use of the Stiel & Thodos method [16,19,30]:

$$(\lambda - \lambda_n)\Gamma Z_c^5 = 1.2210^{-2}[exp(0.535\rho_r) - 1] \quad (8)$$

when $\rho_r < 0.5$

$$(\lambda - \lambda_n)\Gamma Z_c^5 = 1.1410^{-2}[exp(0.67\rho_r) - 1.069] \quad (9)$$

when $0.5 < \rho_r < 2.0$

$$(\lambda - \lambda_n)\Gamma Z_c^5 = 2.6010^{-2}[exp(1.155\rho_r) + 2.016] \quad (10)$$

when $2.0 < \rho_r < 2.8$

where λ is in W/(m.K), $\Gamma = 210\left[\frac{T_c M^3}{P_c^4}\right]^{1/6}$, Z_c is the critical compressibility, and ρ_r is the reduced density $\rho/\rho_c = V_c/V$.

The effect of pressure & temperature on diffusion coefficients in gases is determined by making use of Takahashi correlation [16,19,31]:

$$\frac{D_{AB}P}{(D_{AB}P)^+} = f(T_r, P_r) \quad (11)$$

where D_{AB} = diffusion coefficient, cm^2/s, P = pressure, bar, the superscript $^+$ indicates that low-pressure values are to be used, and the reduced pressure and temperature in the above equation are calculated as follows:

$$T_r = \frac{T}{T_c}$$
$$T_c = y_A T_{cA} + y_B T_{cB}$$
$$P_r = \frac{P}{P_c}$$
$$P_c = y_A P_{cA} + y_B P_{cB}$$

This correlation is valid for a system involving a trace solute diffusing in a supercritical fluid. It is shown to provide satisfactory results but it is based on a very limited set of available experimental data. The graphical representation of the Takahashi function is given in Fig. 1.

Table 1. Physical constants.

Species	T_{nb} (K)	T_c (K)	P_c (atm)	ρ_c (Kg/m^3)	l_{kin} (KJ/kg)	V_c (cm^3/g-mole)	V_n (cm^3/g-mole)	ω	σ (A ^{o})	σ / κ (K)
C_6H_{14}	341.9	507.4	30.0	660.0	334.8	370.0	140.06	0.296	5.949	399.3
C_7H_{16}	371.6	540.2	27.0	682.0	316.3	432.0	162.00	0.351	6.297	419.031
C_8H_{18}	398.82	568.8	24.6	718.5	301.3	492.0	188.8	0.394	6.62	488.15
$C_{10}H_{22}$	447.3	617.6	20.8	728.3	276.1	603.0	233.68	0.490	7.16	540.06
$C_{12}H_{26}$	489.5	658.3	18.0	748.0	256.3	713.0	278.54	0.562	7.655	583.68
$C_{14}H_{30}$	526.7	694.0	16.0	763.0	240.1	830.0	326.62	0.679	8.067	629.08
N_2	77.4	126.2	33.9	807.1	197.6	90.1	31.87	0.039	3.681	91.5
O_2	90.2	154.6	50.4	1135.7	212.3	74.4	26.08	0.025	3.433	113.0
CO_2	00.0	304.1	73.8	000.0	000.0	94.0	33.32	0.239	3.996	190.0
H_2O	373.2	647.3	221.2	958.1	2257.2	57.1	19.76	0.344	2.641	809.1

Table 2. Binary parameter constants, $\bar{k}_{ij}$.

	C_7H_{16} (j=1)	O_2 (j=2)	N_2 (j=3)	CO_2 (j=4)	H_2O (j=5)
C_7H_{16} (i=1)	0.0000	0.1321	0.1440	0.1000	0.1484
O_2 (i=2)	0.1321	0.0000	-0.0119	-0.0289	0.0910
N_2 (i=3)	0.1440	-0.0119	0.0000	-0.0170	0.1030
CO_2 (i=4)	0.1000	-0.0289	-0.0170	0.0000	0.1200
H_2O (i=5)	0.1484	0.0910	0.1030	0.1200	0.0000

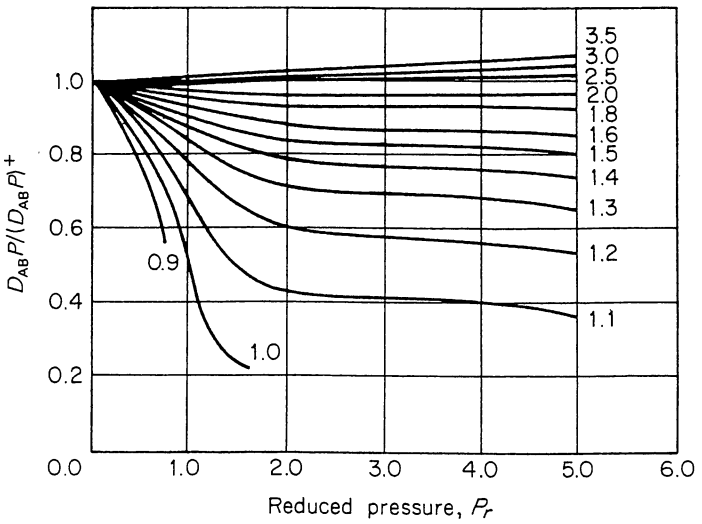


Figure 1 Takahashi correlation showing the variation of the binary diffusion coefficient versus reduced pressure at different reduced temperatures.

7 LIQUID-PHASE EQUATIONS

Here, we summarize the governing equations for the liquid-phase based on a Lagrangian formulation where the equations for particle position and velocity are described by a set of ordinary differential equations. For the particle position of the kth droplet group, we have:

$$\frac{dx_{ik}}{dt} = u_{ik} \tag{12}$$

For the droplet velocity:

$$\frac{du_{ik}}{dt} = \frac{3}{16}\frac{C_D \mu_{gs} Re_k}{\rho_k r_k^2}\left[u_{ig} + u'_g - u_{ik}\right] \tag{13}$$

where

$$C_D = \frac{24}{Re_k}\left(1 + \frac{Re_k^{2/3}}{6}\right) \tag{14}$$

According to Yuen and Chen [32], the droplet drag for a vaporizing droplet could be calculated by a solid-sphere drag correlation but suggested using a correction in the evaluation of the droplet Reynolds number for the average viscosity based on a 1/3 weighting rule as given by Eq. (46). The droplet Reynolds number is given by:

$$Re_k = 2\frac{r_k \rho_g}{\mu_{gs}}\left[\left(u_g + u'_g - u_k\right).\left(u_g + u'_g - u_k\right)\right]^{1/2} \tag{15}$$

By following the approach taken from KIVA [33], a gas turbulence velocity, u'_g, is added to the local mean gas velocity when calculating the droplet drag and vaporization rates. The gas velocity fluctuations is calculated by randomly sampling a Gaussian distribution with mean square deviation, $2/3k$. The Gaussian is given by

$$G(u'_g) = (4/3\pi k)^{-3/2} exp[-3|u'_g|^2/4k] \tag{16}$$

The gas fluctuating component is calculated once at every turbulence interaction time, t_{tur}, and is otherwise held constant [33]. The correlation time is taken to be the minimum of either the eddy time or the transit time taken by the droplet to traverse the eddy. It is given by

$$t_{tur} = min\left(\frac{k}{\epsilon}, c_{tt}\frac{k^{3/2}}{\epsilon}\frac{1}{|u_g + u'_g - u_k|}\right) \tag{17}$$

where c_{tt} is an empirical constant with a value of 0.16432.

The liquid mixture density, ρ_k, in Eq. (13) is calculated as follows [16,18]:

$$\rho_k = \sum_i y_{ki}\rho_{ki} \tag{18}$$

and the individual component liquid density is given by:

$$\rho_{ki} = \frac{M_{ki}}{V_{ki}} \tag{19}$$

where the molar volume, V_{ki}, is

$$V_{ki} = V_{ci}(0.29056 - 0.8775\omega_i)^{c_v} \tag{20}$$

and

$$c_v = \left[1 - \frac{T_k}{T_{ci}}\right]^{\frac{2}{7}}$$

The droplet regression rate is determined from one of three different correlations depending upon the

droplet Reynolds number range. The first correlation as given by Eq. (21) is based on a gas-film analysis developed by Tong & Sirignano [34]. It is based on a a combination of stagnation and flat-plate boundary-layer analysis and is valid for Reynolds numbers in the intermediate range. The last two correlations as given by Eqs. (22) & (23) are taken from Clift et al [35] and are valid when $Re_k \leq 20$. In fact, when the droplet Reynolds number goes to zero, Eq. (23) becomes identical to the droplet regression rate of a vaporizing droplet in a quiescent medium [27].

$$\frac{ds_k}{dt} = -2\frac{\rho_g D_g}{\rho_k}\left[\frac{2}{\pi}Re_k\right]^{1/2} f(B_k) \qquad (21)$$

$if\ Re_k > 20$

$$\frac{ds_k}{dt} = -\frac{\rho_g D_g}{\rho_k}\left[1+(1+Re_k)^{1/3}\right] Re_k^{0.077} ln(1+B_k) \qquad (22)$$

$if\ 1 < Re_k \leq 20$

$$\frac{ds_k}{dt} = -\frac{\rho_g D_g}{\rho_k}\left[1+(1+Re_k)^{1/3}\right] ln(1+B_k) \qquad (23)$$

$if\ Re_k < 1$

where B_k is the Spalding transfer number and is given by:

$$B_k = \frac{(y_{fs} - y_f)}{(1-y_{fs})} \qquad (24)$$

and y_{fs} is given as a summation over the fuel-species mass fractions at the droplet interface:

$$y_{fs} = \sum_i y_{is} \qquad (25)$$

and y_f is given as a summation over the fuel-species mass fractions in the ambient:

$$y_f = \sum_i y_{fi} \qquad (26)$$

and the function $f(B_k)$ is similar to that of a Blassius function [1, 36] and is obtained from an analysis similar to that of Emmon's boundary-layer flow over a flat plate with blowing [36]. The range of validity of this function was extended in Raju and Sirignano [7] to consider the effects associated with a boundary-layer flow with suction.

The internal droplet temperature is determined based on a vortex model [34]. The governing equation for the internal droplet temperature is given by:

$$\frac{\partial T_k}{\partial t} = 17\frac{\lambda_l}{C_{pl}\rho_l r_k^2}\left[\alpha\frac{\partial^2 T_k}{\partial \alpha^2} + (1+C(t)\alpha)\frac{\partial T_k}{\partial \alpha}\right] \qquad (27)$$

where

$$C(t) = \frac{3}{17}\left[\frac{C_{pl}\rho_l}{\lambda_l}\right] r_k \frac{dr_k}{dt} \qquad (28)$$

where α represents the coordinate normal to the stream-surface of a Hill's Vortex in the circulating fluid, and $C(t)$ represents a nondimensional form of the droplet regression rate. The initial and boundary conditions for Eq. (27) are given by:

$$t = t_{injection}, \quad T_k = T_{k,o} \qquad (29)$$

$$\alpha = 0, \quad \frac{\partial T_k}{\partial \alpha} = \frac{1}{17}\left[\frac{C_{pl}\rho_l}{\lambda_l}\right] r_k^2 \frac{\partial T_k}{\partial t} \qquad (30)$$

$$\alpha = 1, \quad \frac{\partial T_k}{\partial \alpha} = -\frac{3}{32}\frac{\rho_k}{\lambda_l}\left[l_{k,eff} - l_k\right]\frac{ds_k}{dt} \qquad (31)$$

where $\alpha = 0$ refers to the vortex center, and $\alpha = 1$ refers to the droplet surface, and the mixture latent heat of vaporization, l_k, is given by

$$l_k = \sum_i \epsilon_i l_{ki} \qquad (32)$$

and the individual component latent heat of vaporization, l_{ki}, is given by [16,18]:

$$l_{ki} = l_{kin}\left(\frac{T_{ci} - T_k}{T_{ci} - T_{bi}}\right)^{0.38} \qquad (33)$$

and the droplet boiling temperature is given by

$$T_{bi} = \frac{l_{kin} M_i / R_u}{l_{kin} M_i/(R_u t_{bni}) - ln(P)} \qquad (34)$$

and, finally, the effective latent heat of vaporization, $l_{k,eff}$, is defined as:

$$l_{k,eff} = l_k + 4\pi\frac{\lambda_l r_k^2}{m_k}\left(\frac{\partial T_k}{\partial r}\right)_s \qquad (35)$$

It is a very useful parameter as it represents the total energy loss associated with the latent heat of vaporization in addition to the the heat loss to the droplet

interior. $l_{k,eff}$ is calculated by means of the following relationship [18]:

$$l_{k,eff} = \frac{C_p(T_g - T_{ks})}{(1+B_k)^{1/Le} - 1} \tag{36}$$

Similar to the internal thermal transport, the internal mass transport of a multi-component fuel is given by:

$$\frac{\partial y_{ki}}{\partial t} = 17\frac{D_k}{r_k^2}\left[\alpha\frac{\partial^2 y_{ki}}{\partial \alpha^2} + (1 + C(t)\alpha)\frac{\partial y_{ki}}{\partial \alpha}\right] \tag{37}$$

The initial and boundary conditions for Eq. (37) are given by:

$$t = t_{injection}, \quad y_{ki} = y_{ki,o} \tag{38}$$

$$\alpha = 0, \quad \frac{\partial y_{ki}}{\partial \alpha} = \frac{1}{17}\left[\frac{r_k^2}{D_k}\right]\frac{\partial y_{ki}}{\partial t} \tag{39}$$

$$\alpha = 1, \quad \frac{\partial y_{ki}}{\partial \alpha} = -\frac{3}{32}\frac{1}{D_k}\left[y_{kis} - \epsilon_i\right]\frac{ds_k}{dt} \tag{40}$$

By knowing the mass fractions of the liquid species at the droplet surface, the corresponding mole fractions are determined by

$$x_{iks} = \frac{y_{iks}/M_i}{\sum_i y_{yks}/M_i} \tag{41}$$

At the droplet interface, the mole fractions of the gas species are obtained by means of Raoult's law:

$$x_{is} = \frac{1}{P} x_{iks} P_{is} \tag{42}$$

where the partial pressure, P_{is}, is determined by means of the Clausius-Clapeyron relationship:

$$P_{is} = exp\left[\frac{l_{ki}}{R_u}\left(\frac{1}{T_{bi}} - \frac{1}{T_{ks}}\right)\right] \tag{43}$$

Then, the corresponding mass fractions in the gas-phase at the droplet interface are given by

$$y_{is} = \frac{x_{is} M_i}{M_a(1 - \sum_i x_{is}) + \sum_i M_i x_{is}} \tag{44}$$

where M_a is the molecular weight of the gas excluding fuel vapor. And the fractional mass vaporization rate of liquid species, ϵ_i, is given by

$$\epsilon_i = y_{is} + (1 - y_{fs})\frac{y_{is} - y_{fi}}{y_{fs} - y_f} \tag{45}$$

It is noteworthy that the thermodynamic and transport properties at the gas film are calculated at the temperature and composition as determined by the following one-third rule:

$$\phi_{avg} = \frac{1}{3}\phi_g + \frac{2}{3}\phi_{ks} \tag{46}$$

The correlations for the gas-phase thermodynamic and transport properties are described in Sections 4 & 6. In a similar way, the liquid-phase thermodynamic and transport properties are determined based on the correlations described in the next section.

8 LOW PRESSURE LIQUID THERMODYNAMIC & TRANSPORT PROPERTIES

The specific heat at constant pressure, C_{pl}, thermal conductivity, λ_l, and viscosity, μ_l, are evaluated by means of the following expressions:

$$C_{pl} = C_{pl0} + C_{pl1}T + C_{pl2}T^2 + C_{pl3}T^3 + C_{pl4}T^4 \tag{47}$$

$$\lambda_l = \lambda_{l0} + \lambda_{l1}T + \lambda_{l2}T^2 + \lambda_{l3}T^3 + \lambda_{l4}T^4 + \lambda_{l5}T^5 \tag{48}$$

$$ln\ \mu_l = \mu_{l0} + \mu_{l1}/T + \mu_{l2}T + \mu_{l3}T^2 \tag{49}$$

where C_{pl} is in J/(kg K), μ_l in (μPA s), and λ_l in (W/m K).

Tables 3-5 provide the polynomial constants used in Eqs. (47)-(49) for some of the species that we found to be of interest in our spray computations. Table 3 provides the constants for the liquid specific heat, C_{pl}, Table 4 for the liquid thermal conductivity, λ_l, and Table 5 for the liquid molecular viscosity, μ_l. These tables are compiled with the data taken mostly from the references of [16,18].

The binary diffusion coefficient, D_{ij}, is evaluated as follows [16]:

$$D_{ij} = \frac{K_{dif} T}{\mu_j V_i^{1/3}} \tag{50}$$

Table 3. Polynomial constants for liquid specific heat.

Fuel	c_{pl0}	c_{pl1}	c_{pl2}	c_{pl3}	c_{pl4}
C_6H_{14}	2.4169	-5.9866e-03	2.0959e-05	-8.4489e-09	0.0
C_7H_{16}	4.8227	-3.6980e-02	1.6777e-04	-3.0987e-07	2.2081e-10
C_8H_{18}	9.2189	-8.8314e-02	3.8869e-04	-7.2539e-07	5.0776e-10
$C_{10}H_{22}$	4.7991	-2.8643e-02	9.3619e-05	-8.9516e-08	0.0
$C_{12}H_{26}$	4.7900	-2.8643e-02	9.3619e-05	-8.9516e-08	0.0
$C_{14}H_{30}$	4.7991	-2.8643e-02	9.3619e-05	-8.9516e-08	0.0

Table 4. Polynomial constants for liquid thermal conductivity.

Fuel	λ_{l0}	λ_{l1}	λ_{l2}	λ_{l3}	λ_{l4}	λ_{l5}
C_6H_{14}	0.37078	-5.4313e-03	4.628e-05	-1.8002e-07	3.2243e-10	-2.1832e-13
C_7H_{16}	0.13236	9.4441e-04	-6.588d-06	1.4617e-08	-1.1244e-11	0.0
C_8H_{18}	0.25652	-7.5401e-04	1.5872e-06	-1.6795e-09	-1.3375e-16	0.0
$C_{10}H_{22}$	0.22179	-2.3699e-04	-6.94e-07	2.0415e-09	-1.5741e-12	0.0
$C_{12}H_{26}$	0.17609	4.2463e-05	-7.4467e-07	6.9446e-10	0.0	0.0
$C_{14}H_{30}$	0.18801	-9.1399e-05	-2.1464e-07	1.1655e-10	0.0	0.0
N_2	-2.629E-1	-1.545E-3	-9.450E-7	0.0	0.0	0.0
O_2	2.444E-1	-8.813E-4	-2.023E-6	0.0	0.0	0.0
CO_2	4.070E-1	-8.438E-4	-9.626E-7	0.0	0.0	0.0
H_2O	-3.838E-1	5.254E-3	-6.369E-6	0.0	0.0	0.0

Table 5. Polynomial constants for liquid molecular viscosity.

Fuel	μ_{l0}	μ_{l1}	μ_{l2}	μ_{l3}
C_6H_{14}	-4.034E+00	8.354E+02	0.0000000	0.0000000
C_7H_{16}	-4.325E+00	1.006E+03	0.0000000	0.0000000
C_8H_{18}	-4.333E+00	1.091E+03	0.0000000	0.0000000
$C_{10}H_{22}$	-4.460E+00	1.286E+03	0.0000000	0.0000000
$C_{12}H_{26}$	-4.562E+00	1.454E+03	0.0000000	0.0000000
$C_{14}H_{30}$	-4.615E+00	1.588E+03	0.0000000	0.0000000
N_2	-2.795E+01	8.660E+02	2.763E-01	-1.084E-03
O_2	-4.771E+00	2.146E+02	1.389E+02	-6.255E-05
CO_2	-3.097E+00	4.886E+01	2.381E-02	-7.840E-05
H_2O	-2.471E+01	4.209E+03	4.527E-02	-3.376E-05

where $K_{dif} = 8.210^{-8}(1 + [\frac{3V_j}{V_i}]^{2/3})$. One should be careful in using this approximation as it is based on a scarce set of available experimental data.

The specific heat for a multicomponent mixture is given by:

$$C_{pm} = \sum_{i=1}^{n} y_i C_{pi} \tag{51}$$

And the thermal conductivity of a multicomponent mixture is calculated by means of the Li method [16].

$$\lambda_m = \sum_{i=1}^{n}\sum_{j=1}^{n} \phi_i \phi_j \lambda_{ij} \tag{52}$$

where

$$\begin{aligned} \lambda_{ij} &= 2(\lambda_i^{-1} + \lambda_j^{-1})^{-1} \\ \phi_i &= \frac{x_i V_i}{\sum_{j=1}^{n} x_j V_j} \end{aligned}$$

where x_i is the mole fraction of the species i, ϕ_i is a volume fraction of the ith species, and V_i is the molar volume of the pure fluid.

9 DETAILS OF DROPLET FUEL INJECTION

The success of any spray model depends a great deal on the specification of the appropriate injector exit conditions. However, a discussion involving the physics of liquid atomization is beyond the scope of this subject matter. A great deal of research still needs to be done before we would be able to model the underlying atomization characteristics of different injectors from the first principles. However, in our present computations, we rely on some widely-used correlations to provide for the initial spray conditions. And the liquid fuel injection is simulated by introducing a discretized parcel of liquid mass in the form of spherical droplets at the beginning of every fuel-injection time step, Δt_{il}.

The spray computations facilitate the use of multiple fuel injectors. The same or a different type of liquid fuel can be specified for each one of different injectors. The initial droplet temperature is assumed to be the same for all different droplet groups of any given injector. The following three variables play an important role in simulating the injector initial conditions:

- no_of_holes() - the number of holes per injector,
- no_of_streams() - the number of droplet streams per hole - it is introduced to distinguish the initial velocity variation within different droplet classes arising from the geometric considerations of a chosen spray, &
- no_of_droplet_groups() - the number of droplet groups per stream - it is introduced to distinguish the droplet-size variation within different droplet classes of a polydisperse spray.

The total number of droplet groups introduced at the beginning of every different injection time step for a given injector is therefore equal to a value given by the multiplication of these three variables. Further details on some of the spray input variables can be found in the spray input file, ncc_injector.in.1 of Table 6, where the integer number followed by the last dot represents the injector number, which in this case happens to be one. The table provides a complete description of the following input variables: out_string, tdrop(n_i), ymki(n_i,n_l), spray_table, steady_spray_model, no_of_holes(n_i), no_of_streams(n_i), no_of_droplet_groups(n_i), lmdis, smdm, cone, ((x_inj(nx), y_inj(nx), z_inj(nx), flowf(nx), v_inj(nx), alpha_inj(nx), beta_inj(nx), theta_inj(nx), dtheta_inj(nx), swlr_angle(nx), size_min(nx), size_max(nx)), nx=1,no_of_holes(n_i)).

The initial droplet distribution for each injector could be defined either by a complete specification of the initial conditions through a spray table or through the specification of some input parameters that invoke certain pre-defined correlations.

When a spray table is defined, one should provide the information on the (x,y z) components of the initial droplet location, the (u, v, w) components of the droplet velocity, and the initial mass flow rate associated with each different droplet group as described in ncc_spray_table.1 of Table 7. This table provides a complete description of the following variables: nos(n_i), (ni,xx_inj, yy_inj, zz_inj, uu_inj, vv_inj, ww_inj, r_inj, fld_d), ,ni=1,nos(n_i)). The initial inputs specified through a spray table should be representative of the integrated averages of the experimental conditions [10-13].

The remainder of the discussion in this section applies to the case when a spray table is not defined. In which case, we need to specify several parameters including no_of_holes(), no_of_streams(), & no_of_droplet_groups(). And depending on what is specified for the input of the logical variable, lmdis, in ncc_injector.in.1, the droplet-size distribution within

Table 6. Typical ncc_liquid_injector.in.01 file.

Input file content	**comments**
heading	title of a description of this file
heading	title of property
nolc(n_i)	denotes the total number of initial liquid components in the n_i-th injector.
heading	title of property
out_string	chemical symbols of initial liquid species: e.g., C6H12.
(ymki(n_i,n_l), n_l=1,nolc(n_i))	initial mass fractions of liquid species
heading	title of controlling parameters
tdrop(n_i)	initial droplet temperature
heading	title of controlling parameters
spray_table, steady_spray_model	If spray_table = .true., initial droplet location, velocity, size, and mass flow rate are defined in the file - ncc_liquid_table.in.1. Otherwise they are determined based on some form of specified spray correlations and configurations. If steady_spray_model = .true., it invokes a steady state spray model commonly used in many spray codes, where whenever the spray solver is called, after first introducing a new group of spray particles, it continues with the liquid-phase computation until after all the particles are taken out of the computational domain. (NOTE: It is **NOT** recommended to use this option as a steady state calculation could be better arrived at by making use of some features of the unsteady option.) The solution from the unsteady option (steady_spray_model = .false.) is determined based on the values assigned to the controlling time steps - dtml, dtil, & dtgl, which are internal to the spray solver.
heading	title of controlling parameters
no_of_holes(n_i), no_of_streams(n_i), & no_of_droplet_groups(n_i)	no_of_holes(n_i) = The number of holes per injector. no_of_streams(n_i) = The number of droplet streams per hole. This variable is introduced mainly to distinguish the variation in the droplet groups based on angular orientation. no_of_droplet_groups(n_i) = The number of droplet groups per stream. This variable is introduced mainly to distinguish the droplet groups based on the droplet-size variation. NOTE: when spray_table = .true., both no_of_holes and no_of_streams are set equal to 1 and all the different droplet groups are grouped into a single variable, no_of_droplet_groups().

Table 6. Typical ncc_liquid_injector.in.01 file (continued).

Input file content	**comments**
heading	title of controlling parameters
lmdis, smdm, & cone	If lmdis = .true., it invokes a correlation for droplet size distribution. Otherwise it assumes a uniform droplet distribution between the maximum and minimum initial droplet sizes as defined by size_max() & size_min(). smdm = Sauter mean diameter If cone = .true., it activates a 3D solid or hollow cone spray configuration as shown in Fig. 2. Otherwise it activates a 2D configuration of either an axis-of-symmetry case of Fig. 3 or a planar case of Fig. 4 depending on the value of the logical variable - axisymmetric.
heading	title of controlling parameters
((x_inj(nx),y_inj(nx), z_inj(nx),flowf(nx), v_inj(nx),alpha_inj(nx), beta_inj(nx),theta_inj(nx), dtheta_inj(nx), swlr_angle(nx), size_min(nx),size_max(nx)), nx=1,no_of_holes(n_i))	(x_inj(nx),y_inj(nx),z_inj(nx)) = spatial coordinates of the initial injector location. flowf(nx) = injector mass flow rate, (units - kgm/s for 3D & axis-of-symmetry & kgm/s/m for 2D planar. REMEMBER FOR AXIS-OF-SYMMETRY, IT IS THE TOTAL FLOW RATE OVER 360 DEGS.) alpha_inj(nx) = angle of rotation towards z-axis. beta_inj(nx) = angle of rotation towards y-axis. theta_inj(nx) = cone angle. dtheta_inj(nx) = half-cone angle. (NOTE: Although dtheta_inj(nx) = theta_inj(nx)/2 for SOLID CONE SPRAY, it is invoked by setting dtheta_inj(nx) either equal to 0 or theta_inj(nx)/2.) Refer to Figs. 2-4, for a better understanding of the angular representation. swlr_angle(nx) = swirl angle. size_min(nx) & size_max(nx) = The variables are associated with the droplet size distribution of lmdis = .false.

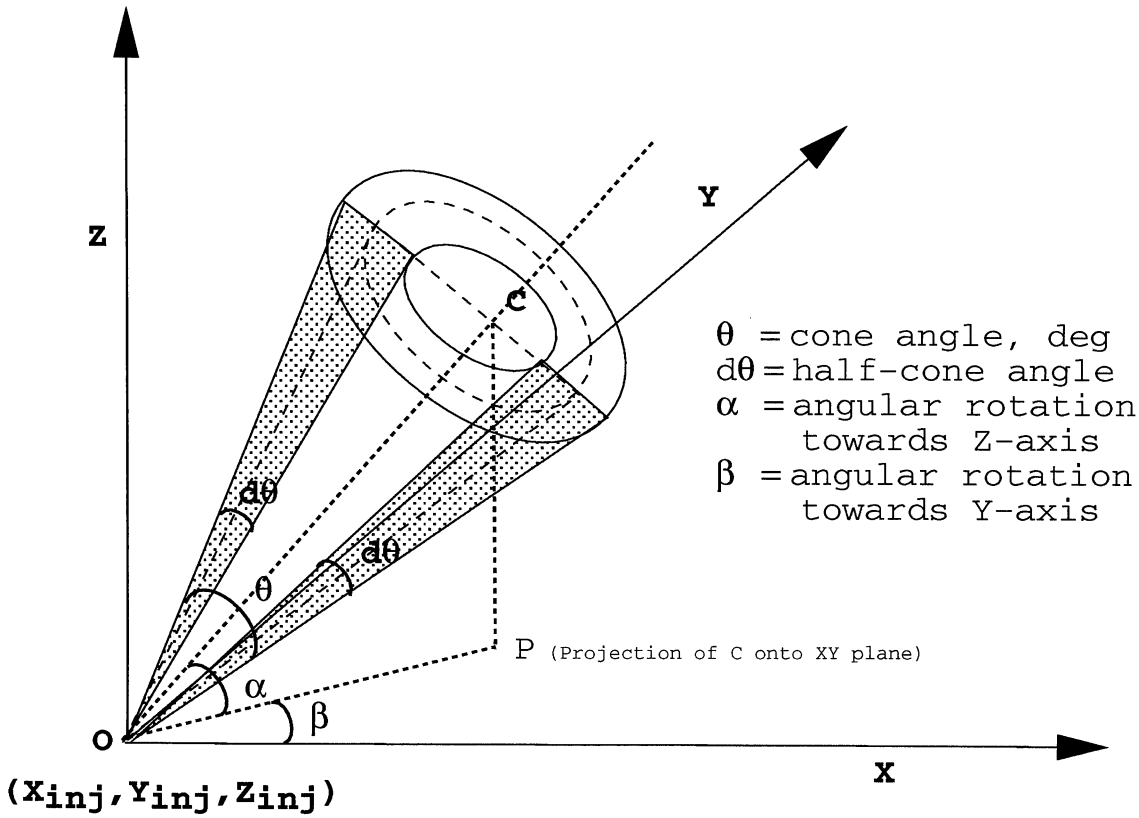


Fig 2a. Geometrical details of fuel injection for a 3D solid or hollow cone spray.

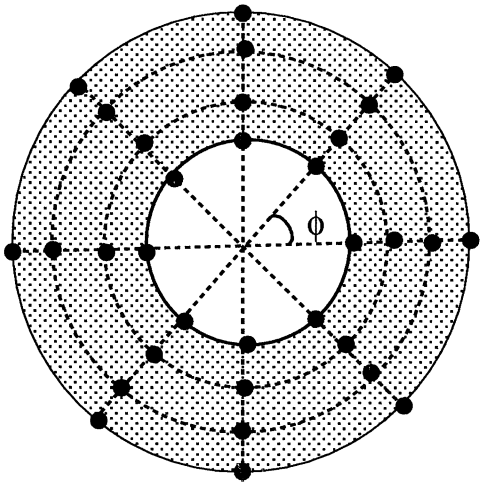


Fig 2b. Initial spray particle orientation in a circular corss-section.

θ = cone angle, deg
dθ = half-cone angle

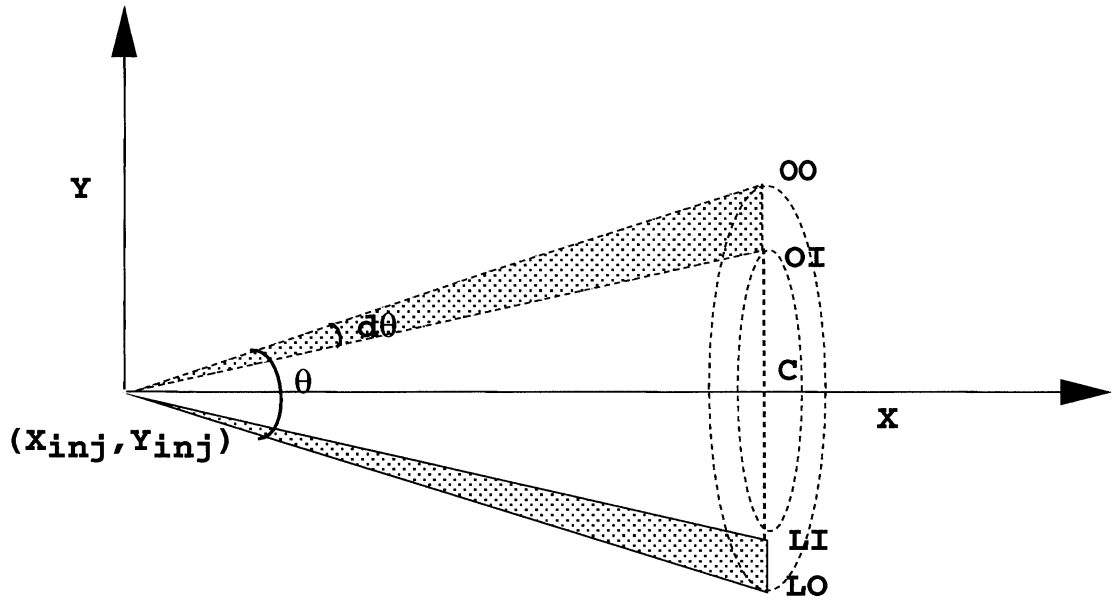
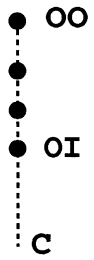


Fig 3a. Geometrical details of fuel injection for an axis-of-symmetry case.



For axis of symmetry:
no_of_streams() is distributed between **OI** to **OO**.

Fig 3b. Initial spray particle concentration in an axis-of-symmetry case.

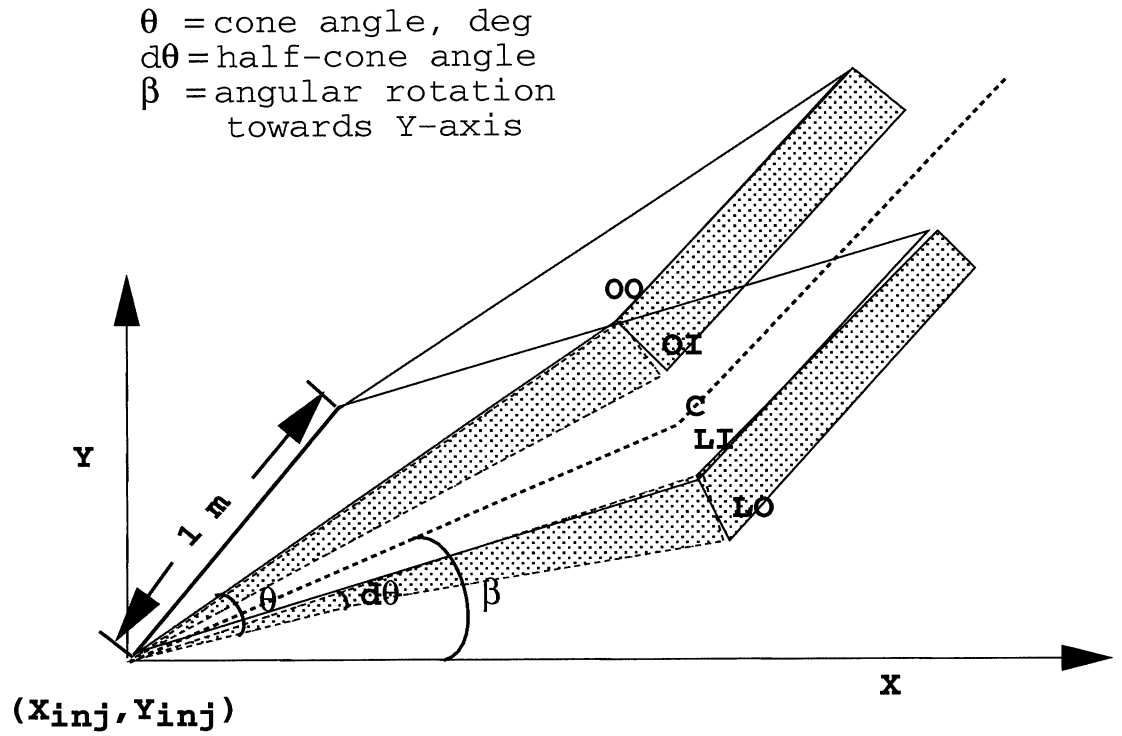


Fig 4a. Geometrical details of fuel injection for a 2D planar case.

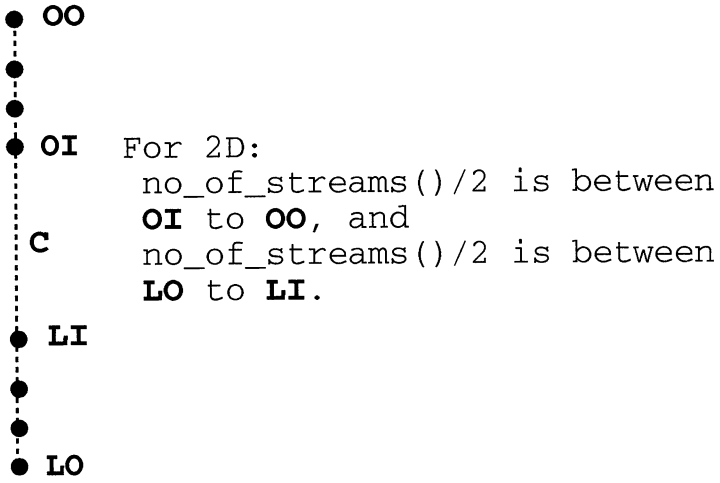


Fig 4b. Initial spray particle concentration in a 2D planar case.

Table 7. ncc_spray_table.in.01 file.

Input file content	comments
nos(n_i)	denotes the total number of droplet groups in the n_i-th injector.
(ni,xx_inj,yy_inj, zz_inj,uu_inj,vv_inj, ww_inj,r_inj,fld_d) ,ni=1,nos(n_i))	ni = number of the droplet group. Its value ranges between 1 to nos(n_i). (xx_inj,yy_inj,zz_inj) = spatial coordinates. (uu_inj,vv_inj,ww_inj) = velocity components. r_inj = droplet size in radius. fld_d = mass flow rate of the ni-th droplet group. (NOTE: SUMMATION OF fld_d OVER ALL THE nos(n_i) DROPLET GROUPS IS EQUAL TO THE TOTAL MASS FLOW RATE OF THE INJECTOR.)

each one of the streams is determined based on either one of the two options:

- In one option, it is calculated based on a correlation typical of those widely used in describing the initial droplet size distribution [4]:

$$\frac{dn}{n} = 4.21\ x\ 10^6 \left[\frac{d}{d_{32}}\right]^{3.5} e^{-16.98\left(\frac{d}{d_{32}}\right)^{0.4}} \frac{dd}{d_{32}} \tag{53}$$

  where n is the total number of droplets and dn is the number of droplets in the size range between d and $d + dd$. This correlation also requires the specification of Sauter mean diameter, d_{32}. Fig. 5 shows the droplet size distribution generated by Eq. (53) for a case studied in [10]. The solid line shows the droplet number variation versus drop size and the dashed line shows the integrated mass variation with drop size. The drop size distribution within the spray is represented by a finite number of droplet classes as given by the variable, no_of_droplet_groups().

- In the second option, the initial droplet sizes are distributed evenly between two specified maximum and minimum drop sizes - size_max() & size_min() as specified in Table 6. And the size interval depends on the value specified for the variable, no_of_droplet_groups(). However, this option is applicable only in certain special cases.

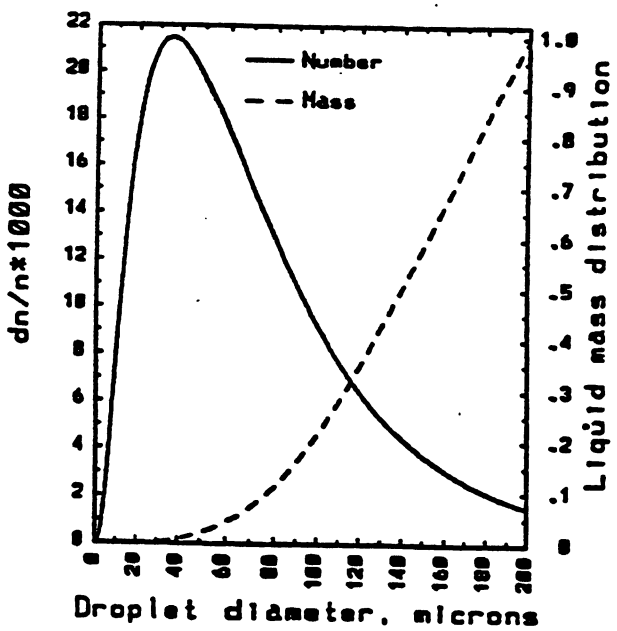


Figure 5 Droplet-size distribution.

Depending on what is specified for the logical variable, cone, of Table 6, the droplet velocity distribution amongst various streams of a given hole is calculated by assuming the spray to be either a solid or hollow cone spray. A graphical illustration of three different cone configurations are shown in Figs. 2 to 4. Figs. 2a & 2b refer to a 3D case, Figs. 3a & 3b

to an axisymmetric case, and Figs. 4a & 4b to a 2D planar case. It is noteworthy that in an axisymmetric case, the x-axis is assumed to be aligned with the axis-of-symmetry.

Fig. 2a shows the geometric details of a hollow cone spray in 3D where θ is known as the cone angle, $d\theta$ is the half-cone angle, and for a solid cone spray $d\theta = \theta/2$. And α represents the angular rotation towards the z-axis and β is the angular rotation towards the y-axis. Fig. 2b shows initial spray stream orientation in a circular cross section. We try to simulate the spray by a finite number of streams as given by the variable, no_of_streams(). Each one of the dark circles in Fig. 2b represents a different stream. The streams are distributed evenly along each one of the different rays which are separated from each other with an angle of separation as given by ϕ. Currently, we have hard-coded the angle of separation to be 45^0 which means we have restricted the number of rays to be eight in 3D. Therefore, when specifying a number for no_of_streams(), it should be borne in mind that it should be a multiple of eight. And the number of streams along each one of the rays, therefore, becomes equal to no_of_streams()/8. In Fig. 2b, the no_of_streams() has a value of 32 and, therefore, the number of streams along each one of the rays for this case is 4.

Fig. 3a. shows the geometric details for an axisymmetric case. Since the computations are performed only in the first quadrant of the x-y plane, all of the specified number of streams, no_of_streams(), are distributed evenly between the lower and upper halfs as shown in Fig. 3a and 3b. Here, the upper half refers to the region between OI to OO and the lower half refers to LO to LI. It is noteworthy that each droplet group in a given stream represents a circular ring of liquid for an axisymmetric case.

The geometric details for a 2D case are shown in Figs. 4a & 4b. Here, it is noteworthy that each droplet group in a given stream represents a planar sheet of liquid. All the specified number of streams, no_of_streams(), are distributed evenly over both sides of the cone center as shown in Fig. 4b.

For each one of the different injector holes, no_of_holes(), it requires the specification of the following parameters as described in ncc_injector.in.1 of Table 6 (However, because of the geometric differences that exist between 3D, 2D, and axisymmetric configurations, some of the input parameters may have different units as noted below):

- The initial (x, y, z) coordinates of the hole location.

- The mass flow rate per hole - however, the definition of the units for the injector mass flow rate per hole differs: it is kgm/s for 3D & the axis-of-symmetry and kgm/s/m for 2D planar (**NOTE:** In an axis-of-symmetry, the specified mass flow rate per hole refers to the entire mass flow rate over 360 degrees).

- The following variables define the angular orientation: α_{inj} =angle of rotation towards z-axis, β_{inj} = angle of rotation towards y-axis, θ_{inj} = cone angle, & $d\theta_{inj}$ = half-cone angle (**NOTE:** Although $d\theta_{inj} = \theta_{inj}/2$ for a solid cone spray, a specified value of zero for $d\theta_{inj}$ also invokes a solid-cone spray configuration).

- The variable, swlr_angle(), allows a means to specify the tangential component in the case of both 3D and axisymmetric sprays.

10 SPRAY SOLUTION ALGORITHM

In order to evaluate the initial conditions that are needed in the integration of the liquid-phase equations, we first need to know the surrounding gas-phase properties at each particle location. But in order to evaluate the gas-phase properties, it is first necessary to identify the computational cell in which a given particle is located. It is a trivial task to track a particle in the regular rectangular coordinates. However, the particle tracking becomes complicated when the computational cell is no longer rectangular in the physical domain and it becomes even more complicated when the particle search is performed within the context of parallel computing.

We have developed and implemented an efficient particle-tracking algorithm for use with parallel computing in an unstructured grid. The search is initiated in the form of a local search originating from the computational cell in which the same particle was found to be located in the previous time step. The location of the computational cell is then determined by first evaluating the dot product of $\underline{x}_{pc} \cdot \underline{a}_n = |x_{pc}|\ |a_n|\ cos\,(\phi)$, where $\underline{x}_{pc}$ is the vector defined by the distance between the particle location and the center of the n-face of the computational cell, $\underline{a}_n$ is the outward area normal of the n-face as shown in Fig. 6, and ϕ is the angle between the two vectors.

A simple test for the particle location requires that the dot product be negative over each and every one of the n-faces of the computational cell. If the test fails, the particle search is carried over to the adjacent cells of those faces for which the dot product turns out to be positive. Some of those n-faces might represent the boundaries of the computational domain while the others represent the interfaces between two adjoining interior cells. The search is first carried over to the adjacent interior cells in the direction pointed out by the positive sign of the dot products. The boundary conditions are only implemented after making sure that all other remaining possibilities point towards a search exterior of the computational domain. This implementation ensures against any inadvertent application of the boundary conditions before the correct interior cell could be identified.

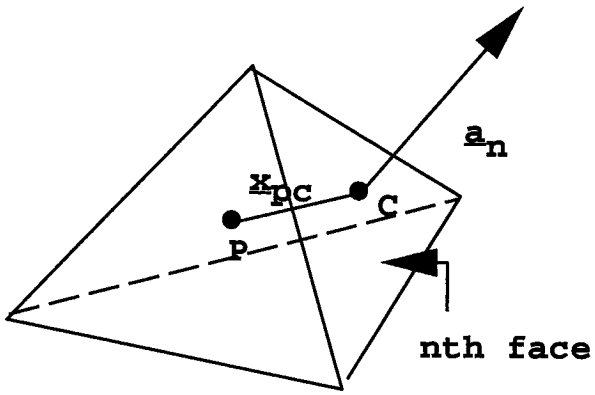


Fig. 6 A vector illustration used in the particle search analysis.

After the gas-phase properties at the particle location are known, the solution for the ordinary differential equations of particle position, size, and velocity are advanced by making use of a second-order accurate Runge-Kutta method. The partial differential equations governing the droplet internal thermal and mass transport are integrated by making use of a fully implicit Newton-Raphson iteration method.

Finally, the liquid-phase source terms of the gas-phase conservation equations (1-4) are evaluated by making use of a time-averaging method.

11 THE FLOW STRUCTURE OF THE SPRAY CODE & A DESCRIPTION OF THE TIME-AVERAGING SCHEME USED IN THE CALCULATION OF THE SOURCE TERMS

- In order to know more about the time-averaging method, we need to know first about the three different time steps that are internal to the spray code: Δt_{ml}, Δt_{il}, and Δt_{gl}.

  Δt_{ml} - the actual time step used in integrating the liquid-phase equations which is determined based on the smallest of the different time scales associated with various rate-controlling phenomena of a rapidly vaporizing droplet. Some of the limiting scales are the average droplet lifetime, the time it takes for the droplet to traverse the local grid spacing, the time it takes for the droplet internal temperature to reach the liquid boiling temperature, & a relaxation time scale associated with droplet drag. The time-scale restriction based on these criteria can become quite severe for smaller drops - for drops of sizes less than a micron.

  Δt_{il} - the injection time step. It is the time step at which a new discretized parcel of different droplet groups are introduced into the computation.

  Δt_{gl} - the global time step. Its introduction seems to provide better convergence in both unsteady and steady-state computations.

- When the spray solver is called it advances the liquid phase equations over a number of iterations as determined by the ratio of $\Delta t_{gl}/\Delta t_{ml}$.

- It then evaluates the time-averaged contribution of the liquid-phase source terms, S_{gl}, of the gas-phase governing equations (1-4) as follows:

$$S_{gl} = \sum_{m=1}^{M} \frac{\Delta t_{ml}}{\Delta t_{gl}} S_{ml} \tag{54}$$

  where

$$\sum_{m=1}^{M} \Delta t_{ml} = \Delta t_{gl} \tag{55}$$

- The values for Δt_{ml}, Δt_{gl}, & Δt_{il} are specified in the input file, ncc_liquid_solver.in, of Table 8.

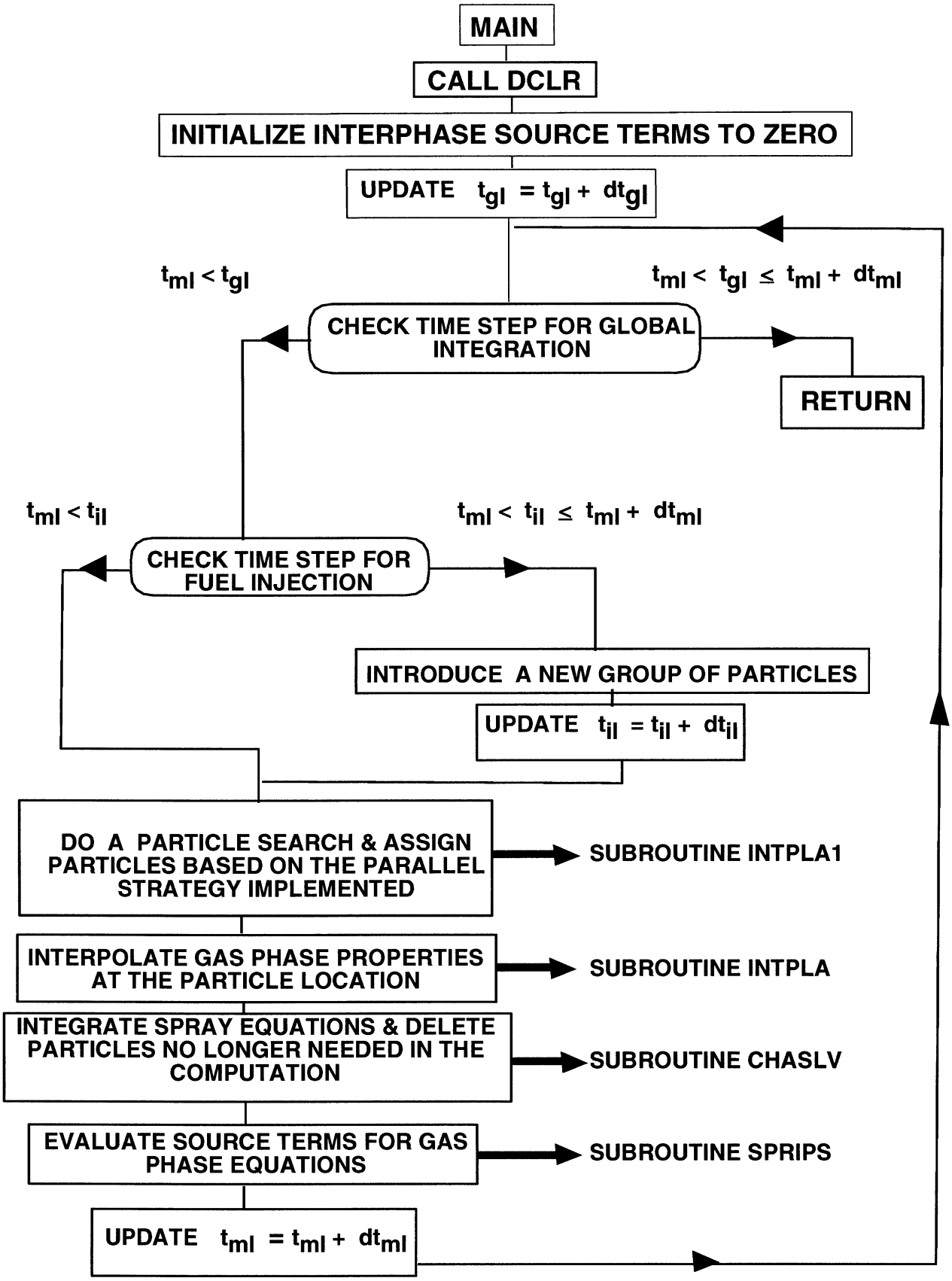


Fig. 7 The flow structure of LSPRAY–II.

- In steady-state computations, it is recommended to use for both Δt_{gl} and Δt_{il} a value of about 1 ms which is roughly equivalent to the average lifetime of the droplets for a typical reacting spray encountered in conventional low-pressure gas-turbine combustors.

The averaging scheme could be explained better through the use of a flow chart shown in Fig. 7. The main spray solver is invoked with a controlling routine, DCLR, which, then, executes the following steps:

1. It first initializes the source terms to zero.
2. Updates the global time, t_{gl}, based on Δt_{gl}.
3. Checks to see if $t_{ml} < t_{gl} < t_{ml} + \Delta t_{ml}$. If it is, it returns control over to the calling routine and supplies the other flow solvers, e.g., flow or EUPDF, with the source terms, S_{gl}, of Eqs. (1)-(4). If not, it proceeds with the next step.
4. Checks to see if it it is time to introduce a new group of particles.
5. Proceeds with solving the liquid-phase equations with calls to the following routines:
   - Calls the particle tracking routine and assigns particles based on the parallel strategy implemented.
   - Interpolates gas-phase properties at the particle location.
   - Advances liquid-phase equations and, then, deletes any particles that are no longer needed in the computations.
6. Evaluates the liquid-phase source-term contributions, S_{ml}, of Eq. 54.
7. Updates the time, t_{ml}, based on Δt_{ml}.
8. It then goes back to step (3) and repeats the whole process again until the computations are completed over a global time step: $\sum_{m=1}^{M} \Delta t_{ml} = \Delta t_{gl}$.

12 IMPLEMENTATION OF THE BOUNDARY CONDITIONS IN LSPRAY-II

The spray code supports most of the boundary conditions that are in use in the current version of the NCC CFD module but not all. Here, we would like to highlight some of the boundary conditions:

- In case of the droplet impingement with the combustor walls, the droplets may evaporate, move along the wall surfaces, and/or reflect with reduced momentum. The physics of droplet impingement with the walls is not completely understood. In our present calculations, it is assumed that the droplets, after having lost most of their momentum upon impingement with the walls, move along the wall surfaces with a velocity equal to that of the surrounding gas.
- The implementation of the periodic boundary conditions becomes rather complicated as it requires taking into consideration several aspects arising from a particle leaving the computational domain from one periodic boundary has to reenter the domain back from a corresponding second periodic boundary with appropriate conditions. The periodic boundary conditions are implemented with the help of some appropriately defined transformation matrices.
- The symmetric boundary condition is implemented in such a way to satisfy the criterion that for every particle crosssing the symmetry line, a similar one re-enters the domain in a direction given by the reflection off of the symmetry line.
- When the particles try to move out of the exit boundary, they are removed out of the computation.

13 DETAILS OF THE COUPLING BETWEEN LSPRAY-II AND THE OTHER SOLVERS

- The spray code is designed to be a stand-alone module such that it could easily be coupled with any other unstructured-grid CFD solver and the same holds true for EUPDF. (However, some grid-related parameters such as area vectors, grid connectivity parameters, etc. need to be supplied separately).
- The spray solver needs information on the gas velocity and scalar fields from the other solvers and, then, it in turn supplies the liquid-phase source terms.

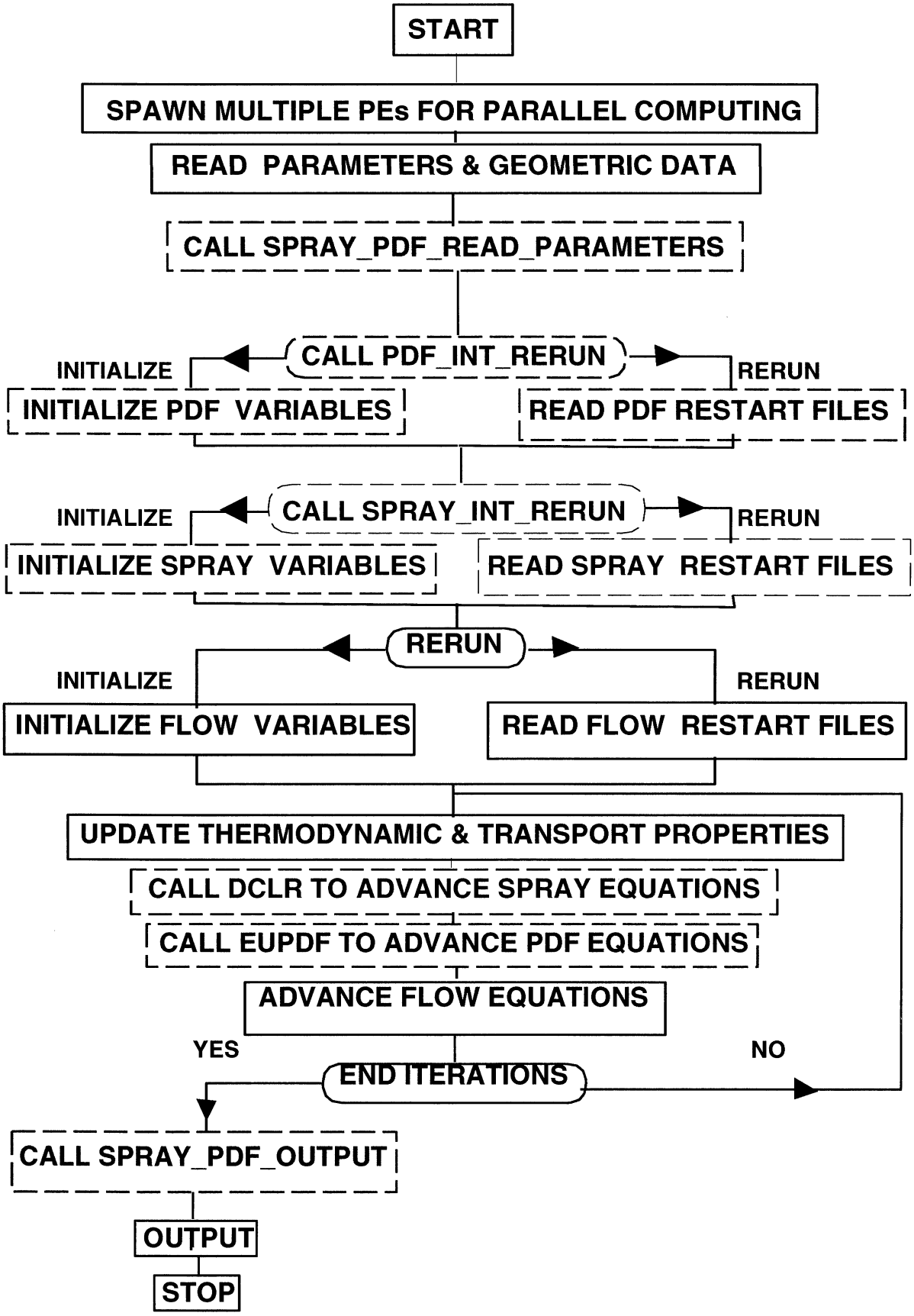


Fig. 8 The overall flow structure of the combined CFD, LSPRAY–II, and EUPDF solvers.

Table 8. ncc_liquid_solver.in file.

Input file content	**comments**
heading	title of controlling parameters
ldread,ispray_mod, (when_start_spray(n_i), n_i=1,no_of_injectors)	If ldread = .true., restarts the calculation from the data stored from the previous computation. Otherwise initiates a new spray computation. ispray_mod controls the calls to the spray solver. The spray solver is called once at every other CFD iteration as given by the number, ispray_mod. when_start_spray() represents the CFD iteration where the computations for the n_ith injector are initiated.
heading	title of controlling parameters
dtml, dtgl, & dtil	dtml = time step for advancing the liquid phase equations. dtgl = global time step. Whenever the spray solver is called, it advances the spray computations over a period of dtgl before returning control over to the calling routine. To be more precise, it advances the liquid phase equations over a number of time steps as determined by (dtgl/dtml). dtil = injection time step. It is where a new group of droplets are introduced into the computation.

- The PDF solver needs information on the mean gas velocity, turbulent diffusivity and frequency from the CFD solver and the liquid-phase source terms from the spray solver, and then it in turn provides the solution for the scalar (species and energy) fields to the flow and spray solvers.
- It should also be noted that both the PDF and spray solvers are called once at every other specified number of CFD iterations.
- All of the three solvers (LSPRAY-II, EUPDF, and CFD) are advanced sequentially in an iterative manner until a converged solution is obtained.
- All three codes (EUPDF, CFD, and LSPRAY-II) were coupled and parallelized in such a way to achieve maximum efficiency.

The coupling issues could be better understood through the use of a flow chart shown in Fig. 8. It shows the overall flow structure of the combined CFD, LSPRAY-II, and EUPDF modules. Both the PDF and spray codes are loosely coupled with the CFD code. The spray code is designed in such a way that only a minimal amount of effort is needed for its coupling with the flow and PDF solvers. The present version of the spray module relies entirely on the use of Fortran common blocks for its information exchange with other modules. Even this reliance should entail only few changes to be made within the spray code for its linkage with different solvers. The PDF code is also structured along similar coupling principles.

The flow chart of Fig. 8 contains several blocks - some shown in black and/or solid lines and the others in color and/or dashed lines. The ones in solid blocks represent the flow chart that is typical of a CFD solver. The ones in dashed blocks represent the additions arising from the coupling of the spray and PDF solvers.

The coupling starts with the calling of the subroutine - **spray_pdf_read_parameters**, which then reads the spray control parameters from the input file, *ncc_liquid_solver.in* of Table 8. This table provides a detailed description of the following input file variables - ldread, ispray_mod, (when_start_spray(n_i), n_i=1,no_of_injectors), dtml, dtgl, & dtil. The coupling is then followed by the calling of the **pdf_int_rerun** subroutine. It initializes PDF computations and, also, it may restart the PDF compu-

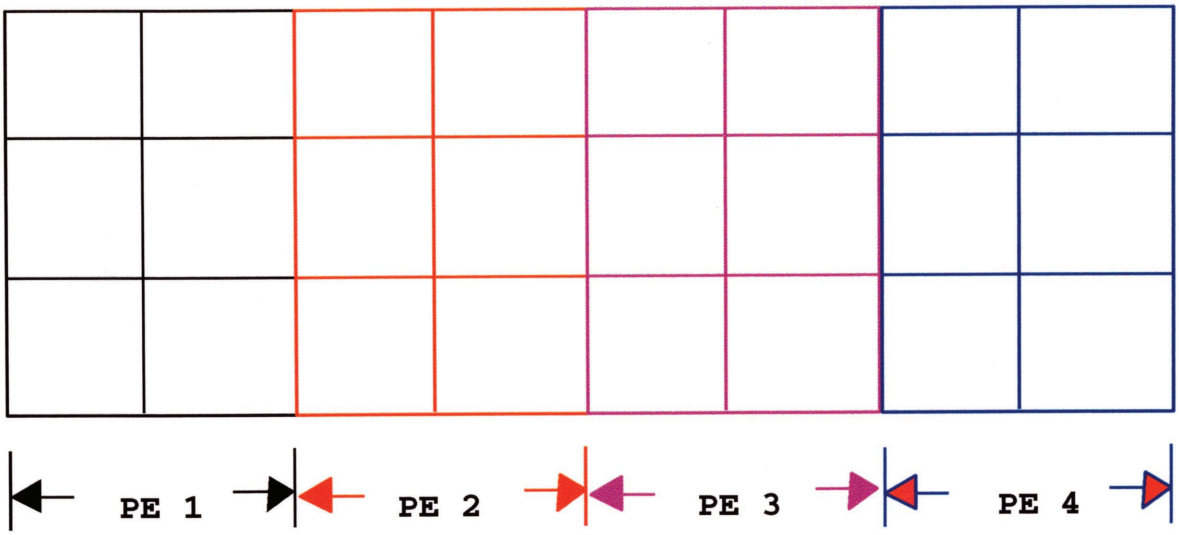


Fig. 9a An illustration of the parallelization strategy employed in the gas flow computations.

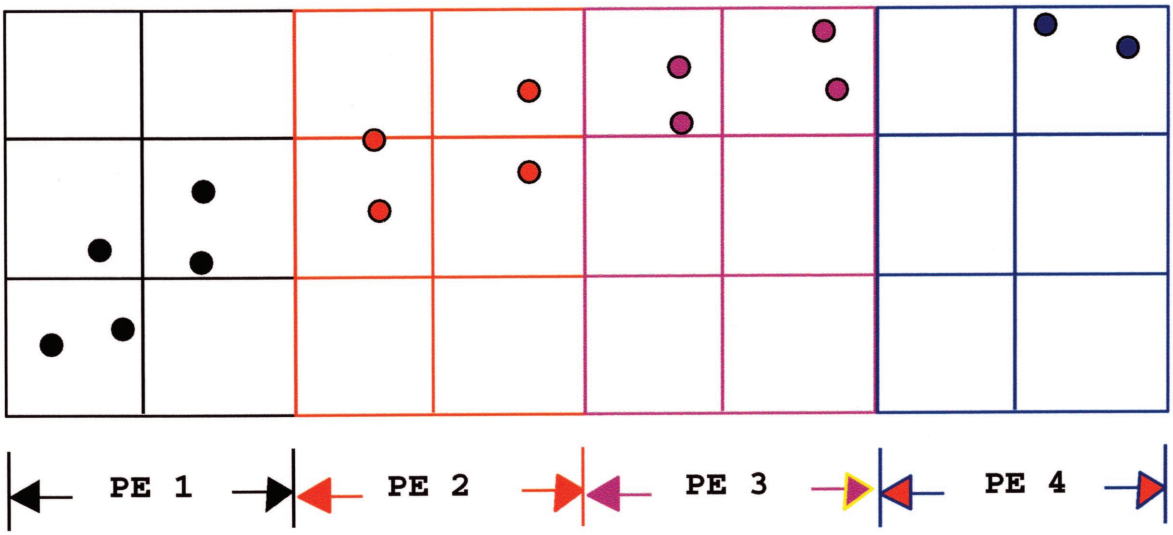


Fig. 9b An illustration of the parallelization strategy employed in the spray computations.

tations if needed from the data stored from a previous iteration. Similarly, we call **spray_int_rerun** for the spray computations. It is noteworthy that the spray computations can be restarted by reading the data from the restart files - *ncc_liquid_params.out* & *ncc_liquid_results.db*. The restart capability is invoked by setting the logical variable, ldread, of the input file, *ncc_liquid_solver.in* of Table 8, to be true. Otherwise, the spray computations are initialized to start from the beginning. Then, the coupling proceeds with the calling of the following subroutines: **dclr** for integrating the spray calculations and **eupdf** for the Monte Carlo PDF. The input variable, ispray_mod of Table 8, controls the calls to the spray integrating routine. The spray solver is called once at every other number of CFD iterations as specified by ispray_mod. And the first call to the spray solver is controlled by the input variable, when_start_spray() of Table 8, which represents the starting CFD iteration number from which the spray computations are initiated. Finally, the coupling ends with the calling of a subroutine, **spray_pdf_output**, which will create a set of new restart files.

14 DETAILS OF THE FORTRAN SUBROUTINES & FUNCTIONS

Table 9 provides a list of all the Fortran subroutines developed as a part of the spray module. This table also provides information on all the Fortran functions. It also describes the purpose of all the individual subroutines and functions.

15 DETAILS OF THE PARALLELIZATION

There are several issues associated with the parallelization of both the spray & PDF computations. The goal of the parallel implementation is to extract maximum parallelism so as to minimize the execution time for a given application on a specified number of processors [37]. Several types of overhead costs are associated with parallel implementation which include data dependency, communication, load imbalance, arithmetic, and memory overheads. The term arithmetic overhead is the extra arithmetic operations required by the parallel implementation. Memory overhead refers to the extra memory needed. Excessive memory overhead reduces the size of a problem that can be run on a given system and the other overheads result in performance degradation [37]. Any given application usually consists of several different phases that must be performed in certain sequential order. The degree of parallelism and data dependencies associated with each of the subtasks can vary widely [37]. The goal is to achieve maximum efficiency with a reasonable programming effort [37].

In our earlier work, we discussed the parallel implementation of a spray algorithm developed for the structured grid calculations on a Cray T3D [10]. These computations were performed in conjunction the Monte Carlo PDF method. The parallel algorithm made use of the shared memory constructs exclusive to Cray MPP (Massively Parallel Processing) Fortran and the computations showed a reasonable degree of parallel performance when they were performed on a NASA LeRC Cray T3D with the number of processors ranging between 8 to 32 [10]. Later on, the extension of this method to unstructured grids and parallel computing written in Fortran 77 with PVM or MPI calls was reported in [11-15]. The latest version in Fortran 77/90 offers greater computer platform independence. In this section, we only highlight some important aspects of parallelization from Refs. [10-15].

- Both the EUPDF and CFD modules are well suited for parallel implementation. For the gas-phase computations, the domain of computation is simply divided into n-Parts of nearly equal size and each part is solved by a different processor. Fig. 9a illustrates a simple example of the domain decomposition strategy adopted for the gas-phase computations where the total domain is simply divided equally amongst the available computer processing elements (PEs). In this case, we assumed the number of available PEs to be equal to four.

- But the spray computations are more difficult to parallelize for the reasons summarized below:

  (1) Non-uniform nature of spray distribution: Most of the particles are usually confined to a small region where the atomizer is located.

  (2) Dynamic nature of Lagrangian particles: Particles keep moving between different subdivided domains of an Eulerian grid (grid used in the CFD computation) which are assigned to different processors. While some new particles are introduced at the time of fuel injection, some others are taken out of computation.

Conceptually, there are several ways to parallelize the spray computations, we, however, devel-

Table 9. Description of LSPRAY-II Fortran subroutines & functions.

Function	**Purpose of the Function**
blasiu(x)	This function returns a solution for the function, $f(B_k)$, of Eq. (21) for use in computing the droplet regression rate.
peng_rob_cmpr (chem_model,ycomp, tij,density, number_of_species)	It provides a value for the pressure based on the solution of the Peng-Robinson EOS.
Subroutine	**Purpose of the Subroutines**
chaslv	This routine has two main functions: (1) It integrates the liquid phase equations. (2) It removes the particles that need to be taken out of the computation.
dclr	This routine is called once at every other CFD iteration as specified by ispray_mod. It is primarily a controlling routine for spray computations. It is only called in conjunction with steady_state_model = .false. This routine has the following functions: (1) It initializes the source terms to zero. (2) Checks to see if new particles need to be introduced. (3) Advances liquid phase equations over an allowable or pre-specified time step, dtml, with calls to the following routines: intpla1 - Interpolates the gas phase properties at the particle location. chaslv - Advances liquid phase equations. intpla - Identifies computational cells and PEs associated with particles. sprips - Evaluates the liquid phase source term contributions of the CFD and PDF equations.

Table 9. Description of LSPRAY-II Fortran subroutines & functions (continued).

Subroutine	Purpose of the Subroutines
	(4) Continues with steps (2) and (3) until the computations are completed over one single global time step, dtgl. (5) Returns control over to other solvers, e.g. CFD and EUPDF, and supply them with the source terms averaged over dtgl.
dclr_steady	This routine is similar to dclr but used only in conjunction with steady_state_model = .true. This routine has the following functions: (1) It initializes the source terms to zero. (2) Introduces a new set of spray particles. (3) Advances liquid phase equations over an allowable or pre-specified time step, dtml, with calls to the following routines: intpla1 - Interpolates gas phase properties at the particle locations. chaslv - Advances liquid phase equations. intpla - Identifies computational cells and PEs associated with particles. sprips - Evaluates the liquid phase source term contributions of the CFD and PDF equations. (4) It repeats step (3) until there are no more spray particles left to be integrated. (NOTE: The particles are taken out of computation when they reach either a certain size of negligible proportion or exit out of the computational domain.) (5) Returns control over to other solvers, e.g. CFD and EUPDF, and also supply them with source terms.

Table 9. Description of LSPRAY-II Fortran subroutines & functions (continued).

Subroutine	**Purpose of the Subroutines**
dropdis(rhol, flowdum,sr, fld,smd,nofg)	This routine computes droplet distribution from the following correlation (also, appears as Eq. 53): $dn/n = a((D/D_{32})^{alp})exp(-b((D/D_{32})^{bet}))dD/D_{32}$ where a, b, alp, and bet are constants.
find_transport_ds(element,ijle,ymgf)	It computes the following properties of the gas mixture at the droplet interface by making use of the one-third rule of Eq. (46): molecular viscosity, gas density, and diffusion coefficient.
find_xyzface (i)	This routine computes x, y, and z locations of all the face centers of an element, i. This information is used in the particle search algorithm.
get_liq_tat_properties (tmp,pmp,j)	Computes the following variable properties of a liquid mixture: density, specific heat, molecular viscosity, gas density, and diffusion coefficient.
hp_transport (tij,pij,x_work, its,avisc,acond)	It calculates the high pressure correction for the gas-phase transport properties: viscosity, conductivity, and diffusion.
hp_diff_table (ct,pt,dabcvi)	This subroutine provides a value for $(D_{AB}\ P)/(D_{AB}\ P)^{+}$ based on the Takahashi high-pressure correlation for the calculation of the diffusion coefficients in a gas mixture. For further details, see Reid et. al. [16], "The properties of Gases and Liquids." The correlation data is read from the file - hp_diff_table.dat.
intpla	This routine performs three main functions: (1) Particle Tracking - It identifies the computational cell in which a particle is located. In parallel computing, it also means identifying the corresponding processor of the computational domain in which a particle is located. (2) It implements appropriate boundary conditions. (3) It reassigns the particles between different PEs based on the parallel strategy employed.

Table 9. Description of LSPRAY-II Fortran subroutines & functions (continued).

Subroutine	**Purpose of the Subroutines**
intpla1	This routine interpolates the gas-phase properties at the particle location. In the present case, a simple first-order interpolation is employed.
mimd_spray	For computational elements whose neighboring cells are assigned to a different processor, it initializes arrays, ipr_fr_id() and ile_fr_id(), for storing information on the processor and element ID numbers of the neighboring cells. This information is needed in order to process the particle movement between the domains of the neighboring processors.
mimd_spray_recv (i_recvfrom)	This subroutine is called by mimd_spray in order to gather the relevant information from the neighboring processors.
mimd_spray_send (i_sendto)	This subroutine is also called by mimd_spray in order to send some relevant information to the neighboring processors.
par_loc(xparz, yparz,zparz, ipare,iparp)	Given the x,y,z coordinates of a particle location, the algorithm identifies the corresponding computational cell in which a particle is located. It also identifies the corresponding processor of the computational grid in which a particle is located. This information is useful when new droplet groups are introduced at the time of injection.
peng_rob (chem_model,ycomp, tij,pressure, number_of_species)	It provides a solution based on the Peng-Robinson EOS which is applicable for real gases at high pressures. It provides a value for the density of a gas mixture as the output variable.
peng_rob_gen (xsp,ppr,tpr, its,rho0)	It takes the following as the input variables: xsp() - the mass fraction of the species, ppr - pressure, tpr - temperature, and its - the number of species. And it returns as the output, rho0 - density based on the solution of the Peng-Robinson EOS.

Table 9. Description of LSPRAY-II Fortran subroutines & functions (continued).

Subroutine	**Purpose of the Subroutines**
peng_rob_sc (ppr,tpr,its)	It provides a solution for Z_{ic} and ρ_{ir} for ith species based on the solution of the Peng-Robinson EOS for a given value of p_{ir} and t_{ir}. It also requires the specification of the total number of species, its.
pr_eos_read_files	This routine reads the following three input files: For **ncc_liquid_physical_properties.dat,** it reads the following input variables of Table 1: tboil: normal boiling point, K tcrit: critical temperature, K pcrit: critical pressure, atm rholn: liquid density (at 1 bar, 273.15 K), kg/m^3 elhin: heat of vaporization at normal boiling point, KJ/kg volc: critical volume, cm^3/g-mole volln: molar volume at normal pressure, cm^3/g-mole aom: Pitzer's acentric factor sigma: characteristic diameter of a molecule, angstroms A^o ep_ka: epsilon/kappa: K For **ncc_liquid_binary_parameters.dat,** it reads the values of the binary parameters, K_{ij}, of Table 2. For **ncc_liquid_transport.dat,** it reads the polynomial polynomial constants of Tables 3-5 used in the evaluation of C_p, thermal conductivity, and viscosity.

Table 9. Description of LSPRAY-II Fortran subroutines & functions (continued).

Subroutine	**Purpose of the Subroutines**
prnspr	It writes standard spray output data.
root(idf,a,b,c,zv,zl)	It provides a value for the compressibility factor, Z, from the solution of Eq. (6). One of the three roots of Eq. (6) yields Z_v and the other Z_l.
spray_int_rerun	This routine initializes the spray solver based on the input read from various input files containing different spray parameters. In addition to initializing some of the variables, it also restarts the computations from the stored data of a previous computation if it is a rerun.
spray_plot- _output	It writes some useful plotting data when used in conjunction with the steady_spray_model = .false.
sprips	This routine computes the liquid-phase source terms of Eqs. (1)-(4) for use in both CFD and Monte Carlo PDF solvers. smlc(i) = liquid-phase contribution of Eq. (1) of Section 4. smlmx(i), smlmy(i), smlmz(i) = liquid-phase contribution of Eq. (3) of Section 4. smle(i) = liquid-phase contribution of Eq. (4) of Section 4.
sy(il,iu,bb, dd,aa,cc)	It is a tri-diagonal matrix solver. It is used in the solution of both Eqs. (27) & (37).
uvw_par(swlr_angle(nx), angle_work,nx,ny,v_inj, nmip,t_rotation, cone,n_cone_rays, cone_rotation,uloc, vloc,wloc,n_l)	It computes the particle injection velocity for different cone configurations of Figs. 2 to 4.

Table 10. CPU time (sec) per cycle versus number of PEs.

		Number of processors		
Solver	Characteristic	2	5	10
CFD	5 steps/cycle	2.50	1.25	0.75
EUPDF	1 step/cycle	6.5	2.9	1.9
LSPRAY-II	100 steps/cycle	1.70	0.64	0.53

Table 11. CPU Time (Sec) Per Cycle Versus Number of PEs.

		Number of Processors				
Solver	Characteristic	1	2	4	8	16
Spray	100 steps/cycle	6.83	5.29	2.94	1.64	0.87
Max. Spray Particles in a PE		2695	2097	1165	623	312
Min. Spray Particles in a PE		2695	598	118	14	0

oped and tested two different domain decomposition strategies [10-14].

- Strategy I:

  The Lagrangian particles were assigned fairly uniform amongst the available processors but the calculations associated with the particle tracking, the interpolation of the gas-phase properties, and the source-term evaluation were computed on the processor of the computational grid in which a particle is located.

  This strategy leads to an uniform loading during integration but leads to excessive message passing.

- Strategy II:

  The Lagrangian particles were assigned to the processor of the computational grid where the particle is located. Fig. 9b illustrates a simple example of the domain decomposition strategy adopted for the liquid-phase computations where the corresponding gas-flow computational domain is divided into equal parts between the four available PEs. In this strategy, the Lagrangian particles are assigned to the processor of the computational grid where a particle is located.

  This strategy lead to a non-uniform loading during integration but leads to less message passing.

Our experience has shown that Strategy II seems to work well on different computer platforms: both massively parallel computers as well as heterogeneous cluster of workstations. So in the present version of the code, we have opted to implement Strategy II over Strategy I.

16 DETAILS OF PARALLEL PERFORMANCE

The details of the combined parallel performance of the CFD, EUPDF, and LSPRAY-II codes involving several different cases can be found in Refs. [10-15]. Here, we only summarize briefly the parallel-performance results for two different cases. One is a 3D test case and more details on this case can be found in the reference [13]. For this case, the calculations were performed on a computational grid comprising of 8430 tetrahedral elements and 100 Monte Carlo PDF particles per cell. The computations were performed on one of the NASA Ames Research Center's parallel computer platforms called Turing which is a SGI Origin work-station with 24 PEs (Processor Elements). Table 10 summarizes the CPU times per cycle taken by the EUPDF, LSPRAY-II, and CFD solvers vs the number of PEs. Both the CFD and PDF solvers show good parallel performance with an increase in the number of processors but for the spray solver it shows reasonable parallel performance.

Next, we would like to summarize the results from [13, 38] showing only the results of spray computations. The results are summarized in Table 11. The computations were performed on Turing at NASA ARC (Ames Research Center) - it is a SGI Origin work-station with a maximum of 24 PEs (Processor Elements). The computations made use of an unstructured grid with a mesh size of 3600 tetrahedral elements. And it made use of about 2,695 Lagrangian particles for the spray computations and one hundred Monte Carlo particles per element for the PDF computations. In a given cycle of one global time step, dt_{gl}, the spray equations were advanced over one hundred time steps as given by $dt_{gl}/dt_{ml} = 100$. The first row of Table 11 summarizes the CPU times per PE per cycle taken by the spray solver vs the number of PEs. As expected, the CPU time goes down with an increase in the number of processors. In an ideal case, one would expect an inverse reduction in cpu time with an increase in the number of proces-

sors. Here, we don't get such an ideal performance because of the resulting non-uniform distribution of spray particles, between various participating processors, from the implementation of Stratergy II. To get an idea of the spray particle distribution, we have tabulated the maximum and minimum number of particles found between various processors. When we go from 1 to 2 PEs, 2097 particles are assigned to one and the rest to the second. With four they are distributed between 1165 and 118, with eight between 623 and 14, and with sixteen from 312 to 0. The results clearly show that the reduction in the CPU time varies almost linearly with the reduction in the number of maximum particles.

17 A SUMMARY OF SOME RECENT VALIDATION CASES INVOLVING BOTH REACTING AND NON-REACTING SPRAY COMPUTATIONS

A total of the following five cases were validated:

1. A reacting methanol spray with no-swirl.
2. A non-reacting methanol spray with no-swirl.
3. A confined swirl-stabilized n-heptane reacting spray.
4. An unconfined swirl-stabilized n-heptane reacting spray.
5. A confined swirl-stabilized kerosene reacting spray.

The experimental data for the first two cases was provided by McDonell & Samuelsen from the University of California at Irvine [38]. Both the cases are without swirl; one is a reacting case and the other is non-reacting. The data for the third and fourth cases was provided by Bulzan from the NASA Glenn Research Center [39-40]. Both the cases are swirl-stabilized reacting cases, one is an unconfined flame and the other is confined. The data for the last case was provided by El Banhawy & Whitelaw from Imperial College [4]. It is a confined swirl-stabilized kerosene spray flame. The first three cases made use of unstructured grids and the last two structured grids.

Here, we would like to provide a brief summary of the validation cases but a detailed presentation of the results and discussion can be found elsewhere in the papers [10-13]. The comparisons involved both gas and drop velocities, drop size distributions, drop spreading rates, and gas temperatures. The results were in reasonable agreement with the available experimental data. The comparisons also involved the results obtained from the use of the Monte Carlo PDF method as well as those obtained from a conventional CFD solution without the Monte Carlo PDF method. For the first case of McDonell & Samuelsen's reacting spray flame, the detailed comparisons clearly highlighted the importance of chemistry/turbulence interactions in the modeling of reacting sprays [13]. The results from the PDF and non-PDF methods were found to be markedly different with the PDF solution providing a better approximation to the reported experimental data. The PDF solution showed that most of the combustion occured in a predominantly diffusion-type of flame environment and the rest occuring in a predominantly premixed-type of flame environment. However, the non-PDF predictions showed incorrectly that most of the combustion occured in a predominantly vaporization-controlled regime. The Monte Carlo temperature distribution showed that the functional form of the PDF for the temperature fluctuations varied substantially from point to point. The results brought to the fore some of the deficiencies associated with the use of assumed-shape PDF methods in spray computations.

18 CONCLUDING REMARKS

- This manual provides a complete description of LSPRAY-II - a Lagrangian spray solver developed for application with parallel computing and unstructured grids.
- It facilitates the calculation of the multi-component liquid sprays with variable properties valid over a wide range of low pressure conditions.
- It provides the user with a basic understanding of the the spray formulation and the LSPRAY-II code structure, and complete details on how to couple the spray code to any other flow code.
- The basic structure adopted for the grid representation and parallelization for the gas side of the flow computations follows the guidelines established for NCC.
- Also, we have extended the joint scalar Monte Carlo PDF method to two-phase flows and,

thereby, demonstrating the importance of chemistry/turbulence interactions in the modeling of reacting sprays.

- Based on the validation studies involving several confined and unconfined spray flames, the results were found to be encouraging in terms of their ability to capture the overall structure of a spray flame.
- The source code of LSPRAY-II will be available with NCC as a complete package.

19 REFERENCES

1. W.A. Sirignano, Fluid Dynamics of Sprays, Journal of Fluids Engineering, vol. 115, no. 3, pp. 345-378, September 1993.
2. Crowe, C.T., "Numerical Models for Dilute Gas-Particle Flows," Journal of Fluids, Vol. 104, pp. 297-301, 1982.
3. C.T. Crowe, M.P. Sharma, and D.E. Stock, The Particle-Source-in Cell (PSI-CELL) Model for Gas-Droplet Flows, J. Fluids Eng., vol. 99, pp. 325, 1977.
4. Y. El Banhawy and J.H. Whitelaw, Calculation of the Flow Properties of a Confined Kerosene-Spray Flame, AIAA J., vol. 18, no. 12, pp. 1503-1510, 1980.
5. Faeth, G.M., "Mixing, transport, and Combustion in Sprays," Progress Energy Combustion Science, Vol. 13, pp. 293-345, 1987.
6. Raju, M.S., and Sirignano, W.A., "Spray Computations in a Centerbody Combustor," Proceedings of the 1987 ASME-JSME Thermal Engineering Joint Conference, Vol. 1, pp. 61-71, Honolulu, HI, March 1987. Also see Journal of Engineering for Gas Turbines and Power, Vol. 1, No. 4, pp. 710-718, October 1989.
7. Raju M.S., and Sirignano, W.A., "Multi-Component Spray Computations in a Modified Centerbody Combustor," Journal of Propulsion and Power, Vol. 6, No. 2, March-April 1990.
8. Raju, M.S., "AGNI-3D: A Computer Code for the Three-Dimensional Modeling of a Wankel Engine," Computers in Engine Technology, Proceedings IMechE, London, United Kingdom, pp. 27-37, 1991.
9. Raju, M.S., "Heat Transfer and Performance Characteristics of a Dual-Ignition Wankel Engine," Journal of Engines, the 1992 SAE Transactions, Section 3, pp. 466-509.
10. M.S. Raju, Application of Scalar Monte Carlo Probability Density Function Method For Turbulent Spray Flames, Numerical Heat Transfer, Part A, vol. 30, pp. 753-777, 1996.
11. M.S. Raju, Current Status of the Use of Parallel Computing in Turbulent Reacting Flows: Computations Involving Sprays, Scalar Monte Carlo Probability Density Function & Unstructured Grids, Advances in Numerical Heat Transfer, vol. 2, ch. 8, pp.259-287, 2000.
12. M.S. Raju, Scalar Monte Carlo PDF Computations of Spray Flames on Unstructured Grids With Parallel Computing, Numerical Heat Transfer, Part B, No. 2, Vol. 35, pp. 185-209, March 1999.
13. M.S. Raju, On the Importance of Chemistry/Turbulence Interactions in Spray Computations, Numerical Heat Transfer, Part B: Fundamentals, No. 5, Vol. 41, pp. 409-432, 2002.
14. M.S. Raju, LSPRAY - A Lagrangian Spray Solver - User's Manual, NASA/CR-97-206240, NASA Lewis Research Center, Cleveland, Ohio, November 1997.
15. M.S. Raju, EUPDF - An Eulerian-Based Monte Carlo Probability Density Function (PDF) Solver - User's Manual, NASA/CR-1998-20401, NASA Lewis Research Center, Cleveland, Ohio, April, 1998.
16. Reid, R.C., Prausnitz, J.M., and Poling, B.E., "The Properties of Gases and Liquids," 4th Edition, McGraw-Hill Book Company, 1987.
17. Prausnitz, J.M., and Chueh, P.L., "Computer Calculation For High pressure Vapor-Liquid Equilibria," Prentice-Hall, Englewood Cliffs, N.J., 1968.

18. Aggarwal, S.K., Shu, Z., Mongia, H., and Hura, H., "Multicomponent Fuel Effects on the Vaporization of a Surrogate Single-Component Fuel Droplet," AIAA Paper 98-0157, 36th Aerospace Sciences Meeting, Reno, Nevada, Jan. 12-15, 1998.

19. Aggarwal, S.K., Shu, Z., Mongia, H., and Hura, H., "Multicomponent Single-Component Fuel Droplet Evaporation Under High Pressure Conditions," 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit, Cleveland, OH, July 13-15, 1998.

20. Bellan, J., "Supercritical (and Subcritical) fluid Behavior and Modelling: Drops, Streams, Shear and Mixing Layers, Jets, and sprays," Prog. Energy Combust. Sci., Vol. 26, pp. 329-366, 2000.

21. R. Ryder, CORSAIR User's Manual: Version 1.0, SID: Y965, Pratt and Whitney Engineering, United Technologies Corporation, 25 January 1993.

22. N.S. Liu and R.M. Stubbs, Preview of National Combustion Code, AIAA 97-3114, AIAA/ASME/SAE/ASEE 33rd Joint Propulsion Conference, Seattle, Wash., July 6-9, 1997.

23. K.-H. Chen, A.T. Norris, A. Quealy, and N.-S. Liu, Benchmark Test Cases For the National Combustion Code, AIAA 98-3855, AIAA/ASME/SAE/ASEE 34th Joint Propulsion Conference, Cleveland, Ohio, July 13-15, 1998.

24. McBride, S. Gordan, and M. Reno, "Coefficients for Calculating Thermodynamic and Transport Properties of Individual species," NASA TM-4513, NASA Lewis Research Center, 1993.

25. C.R. Wilke, "A Viscosity Equation for Gas Mixture," Chem. Physics, Vol. 18, no. 4, pp. 517-519, April 1950.

26. F.M. White, Viscous Flows, McGraw-Hill Inc., 1974.

27. K. Kuo, Principles of Combustion, Wiley and Sons, 1976.

28. Peng, D.Y., and Robinson, D.B., Ind. Eng. Chem. Fundam., Vol. 15, pp. 59, 1976.

29. Reichenberg, D., "The Viscosities of Gases at High Pressures," Natl. Eng. Lab., Rept. Chem. 38, East Kilbride, Glascow, Scotland, August 1975.

30. Stiel, L.I., and Thodos, G., AICHE J., Vol. 10, pp. 26, 1964.

31. Takahashi, S., J. Chem. Eng. Japan, Vol.7, pp. 417, 1974.

32. Yuen, M.C., and Chen, L.W., "On Drag of Evaporating Droplets," Combust. Sci. Technol., Vol. 14, pp. 147-154, 1976.

33. A.A. Amsden, P.J. O'Rourke, and T.D. Butler, "KIVA-II: A Computer program For Chemically Reactive Flows With Sprays," LA-11560-MS, Los Alamos National Laboratory, Los Alamos, New Mexico 87545.

34. A.Y. Tong and W.A. Sirignano, Multi-component Transient Droplet Vaporization With Internal Circulation: Integral Formulation and Approximate Solution, Numerical Heat Transfer, vol. 10, pp. 253-278, 1986.

35. R. Clift, J.R. Grace, and M.E. Weber, Bubbles, Drops, and Particles, Academic, New York, 1978.

36. H. Schlichting, Boundary-Layer Theory, McGraw-Hill Series in Mechanical Engineering: McGraw-Hill, Inc., New York, 1968.

37. J.S. Ryan and S.K. Weeratunga, Parallel Computation of 3-D Navier-Stokes Flowfields for Supersonic Vehicles, AIAA 93-0064, AIAA 31st Aerospace Sciences Meeting, Reno, Nevada, 1993.

38. V.G. McDonell and G.S. Samuelson, An Experimental Data Base for the Computational Fluid Dynamics of Reacting and Nonreacting Methanol Sprays, J. Fluids Engineering, vol. 117, pp.145-153, 1995.

39. D.L. Bulzan, Structure of a Swirl-Stabilized, Combusting Spray, NASA Technical Memorandum: NASA TM-106724, Lewis Research Center, Cleveland, Ohio, 1994.

40. D.L. Bulzan, Velocity and Drop Size Measurements in a Confined, Swirl-Stabilized Combusting Spray, AIAA 96-3164, 32rd AIAA/ ASME/ SAE/ASEE Joint Propulsion Conference, July 01-03, 1996/Buena Vista, FL.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 2004	Final Contractor Report

4. TITLE AND SUBTITLE

LSPRAY–II: A Lagrangian Spray Module

5. FUNDING NUMBERS

WBS–22–714–20–06
NAS3–00145

6. AUTHOR(S)

M.S. Raju

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

QSS Group, Inc.
21000 Brookpark Road
Cleveland, Ohio 44135

8. PERFORMING ORGANIZATION REPORT NUMBER

E–14400

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA CR—2004-212958

11. SUPPLEMENTARY NOTES

Project Manager, Nan-Suey Liu, Turbomachinery and Propulsion Systems Division, NASA Glenn Research Center, organization code 5830, 216–433–8722.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited
Subject Categories: 07, 20, 01, and 02 Distribution: Nonstandard

Available electronically at http://gltrs.grc.nasa.gov

This publication is available from the NASA Center for AeroSpace Information, 301–621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

LSPRAY-II is a Lagrangian spray solver developed for application with parallel computing and unstructured grids. It is designed to be massively parallel and could easily be coupled with any existing gas-phase flow and/or Monte Carlo Probability Density Function (PDF) solvers. The solver accommodates the use of an unstructured mesh with mixed elements of either triangular, quadrilateral, and/or tetrahedral type for the gas flow grid representation. It is mainly designed to predict the flow, thermal and transport properties of a rapidly vaporizing spray because of its importance in aerospace application. The manual provides the user with an understanding of various models involved in the spray formulation, its code structure and solution algorithm, and various other issues related to parallelization and its coupling with other solvers. With the development of LSPRAY-II, we have advanced the state-of-the-art in spray computations in several important ways.

14. SUBJECT TERMS

Spray combustion modeling; Sprays; Combustion; Turbulent reacting flows; CFD

15. NUMBER OF PAGES

42

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	