



NRL/MR/6041--18-9816

Liquid-Fueled Detonation Modeling at the U.S. Naval Research Laboratory

DOUGLAS A. SCHWER

EUGENE P. O'FALLON, JR.

DAVID KESSLER

*Laboratories for Computational Physics and Fluid Dynamics
Materials Science and Component Technology Directorate*

October 3, 2018

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

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 this 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 Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 03-10-2018		2. REPORT TYPE Memorandum Report		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE Liquid-Fueled Detonation Modeling at the U.S. Naval Research Laboratory				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER 64-1J27-0-8-5	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Douglas A. Schwer, Eugene P. O'Fallon, Jr.* and David A. Kessler				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375				8. PERFORMING ORGANIZATION REPORT NUMBER NRL/MR/6041--18-9816	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR / MONITOR'S ACRONYM(S)	
				11. SPONSOR / MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES *NREIP Student Intern					
14. ABSTRACT This report details the development of detonation spray models at the U.S. Naval Research Laboratory (NRL). Previous work on sprays using particle tracking methods are adapted to the current CFD codes, specifically the DASLIB programs and the JENRE program, which has recently been extended for reacting flow fields. Initial results compare a series of shock tube simulations seeded with glass particle and water droplets to earlier work. Results show that the new DASLIB programs match previous results almost exactly, while the JENRE program also compares very well. The current simulations extend the previous work by showing more highly resolved two-dimensional results and improved thermodynamics. The report also considers a detonation tube with JP-10 droplets and gaseous oxygen. The gas-phase stoichiometric JP-10/oxygen detonation matches the calculated C-J detonation velocity of 2300 m/s, while the liquid JP-10/oxygen detonation with 10 mm droplets results in a detonation velocity of 2170 m/s. Two-dimensional liquid JP-10/oxygen simulations showed the formation of detonation cells and unique features not found in gaseous detonations.					
15. SUBJECT TERMS Liquid-Fueled, JP-10, Detonation, Particle-Tracking Modeling					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited	50	Douglas Schwer (202) 767-3615

This page intentionally left blank.

Contents

EXECUTIVE SUMMARY	E-1
1. INTRODUCTION	1
2. MULTI-PHASE MODEL.....	2
2.1 Gas-Phase Conservation Equations	2
2.2 Particle-Tracking Equations	3
2.3 Spray Gas-Phase Source Terms	7
3. SOLUTION PROCEDURE	8
3.1 DASLIB	8
3.2 JENRE	10
4. SHOCK-TUBE GEOMETRY AND CONDITIONS	12
5. GLASS PARTICLE SIMULATIONS.....	13
5.1 One-Dimensional Simulations with Particles.....	13
5.2 Two-Dimensional Simulations with Particles	16
6. WATER DROPLET SIMULATIONS.....	18
7. DETONATION SIMULATIONS	20
7.1 Detonation Tube and Conditions	20
7.2 Detonation Simulations	21
8. CONCLUSIONS.....	22
ACKNOWLEDGMENTS	23
APPENDIX A—ENTHALPY SHIFT FOR SPECIES ENTHALPIES	31
APPENDIX B—SAMPLE INPUT FILES.....	32
B.1 Two-Dimensional Glass Particle Simulation	32
B.2 Two-Dimensional Water Droplet Simulation	35
B.3 JP-10 Liquid Droplet Detonation File	39

List of Figures

1	One-dimensional shock tube geometry.....	12
2	Two-dimensional shock tube geometry.....	13
3	Comparison between Daslib and Jenre codes for Case 4, Mach 2.42, with no particles.	14
4	Comparison of Case 4, Mach 2.42 driver conditions, 6 ms into the simulation with (solid) and without (dashed) particle heating.	16
5	Schwer[13] two-dimensional simulation for Case 4 at 6 ms. Resolution is 0.5 cm. ...	17
6	DASLIB two-dimensional simulation for Case 4 at 6 ms. Resolution is 0.2 cm.	17
7	DASLIB two-dimensional simulation for Case 5 at 6 ms. Resolution is 0.2 cm.	18
8	Saturation pressure as function of temperature for water and steam.	19
9	Comparison of Case 4, Mach 2.42 driver conditions, 6 ms into the simulation.	24
10	Two dimensional simulation of water droplets for Case 5, 6 ms into the simulation...	25
11	Saturation pressure as a function of temperature for JP-10.....	25
12	Geometry of the JP-10 liquid detonation tube.	25
13	Comparison of gaseous and liquid-droplet JP-10/oxygen detonation simulations.	26
14	The maximum pressure for each cell at the end of the simulation.	26
15	Two-dimensional liquid spray JP-10/oxygen detonation simulation.	27
16	The maximum pressure for each cell at the end of the simulation.	27
17	Two dimensional liquid spray JP-10/oxygen detonation with $\phi = 0.77$	27

List of Tables

- 1 Driver conditions for each shock tube case as stated in Schwer[13]. Driven section:
 $P_1 = 1$ atm, $T_1 = 300$ K, air for both driver and driven sections with $\gamma = 1.4$ 13

Nomenclature

$A_{\text{Ant}}, B_{\text{Ant}}, C_{\text{Ant}}$	=	Antoine equation coefficients for saturation pressure
$\mathbf{a}_{p,i}$	=	acceleration of particle i
B	=	Spalding transfer number
c_k	=	concentration of species k
C_p	=	specific heat gas mixture
$D_{p,i}$	=	diameter of particle i
E	=	gas total energy
$\mathbf{F}_{d,i}$	=	drag force on particle i
$\mathbf{F}_{g,i}$	=	gravity force on particle i
f_v	=	void fraction of dispersed-phase
$h_{p,i}$	=	specific enthalpy of particle i
L_v	=	latent heat of vaporization
M	=	molecular-weight of gas
M_k	=	molecular-weight of species k
M_l	=	molecular-weight of liquid species
$m_{p,i}$	=	mass of particle i
$\dot{m}_{\text{ss},i}$	=	spherically-symmetric vaporization rate for particle i
$\dot{m}_{p,i}$	=	vaporization rate of particle i
N	=	total number of simulated particles
N_s	=	total number of gas-phase species
$N_{p,i}$	=	number of virtual particles for simulation particle i
$\text{Nu}_{p,i}$	=	Nusselt number of particle i
P	=	gas pressure
$P_{l,\text{sat}}$	=	saturation pressure of liquid
Pr	=	Prandtl number of gas
$Q_{p,i}$	=	heat flux to particle i
$\mathbf{q}$	=	heat flux vector
R	=	ideal gas constant
$\text{Re}_{p,i}$	=	Reynolds number of particle i
$\dot{s}_e$	=	energy source from dispersed-phase
$\dot{s}_k$	=	species k source from dispersed-phase
$\dot{\mathbf{s}}_m$	=	momentum source from dispersed-phase
T	=	gas temperature
T_{bn}	=	normal boiling temperature
$T_{p,i}$	=	temperature of particle i
$T_{\text{rel},i}$	=	relative temperature between gas and particle i
T_s	=	temperature at the surface of particle i
V_i	=	volume of particle i
$\mathbf{v}$	=	gas velocity
$\mathbf{v}_{d,k}$	=	diffusion velocity of species k
$\mathbf{v}_{p,i}$	=	velocity of particle i
W_m	=	interpolation weight for cell m
$\dot{w}_k$	=	production rate of species k from reactions
$\mathbf{x}_{p,i}$	=	location of particle i

y_c	=	conservative state vector
$X_{k,s}$	=	mole fraction of species k at particle surface
Y_k	=	mass fraction of species k
Y_l	=	far-field mass fraction of vaporized liquid
$Y_{l,s}$	=	mass fraction of vaporized liquid at surface of particle
$\Delta h_{\text{rxn},j}$	=	heat of reaction for reaction j
ΔV_m	=	volume of computational cell m
β	=	vaporization coefficient
λ	=	gas thermal conductivity
μ	=	gas viscosity
ρ	=	gas density
ρ_s	=	spray density
$\tilde{\rho}_s$	=	non-dimensional spray density or mass loading
τ	=	viscous stress tensor
ϕ	=	local equivalence ratio
$\dot{\omega}_j$	=	reaction rate for reaction j

This page intentionally left blank.

EXECUTIVE SUMMARY

This report details the development of detonation spray models using the computational fluid dynamic (CFD) codes developed in the Laboratories for Computational Physics and Fluid Dynamics (LCP) at the U.S. Naval Research Laboratory (NRL). Previous work at LCP on sprays using particle tracking methods are adapted to the current CFD codes, specifically the DASLIB libraries and applications (used for much of the Rotating Detonation Engine work done at NRL) and the JENRE code, originally developed for jet-noise simulations, but recently extended for reacting, moderate and high-speed flow fields. DASLIB uses Flux-Corrected Transport (FCT) algorithms developed at LCP and used extensively for previous detonation work. The JENRE code, however, uses a discontinuous-Galerkin (DG) formulation and considerably different algorithms and numerics, and has capabilities that are not currently in the FCT-based codes, some of which are impossible to implement in the older algorithms. Initial results compare a series of shock tube simulations for glass particle and water droplet simulations presented in earlier work. A similar set of one-dimensional and two-dimensional simulations are done with the new codes, and show that the new DASLIB applications can match previous results almost exactly, while the JENRE code also compares very well with the previous work. The current simulations are extended by showing more highly resolved two-dimensional results, and improved thermodynamics for the water droplet cases. The current simulations also extend the previous work by considering a detonation tube with JP-10 droplets and gaseous oxygen. The gas-phase stoichiometric JP-10/oxygen detonation matches the calculated C-J detonation velocity at 2300 m/s, while the liquid JP-10/oxygen detonation with 10 μm droplets results in a detonation velocity of 2170 m/s. Two-dimensional liquid JP-10/oxygen simulations show the formation of detonation cells, and also the persistence of fuel vapor far behind the detonation wave.

This page intentionally left blank.

LIQUID-FUELED DETONATION MODELING AT THE U.S. NAVAL RESEARCH LABORATORY

1. INTRODUCTION

Engines based on the higher thermodynamic efficiency of the detonation cycle have long been an attractive target for advanced propulsion concepts, but the unsteady nature of the detonation wave along with high pressure pulses, heat transfer rates, and highly non-linear detonation wave dynamics have made realizing the potential of the detonation cycle challenging. Over the last decade, there has been increasing interest in rotating detonation engines (RDEs), which currently hold the most promise to realize the potential of the detonation cycle[1]. There have been an extensive number of experimental, numerical, and theoretical studies on RDEs. For numerical studies, both high fidelity[2,3,4] and low fidelity[5,6] simulations have provided insight into the thermodynamic cycle, efficiency, and operation of both idealized RDEs and experimental rigs and propulsion devices. More simplified reduced-order-models[7,8] have also been generated for design tools to aid the development of RDEs for propulsion. All of these methods have contributed significantly to the understanding of RDEs in different ways.

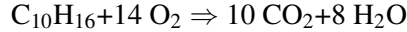
Much of that research has been focused on gaseous fuels, however, practical RDEs will have to use liquid fuels if they are to be competitive with current gas turbines. Liquid fuels create a lot of complexity that is not present even in the high fidelity models mentioned above. In addition to the spray injection parameters, droplet vaporization in the turbulent environment in the fill region, breakup and burning in the detonation wave, and burning in the deflagration zone are all important processes. Previous numerical work on spray detonations has been done in the context of pulsed detonation engines [9,10] and general multi-phase detonations [11], but there is almost no related work on RDEs. Some experimental work has, however, shown that liquid-fueled RDEs are possible [12].

This report details the development of detonation spray simulations using the computational fluid dynamic (CFD) codes developed at the Laboratories for Computational Physics and Fluid Dynamics (LCP) at the U.S. Naval Research Laboratory (NRL). Previous work at LCP on sprays using particle tracking methods [13] are adapted to the current CFD codes, specifically the DASLIB libraries and applications (used for much of the Rotating Detonation Engine work done at NRL) and the JENRE code, originally developed for jet-noise simulations, but recently extended for reacting, moderate and high-speed flow fields. Simulations from the previous work are computed using the new codes to ensure that the particle models have been incorporated correctly. Specifically, one and two-dimensional glass particle simulations without vaporization and one-dimensional water droplet simulations are compared with the original work. In addition, a two-dimensional water droplet simulation is done to look at the effect of vaporization on the development of the flow-field. Finally, detonation calculations were accomplished for a spray JP-10/oxygen using

conditions obtained from [9]. This work will form the foundation for doing more detailed, high-fidelity calculations as well as for developing reduced-order spray models that can be used for full engine calculations.

2. MULTI-PHASE MODEL

Before discussing the conservation equations, we specify the regime of the multi-phase flow. For these simulations, the volume fraction is assumed to be high enough to have two-way coupling, but low enough that there are no particle-particle interactions. We also assume that the amount of volume taken up by the particles is small enough to ignore nozzling effects. To check this, we consider a typical volume fraction. Since we are interested in JP-10/oxygen detonation, we consider JP-10 droplets in an oxygen environment. For stable detonation, the equivalence ratio, ϕ , needs to be $\mathcal{O}(1)$. JP-10 is simply $C_{10}H_{16}$, so the reaction is the following:



For a stoichiometric mixture where $\phi = 1$ (meaning all of the fuel and oxidizer are consumed), there are 14 O_2 molecules for every JP-10 molecule. On a mass basis, this becomes 136.2 g JP-10/447.9 g O_2 , or 0.3041 g JP-10/g O_2 . At 298.15 K and 1 atm, the density of an O_2 gas mixture is $\rho_{O_2} = 1.308 \times 10^{-3}$ g/cm³, whereas the density of JP-10 liquid is $\rho_l = 0.932$ g/cm³. Therefore, the volume fraction of JP-10 is $f_v = (\rho_{O_2}/\rho_{JP-10})(F/O)_{\text{stoich}} = 0.0004267$, meaning that only 0.04267% of the volume is taken up by droplets, well within the dispersed-phase range. For air mixtures, the volume fraction will be even smaller.

2.1 Gas-Phase Conservation Equations

The gas-phase conservation equations follow the viscous Navier-Stokes equations for a compressible gas:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{v} = \dot{s}_\rho \quad (1)$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot \rho \mathbf{v} \otimes \mathbf{v} = -\nabla P + \nabla \cdot \mathbb{T} + \dot{\mathbf{s}}_m \quad (2)$$

$$\frac{\partial \rho e_t}{\partial t} + \nabla \cdot (\rho e_t + P) \mathbf{v} = \nabla \cdot \mathbb{T} \cdot \mathbf{v} - \nabla \cdot \mathbf{q} + \rho \mathbf{g} \cdot \mathbf{v} + \sum \dot{\omega}_j \Delta h_{\text{rxn},j} + \dot{s}_e \quad (3)$$

$$\frac{\partial c_k}{\partial t} + \nabla \cdot c_k \mathbf{v} = -\nabla \cdot c_k \mathbf{v}_{d,k} + \dot{w}_k + \dot{s}_k \quad (4)$$

Here, the conservative set of variables is $y_c = (\rho, \rho \mathbf{v}, \rho e_t, c_k)^T$, reaction source terms are specified as $\dot{\omega}_j$ for each reaction j and species reaction term $\dot{w}_k$. The density is related to the concentrations by $\rho = \sum_i c_i M_i$, where M_i is the molecular weight of species i . By introducing a specific heat

of reaction term $\Delta h_{\text{rxn},j}$ we can shift species enthalpies by a constant without affecting the fluid dynamics. This is demonstrated in Appendix A. The spray source terms are specified as $\dot{s}_\rho$, $\dot{\mathbf{s}}_m$, $\dot{s}_e$, and $\dot{s}_k$ for the mass, momentum, energy, and species conservation equations respectively. The exact form of these sources will be discussed below. To close this system of equations, an equation of state relating the conservative set of variables to pressure is required, $P = f(y_c)$. The exact form of this function varies depending on the problem, and will be described in the solution procedure.

Here, the viscous terms are included in the viscous stress tensor, $\mathbb{T}$, the heat flux vector, $\mathbf{q}$, and the species diffusion velocity, $\mathbf{v}_{d,k}$. For the shock and detonation tubes in this report, these terms are negligible compared to the convective, reactive, and spray terms. On the particle level, however, the viscous terms are critical, therefore, we typically specify a viscosity μ and thermal conductivity λ as a function of temperature and mixture composition.

Note that the gas-phase equations do not have any volume fraction terms, because we are making the dispersed-phase approximation and the volume fractions are typically very low.

2.2 Particle-Tracking Equations

For these simulations, the dispersed-phase is computed using a Lagrangian particle-tracking procedure. The particle is characterized by its mass $m_{p,i}$, diameter $D_{p,i}$, temperature $T_{p,i}$, droplet enthalpy $h_{p,i}$, location $\mathbf{x}_{p,i}$ and velocity $\mathbf{v}_{p,i}$. The single-particle equation of motion is based on a simple inertial balance equation $m\mathbf{a} = \sum \mathbf{F}$:

$$m_{p,i}\mathbf{a}_{p,i} = m_{p,i}\frac{d\mathbf{v}_{p,i}}{dt} = \mathbf{F}_{d,i} + \mathbf{F}_{g,i} \quad (5)$$

where $\mathbf{F}_{d,i}$ is the aerodynamic drag, and $\mathbf{F}_{g,i}$ is the buoyancy force on particle i . The particle equation (5) can be integrated twice to determine the velocity $\mathbf{v}_{p,i}$ and location $\mathbf{x}_{p,i}$ of the particle during the simulation. In addition to the inertia of the particle, we also have heat transfer to and from the particle. This can be written as an energy conservation equation in terms of the particle specific enthalpy, $h_{p,i}$:

$$m_{p,i}\frac{dh_{p,i}}{dt} = Q_{p,i} \quad (6)$$

For this work, the buoyancy force is negligible. The Reynolds number for the particle $\text{Re}_{p,i}$ is written in terms of a relative or slip velocity, $\mathbf{v}_{\text{rel},i} = (\mathbf{v} - \mathbf{v}_{p,i})$, the diameter of the particle, $D_{p,i}$, and the gas-phase viscosity and density at the droplet surface, μ/ρ . The drag term, $\mathbf{F}_{d,i}$ is written in terms of the Stokes drag and a high-velocity correction term:

$$\mathbf{F}_{d,i} = 3\pi D_{p,i}\mu(\mathbf{v} - \mathbf{v}_{p,i})f_c(\text{Re}_{p,i}) \quad (7)$$

The high velocity correction term, $f_c(\text{Re}_{d,i})$ is the ratio of the actual coefficient of drag to the Stokes drag coefficient, $C_{d,\text{St}} = 24/\text{Re}_{d,i}$ and has been the subject of extensive research. The simplest high velocity correction term is the Schiller and Naumann law [14], expressed as:

$$f_c(\text{Re}_{d,i}) = (1 + 0.15\text{Re}_{p,i}^{0.687}) \quad (8)$$

This correlation works well for Reynolds numbers under 800. For particles on the order of one micron, the Reynolds number will always be well below 800. A number of other drag laws were evaluated in Schwer [13], and the Tedeschi *et al.* [15] drag law was applied up to a Mach number of 1:

$$f(\text{Re}_{p,i}, \text{Ma}_{p,i}) = (1 + 0.15 \text{Re}_{d,i}^{0.687}) \left[1 + \frac{\text{Re}_{p,i}^2}{\text{Re}_{p,i}^2 + 100} \exp\left(\frac{-0.225}{\text{Ma}_{p,i}^{2.5}}\right) \right] \quad (9)$$

The JENRE code uses the drag law of Abraham[16], which is described as:

$$f(\text{Re}_{p,i}) = \left(1 + 0.1104 \sqrt{\text{Re}_{p,i}}\right)^2 \quad (10)$$

The heat flux to the particle, $Q_{p,i}$, is defined by the relative temperature, $T_{\text{rel},i} = T - T_{p,i}$, thermal conductivity, λ , and a Nusselt number, $\text{Nu}_{p,i}$, which is itself dependent on the Reynolds number and Prandtl number, $\text{Pr} = \mu C_p / \lambda$. The heat transfer term for particles is:

$$Q_{p,i} = \pi D_{p,i} \lambda \text{Nu}_{p,i} T_{\text{rel},i} \quad (11)$$

The Nusselt number, $\text{Nu}_{p,i}$ is usually given by a non-dimensional expression. For this work, we use a slightly modified Ranz and Marshall correlation [17] as used in Sivier [18]. It is given as:

$$\text{Nu}_{p,i} = [2 + 0.459 \text{Pr}^{1/3} \text{Re}_{p,i}^{0.55}] \quad (12)$$

For this report, we also make the assumption that the particle or droplet is at a uniform temperature, in which case the enthalpy of the particle, $h_{p,i}$ can be directly related to the particle temperature $T_{p,i}$, and is usually given in terms of a curve-fit similar to the gaseous enthalpies.

2.2.1 Droplet Vaporization

In cases where the particles are liquid and vaporizing, additional terms must be computed, especially for mass and heat transfer. The particle inertial equation (5) remains the same. We now have an additional equation to determine the amount of mass vaporized from the droplet surface, and the corresponding reduction in diameter of the particle. The vaporization model used for all of the liquid simulations is based on spherically symmetric isolated droplets. Convective effects are specified by modifications based on empirical correlations. The model assumes the vapor pressure at the surface of the droplet is the saturation pressure for the liquid at the local surface temperature, T_s . In the cases of infinite conductivity considered here, $T_s = T_{p,i}$, and assuming a unity Lewis number, the vaporization coefficient, β , is calculated as:

$$\beta = \frac{8\lambda}{\rho_l C_p} \ln|B + 1| \quad (13)$$

where C_p is the gas-phase specific heat, and B is the Spalding transfer number defined as:

$$B = \frac{Y_l - Y_{l,s}}{Y_{l,s} - 1} \quad (14)$$

where Y_l is the mass-fraction of the *gas*-phase vaporizing species in the computational cell (considered the “far-field” in the analysis), $Y_{l,s}$ is the mass-fraction at the surface of the particle and is calculated based on the partial pressure of the liquid,

$$Y_{l,s} = \frac{P_{l,\text{sat}}}{P} \frac{M_l}{M} \quad (15)$$

where $P_{l,\text{sat}}$ is the saturation pressure of the liquid at the surface temperature T_s , P is the far-field pressure (also felt at the surface of the droplet), M_l is the molecular weight of the vaporizing species, and M is the overall molecular weight of the gases at the liquid surface, $M = \sum X_{k,s} M_k / \sum X_{k,s}$.

Assuming the D^2 -law, the spherically-symmetric mass vaporization rate for a single particle is given by the expression:

$$\dot{m}_{\text{ss},i} = \rho_l \frac{dV}{dt} = \rho_l \frac{\pi D_{p,i}}{4} \frac{dD_{p,i}^2}{dt} = -\rho_l \frac{\pi D_{p,i} \beta}{4} \quad (16)$$

The spherically-symmetric vaporization term is then corrected using the correlation of Ranz and Marshall [17] to determine the convective vaporization rate,

$$\dot{m}_{p,i} = \dot{m}_{\text{ss},i} [1 + 0.3 \text{Pr}^{1/3} \text{Re}_{p,i}^{1/2}] \quad (17)$$

Similarly, the Nusselt number defined in equation (12) is multiplied by a vaporization factor:

$$\text{Nu}_{p,i} = [2 + 0.459 \text{Pr}^{1/3} \text{Re}_{p,i}^{0.55}] \left[\frac{\ln|1+B|}{B} \right] \quad (18)$$

In addition to the change in Nusselt number due to vaporization (thus changing the heat flux to the particle), the particle energy must also account for vaporization. That is, the change in enthalpy of a droplet is the heat flux to the droplet minus energy loss due to vaporization.

$$m_{p,i} \frac{dh_{p,i}}{dt} = Q_{p,i} - \dot{m}_{p,i} L_v \quad (19)$$

2.2.2 Saturation Pressure Calculation

One of the key terms in the above analysis is the saturation pressure of the liquid at the droplet surface, $P_{l,\text{sat}}$. The saturation pressure is typically a function of the latent heat of vaporization and the temperature at the surface. In many cases, this thermodynamic property can be supplied as a curve-fit through the Antoine equation, which has the form:

$$\ln P = A_{\text{Ant}} - \frac{B_{\text{Ant}}}{C_{\text{Ant}} + T} \quad (20)$$

where A_{Ant} , B_{Ant} , and C_{Ant} are constants. $\exp(A_{\text{Ant}})$ is in pressure units (typically atm or Pa), and B_{Ant} and C_{Ant} are in Kelvins. For accurate vapor pressures over the entire range of temperatures, typically one Antoine curve-fit is not appropriate and the saturation curve is broken into two smaller curve-fits separated by the normal boiling point, T_{bn} (the temperature where the saturation pressure is one atm).

If these curve-fits are not supplied, but the enthalpy curve-fits are known for both the liquid and the gas-phase, the saturation pressure can be generated using the Clausius-Clapeyron equation. In differential form, this equation is

$$\frac{dP}{dT} = \frac{PL_v}{T^2} \quad (21)$$

Here, L_v is the latent heat of vaporization, and is defined as the difference in specific enthalpy for the gas and liquid phase, $L_v(T) = h_g(T) - h_l(T)$. In the special case of a constant latent heat of vaporization, L_v , the Clausius Clapeyron can be integrated to get an expression for the saturation pressure:

$$\ln P_2 - \ln P_1 = -\frac{L_v}{R} \left(\frac{1}{T_2} - \frac{1}{T_1} \right) \quad (22)$$

Note that in the general case over a wide temperature range, this equation is not appropriate to use. Also, the Clausius-Clapeyron equation assumes that the gas-phase is an ideal gas. As the temperatures and pressures approach the critical point, this is definitely not appropriate.

2.2.3 Droplet Breakup

For this memo-report, droplet breakup is not considered. However, it is definitely an important mechanism in shocked spray flows and is planned to be addressed in the near future. Some preliminary work was done in Ref. 13, and we plan to build on this work for our future simulations.

2.2.4 Volume Fraction of the Dispersed-Phase

For these simulations, the number of particles simulated is calculated from the size distribution of the particles and the spray or particle density. The spray density, ρ_s , is calculated as the density of the particles ρ_l multiplied by the volume fraction, f_v , of the particles. The volume fraction, in turn, is simply the ratio of all the particle volumes to the computational volume. The spray or particle density is often given in terms of mass loading, which is defined as $\tilde{\rho}_s = \rho_s/\rho$. For dispersed-space flows with two-way coupling, the volume fraction of particles is assumed to be negligible compared to the gas volume, as is typically on the order of 0.01 or less, and the mass loading tends to be $\mathcal{O}(1)$.

From the spray density, we can compute the number of particles that we need to have for our simulation. For one and two-dimensional simulations, this can be tricky, as we will demonstrate. For a single computational cell m the spray density is written as:

$$\rho_s = \rho_l f_v = \rho_l \sum_{i=1}^{N_m} V_i / \Delta V_m \quad (23)$$

where N_m are the number of particles in cell m and ΔV_m is the volume of cell m . The volume of each particle, assumed to be a sphere, is simply $V_i = \pi D_i^3/6$. The volume of the cell needs to be in units of $[L^3]$. Typically what is done in one-dimensional and two-dimensional codes is that the unresolved dimensions are assumed to be unit length. For the special case of mono-dispersed seeding, where all the particles have the same diameter, $D_i = D$, we compute the number of particles per cell as follows:

$$N_m = \rho(\rho_s/\rho)6\Delta V_m/D^3 = \rho(\rho_s/\rho)6A\Delta x/(\rho_l\pi D^3) \quad (24)$$

where A_m is the cross-sectional area of cell m , and is typically simply 1 $[L^2]$. But this poses a problem for determining the number of particles in our simulation. As an example, the first simulation we do is seeded with glass particles that have a mean diameter of $D = 26 \mu\text{m}$, distributed over region of 476 cm with a mass loading of $\tilde{\rho}_s = 0.63$. Glass has a density of $\rho_l = 2.5 \text{ g/cm}^3$. The domain is divided into cells that are 0.2 cm long each having a cross-sectional area of 1 cm^2 . For the simulation, this results in 6,462 particles per cell, and a total of 15.4 million particles. For a one-dimensional simulation, this is far too many particles to track, and is unnecessary. If we had set the cross sectional area to $A_m = 0.2^2 \text{ cm}^2$, making the cells a perfect cube, the number would be far less, a more reasonable 258 particles/cell and 615 thousand total particles. Even this is quite a costly simulation, and unnecessary. For MKS units, the situation is far worse, requiring 64.6 million particles/cell, and a total of 154×10^9 particles for the simulation!

We use the concept of virtual particles to address this issue. The idea is that each simulated particle represents a specified number of virtual particles, which all affect the gas-phase in the same way and are in a similar location. This number, $N_{p,i}$, appears in the volume fraction calculation and also in the gas-phase source terms. This may appear arbitrary, but so is the cross sectional area of the simulation. Numerically, the critical parameter affecting the gas-phase solution and spray statistics is the number of particles per cell. As we found in our previous work [13], the total number of simulated particles does not need to be very large to give smooth gas-phase results. Particle statistics (such as volume fraction), on the other hand, do require a good number more particles). For this simulation, we set $N_{p,i} = 2000$, which will result in about 3.2 particles/cell, and around 7700 total particles for the glass simulations with mass load of $\tilde{\rho}_s = 0.63$. This produces noisy particle statistics, but fairly smooth gas-phase solutions. The resultant particle loading equation becomes:

$$\rho_s = \rho_l f_v = \rho_l \sum_{i=1}^{N_m} N_{p,i} V_i / \Delta V_m \quad (25)$$

2.3 Spray Gas-Phase Source Terms

The dispersed-phase flows considered here have a sufficient number of particles that the dispersed-phase has a considerable effect on the gas-phase flow. In the simplest case, each particle contributes to the gas-phase spray source terms ($\dot{s}_p$, $\dot{s}_m$, $\dot{s}_e$, and $\dot{s}_k$) only in the cell where it is located. This results in generally a very noisy source function to the gas-phase, so to create a smoother source function many programs will distribute the source terms over several cells. For any given particle i , we distribute the source over M cells with a weighting function W_m , where $\sum_{m=1}^M W_m = 1$. The source

functions are derived from the particle equations specified above, scaled by a number of virtual particles per simulated particle, $N_{p,i}$.

$$\dot{s}_\rho = \sum_{i=1}^N \sum_{m=1}^M \frac{W_m}{\Delta V_m} N_{p,i} \dot{m}_{p,i} \quad (26)$$

$$\dot{\mathbf{s}}_m = - \sum_{i=1}^N \sum_{m=1}^M \frac{W_m}{\Delta V_m} N_{p,i} [\mathbf{a}_{p,i}/m_{p,i} - \dot{m}_{p,i} \mathbf{v}_{p,i}] \quad (27)$$

$$\dot{s}_e = - \sum_{i=1}^N \sum_{m=1}^M \frac{W_m}{\Delta V_m} N_{p,i} \left[Q_{p,i} + \mathbf{a}_{p,i} \cdot \mathbf{v}_{p,i}/m_{p,i} - \dot{m}_{p,i} \left(h_{p,i} + L_v + \frac{1}{2} \mathbf{v}_{p,i} \cdot \mathbf{v}_{p,i} \right) \right] \quad (28)$$

$$\dot{s}_k = \sum_{i=1}^N \sum_{m=1}^M \frac{W_m}{\Delta V_m} N_{p,i} \dot{m}_{p,i} \quad (29)$$

where N is the total number of simulated particles, and M is the number of weights. $N_{p,i}$ represents the number of virtual particles per simulated particle, and is simply a multiplier for the gas-phase source terms and volume fraction terms. For one-dimensional problems, this is typically two, although this is treated somewhat differently in DASLIB and JENRE. ΔV_m is the volume of each of the cells over which the particle source is being distributed.

The last term in both the energy and momentum equations represent the fact that mass is added to the gas-phase at the droplet velocity and droplet temperature, since $h_{p,i} + L_v$ is simply the gas-phase enthalpy at the droplet temperature.

3. SOLUTION PROCEDURE

3.1 DASLIB

DASLIB is written in C as a library of different structures and their associated functions. It uses a preprocessor to incorporate single-inheritance and simple type generics to reduce code duplication, and is broken into a utility library, physical property library, CFD library, parallel CFD library, and a solutions library. Applications are then constructed for the different solutions to different types of problems. The DASLIB applications use many of the same algorithms that have been developed in house at LCP&FD over the last thirty years and used extensively for both basic and applied research in high speed and detonation flow-fields. For this report, we use two specific applications, the `das_shocktube` and the `das_detonate` applications. Both programs use the core parallel CFD library and utility functions, but incorporate different equation of states and different state vectors (`das_detonate` includes species conservation equations that are unnecessary for non-reacting shock tubes).

3.1.1 Gas-Phase Solution Procedure

For the gas-phase, the FCT-algorithm of Boris and Book [19] is used, which is especially suited for high-speed flow. The version of the algorithm used in the DASLIB applications is described in detail in NRL/MR/6410-93-7192 [20]. The algorithm is based on a very efficient one-dimensional generalized conservation solver. Operator splitting is used for multi-dimensional simulations. The applications also use domain decomposition for distributing the gas-phase solution over several MPI processes[21]. This method is very efficient and scalable for explicit time-integration techniques such as the FCT algorithm. Complex geometry is included using the Virtual-Cell-Embedding (VCE) technique of Landsberg and Boris [22], although it is not used for simulations in this report.

3.1.2 Particle-Tracking Solution Procedure

For the tracked particles, the 4th-order Adams-Bashforth and Adams-Molton predictor-corrector schemes are used for the temporal integration [13]. The vaporization, drag, and heat transfer terms are all evaluated explicitly. For the simulations, we assume all the mass, momentum, and energy transfer occurs between the droplet and the cell in which the droplet originated at the start of the time-step. This was done to simplify the parallel algorithm, but results in a high level of noise in the computation. The particles are distributed over different processes dependent on the gas-phase domain decomposition. No specific load balancing is done for the particle simulations, with the result that the simulations can become very poorly balanced due to particles being pushed to one side. Future work will discuss how to distribute the source term appropriately, and how to more evenly load balance these simulations.

3.1.3 *das_shocktube* Application

The *das_shocktube* application was written to simulate flow-fields with one species (or a uniform mixture) that is calorically perfect. To close the system of equations presented in Section 2, these assumptions results in following relation between the conserved state, y_c , and pressure, P :

$$\rho e_t = \frac{P}{\gamma - 1} + \frac{1}{2}\rho v^2 \quad (30)$$

For the particle tracking drag and heating terms, we use the Sutherland expression for expressing the gas viscosity as a function of local temperature,

$$\mu(T) = 1.458 \times 10^{-5} \left(\frac{T^{3/2}}{110 + T} \right) \quad (31)$$

We assume the Prandtl number is 0.75 to calculate the thermal conductivity of the gas. Because we do not track species, we cannot do droplet vaporization using this application, and use the *das_detonate* code instead, described below.

3.1.4 *das_detonate* Application

The *das_detonate* application was written to simulate reacting flow-fields with multiple species and chemical reactions. It solves additional conservation equations for each of the species, and can handle global reaction mechanisms, detailed kinetic mechanisms, and induction-time parameter models.

$$\rho e_t = \sum_{i=1}^{N_s} c_i h_i(T) - P + \frac{1}{2} \rho v^2 \quad (32)$$

The species enthalpies $h_i(T)$ are based on curve-fits. *das_detonate* uses 6th-order curve-fits for the specific heat developed from NASA thermodynamic curve-fits. For the DASLIB programs, the enthalpies are shifted before beginning the simulation so that the internal energy is 0 at $T = 0$ K, assuming a constant specific heat below T_{ref} . The procedure for shifting the coefficients is outlined in Appendix A. Because the energy only appears as a derivative in the conservation equation, shifting the species enthalpies by a constant does not change the solutions. This shift, however, does come into play when computing source terms and vaporization terms. In that case, one needs to account for the shift as a heat-release term (for equations) and as a heat of vaporization term for evaporation.

The *das_detonate* application also includes several induction-time parameter models (IPM's) for computing shock-driven reacting flow-fields such as detonations. IPM's require the solution of an additional conserved scalar, τ . τ represents the accumulation of radicals after a shock-wave passes. Once enough radicals accumulate, when $\tau > 1$, heat-release begins.

$$\frac{\partial \rho \tau}{\partial t} + \nabla \cdot \rho \tau \mathbf{v} = \rho / t_{ind}(T, P) \quad (33)$$

The induction-time function, $t_{ind}(T, P)$ can be computed either through a correlation using experimental data, or it can be found by doing a set of numerical calculations with a detailed kinetic mechanism. Work has been done to expand this to considering different equivalence ratios for hydrogen/air, and also different fuel mixtures using a detailed hydrocarbon kinetic model [23, 24].

Heat-release is treated by converting reactant to product species, and releasing the appropriate amount of heat to recover the C-J detonation velocity. For models with curve-fit thermodynamics, we recover the C-J velocity by presuming that the product species is a mixture of the C-J products computed from an equilibrium code [25]. The characteristic heat-release time is typically not critical for these computations, so long as it is fast enough to release the majority of heat before the C-J point. This is discussed in more detail in Ref. 9.

3.2 JENRE

The second code in which we are implementing this dispersed-phase model is the JENRE code developed in house at the U.S. Naval Research Laboratory. JENRE implements all of these capabilities in C++ with extensive use of templates, so that the various physics, numerics, and mesh representation capabilities are implemented in a generic way, but specific models are then combined at compile-time in order to ensure high efficiency. One of the overarching goals of

JENRE has been designed to achieve the highest computational performance possible while maintaining the flexibility to simulate geometrically and physically complex flow-fields. To this end, JENRE is designed to exploit the parallelism inherent in modern computer architectures at multiple levels, including coarse-grained, distributed memory parallelism, as well as fine-grained, shared-memory parallelism, so that it runs efficiently on the highest performance computing architectures available, such as GPU clusters. Current simulations are done mainly on a GPU cluster available at NRL.

The first form of parallelism achieved by JENRE is coarse-grained, distributed-memory parallelism, which is implemented through domain decomposition with inter-processor communication performed using MPI[26]. In addition, Propel provides parallel IO support using MPI-IO and on-the-fly parallel partitioning using the open-source ParMETIS library[27].

The second form of parallelism achieved by JENRE is fine-grained, shared-memory parallelism. Such parallelism is implemented using the open-source Thrust library[28,29] which provides generic algorithms (e.g., copy, transform, reduce, compress, prefix-sum, sort), function objects, and iterators, which are used as a foundation upon which the mesh and flow solver operations in JENRE are implemented. Since the Thrust library is optimized for either a serial, OpenMP, Thread-Building-Blocks (TBB), or CUDA backend, Propel can be compiled to produce either an optimized serial CPU code, multi-core CPU code, or many-core GPU code, and could easily be extended to future computational architectures that might emerge. This approach greatly reduces the time-consuming process of rewriting specialized, low-level CUDA or OpenCL code.

A discontinuous-Galerkin based framework has recently been developed for JENRE to replace the previous continuous-Galerkin FEM discretization. This version of JENRE implements discontinuous-Galerkin finite elements discretizations of the Navier-Stokes equations together with Hartmann shock-capturing [30]. Although both codes share the same design philosophy and many core utilities, the numerics for JENRE-DG are completely different from JENRE-CG, and so the solutions are not interchangeable except at the most basic level. The shock capturing ensures monotonicity at shocks and sharp features. The DG representation uses separate polynomial representations of the solution for each cell and is allowed to be discontinuous between cells. Facial interface conditions are used to link the cells together. This provides for a compact representation of the conservation equations for higher order representations, and a straight-forward path for generating these higher order representations. This is especially important at boundaries where higher order accuracy is retained without any special treatments of cells near the boundary. The DG method also supports jumps across shock-waves exactly as long as the shock-wave is aligned with the grid cells. Matched with moving grids and space-time representations, this intriguing possibility obviates the need for shock-capturing terms, and is currently an area of active research at NRL[31].

For this paper, we use stationary grids and an explicit Runge-Kutta time-marching algorithm, making use of shock capturing to handle discontinuities that appear within cells. Our experience with the JENRE-DG framework has shown this to produce superior results, even with stationary grids and shock capturing, and it provides a clear path forward. We focus on p1 and p2 representations of the solution, however, we have the capability to have much higher order representations.

3.2.1 Particle-Tracking Solution Procedure

JENRE utilizes an N -stage Runge-Kutta time advancement scheme to track the evolution of the particle temperature, mass, and position. For most of the calculations presented in this report,

we have chosen $N = 4$, which results in a scheme that is fourth-order accurate in time. Particle contributions to the source terms in the gas-phase conservation equations are accumulated on a per-cell basis. In the current implementation, the source-term contributions are not distributed among neighboring cells. For practical applications, we have observed that the resulting noise in the source term fields can be handled naturally using the DG representation; however, methods for higher-order reconstruction of the source term fields are under development that will be able to fully exploit the underlying high-order DG spatial discretizations.

4. SHOCK-TUBE GEOMETRY AND CONDITIONS

The shock tube geometry aims to match the results from Ref. 13 on comparing particle-tracking and sectional approaches for shock driven flows. This geometry is for a 1D shock tube and is originally based on experiments of Sommerfeld[32]. The total length of the shock tube is 781 cm with a driver section of 200 cm, a driven section without particles of 105 cm, and a seeded driven section of 476 cm. The unseeded driven section allows for the shockwave to develop before interacting with the particles. The driven section, unseeded and seeded, is at 1 atm and 300 K. The Driver section conditions vary as a range of 6 different cases discussed in Schwer[13], and are shown in Table 1. The cell size is set to 0.2 cm to give us good resolution for the shock wave velocity.

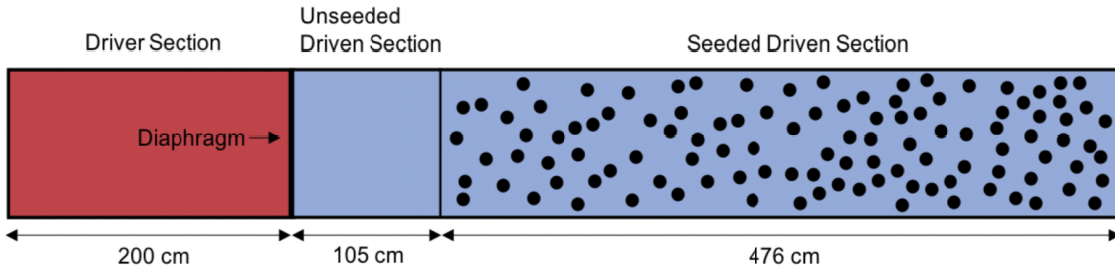


Figure 1 — One-dimensional shock tube geometry.

The two-dimensional simulations have a similar shock tube setup with some notable differences. For the 2D cases the shock tube height is set to 100 cm, while both the driver section and seeded driven section are only half the total height of the shock tube at 50 cm. This introduces both axial and longitudinal shocks, and creates a complex flow through the shock tube. As with the 1D simulations, the driven section is 1 atm and 300 K, and the driver section conditions are shown in Table 1. The 2D geometry is shown in Figure 2.

The six sets of driver conditions used in the original simulations [13] are shown below. Condition 1 corresponds to experimental conditions from Sommerfeld [32]. These conditions were used for both the glass particle and water droplet simulations. The detonation simulations used a different detonation tube and conditions and are described in that section.

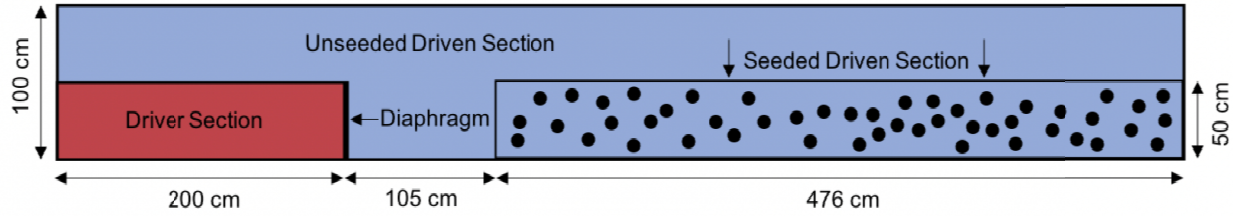


Figure 2 — Two-dimensional shock tube geometry.

The first simulations accomplished examined the shock tube solution without any particles. This ensured that both the DASLIB and JENRE codes were consistent with no differences in the gas models. The main difference between the solutions is some noise generated in the DASLIB solution that is not in the JENRE solution, although there also appears to be slightly more dissipation in the JENRE solution. Overall the codes compared well with each other. Figure 3 shows the pressure and temperature solution using DASLIB and JENRE without particles for the Case #4.

Case Number	P_2/P_1	ρ_2/ρ_1	T_2/T_1	Shock Wave Mach Number
1	4	1	4	1.49
2	5.655	1	5.655	1.70
3	8	1	8	1.96
4	16	2	8	2.42
5	32	4	8	2.95
6	64	8	8	3.54

Table 1 — Driver conditions for each shock tube case as stated in Schwer[13]. Driven section: $P_1 = 1$ atm, $T_1 = 300$ K, air for both driver and driven sections with $\gamma = 1.4$.

5. GLASS PARTICLE SIMULATIONS

The next set of simulations contained a the seeded section with glass particles that are $26\ \mu\text{m}$ in diameter and have a mass-loading of $\tilde{\rho}_s = 0.63$. Since no vaporization takes place, we use a simple perfect gas assumption for the air and a constant specific heat for the glass. The density of glass is $\rho_l = 2.5\ \text{g/cm}^3$, and the specific heat of glass is $C_l = 766 \times 10^4\ \text{ergs/g}$. DASLIB computations are done with the `das_shocktube` application.

5.1 One-Dimensional Simulations with Particles

For this series of simulations, we are focused on comparisons with earlier results for the two codes and less on the physics of shock wave attenuation using glass particles and water droplets. The interested reader is referred to earlier reports for the physical mechanisms behind the shock wave particle interactions. For a more detailed discussion of the physics, please consult [13].

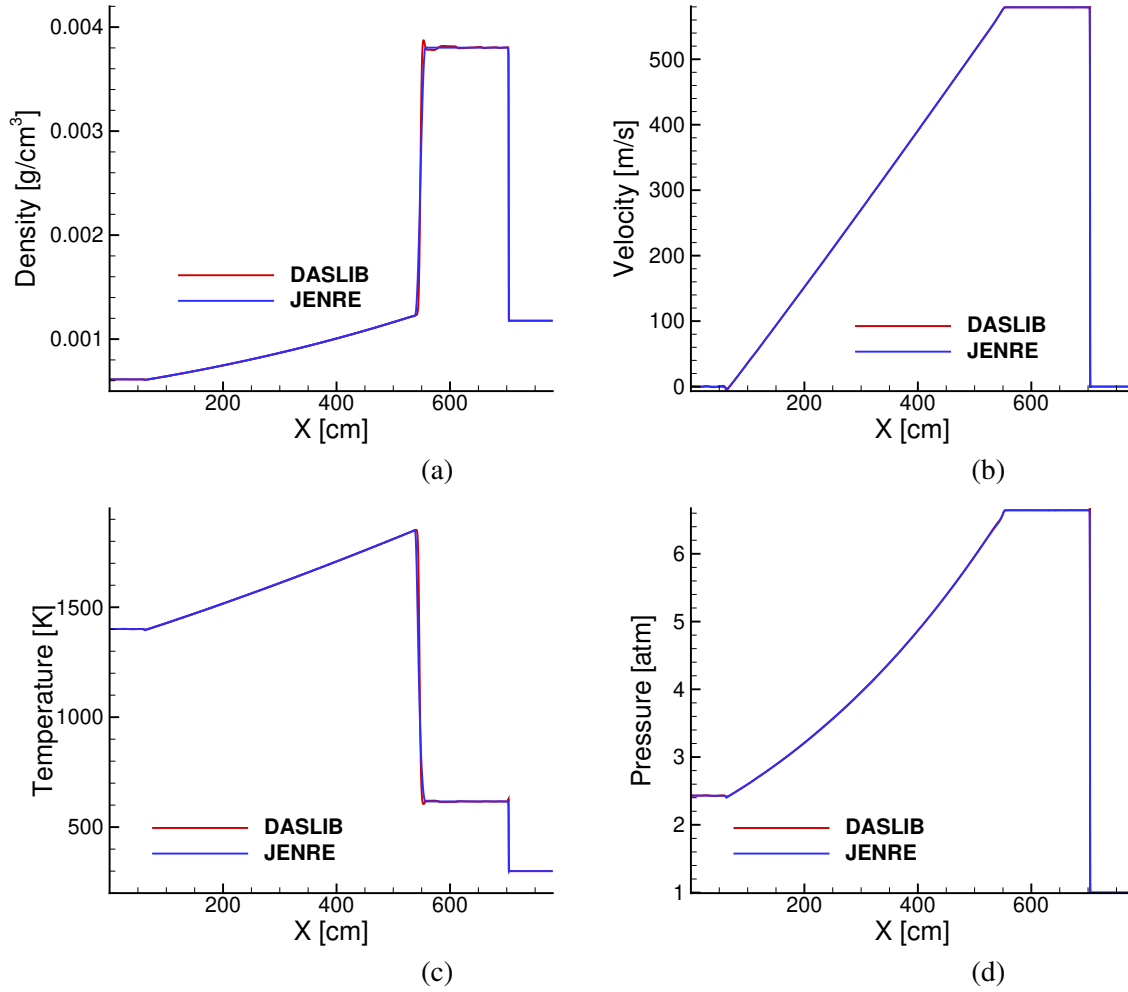


Figure 3 — Comparison between Daslib and Jenre codes for Case 4, Mach 2.42, with no particles.

Besides the basic algorithms discussed above in Section 3, there are some other differences between the DASLIB and JENRE codes that have an effect on the solution. The largest of these is the way the particles are initialized. For the DASLIB programs, particles are specified over a geometric range (ellipses, boxes, etc.), a spray density loading ρ_s (in dimensional form) for the entire geometric range, and a number of size bins, each having a specific range of sizes. The total mass of the dispersed-phase is calculated, and then particles are added to the flow based on a random number generator. The location can be anywhere within the specified spray geometry. Another random number is used to determine the size bin for the droplet, and finally another for the actual size within that bin. The mass of the particle (along with its virtual particles) is added to the total particle mass. Particles are added until the total particle mass is equal to the specified particle mass. This tends to generate a lot of variation within the spray field, resulting in (for instance), a noisy volume fraction compared with uniform particle placement, but the hope is that this generality will be beneficial when computing more complex spray fields. An example of a particle specification is given below:

```

Initialize-particles
  Number-bins          5
  Number-virtual-particles 2000
  Diameters            5e-4 10e-4 15e-4 20e-4 25e-4 30e-4
  Volume-distribution   0.05 0.15 0.35 0.40 0.05
  Mass-loading          0.0007371
  Box
    Location           543 25
    Size                476 50
  End
End

```

In this example, we have five bins and droplet sizes ranging from 5 μm to 30 μm . The particles are distributed into the 5 bins based on the volume-distribution, which adds up to 1. The total spray density for all of the particles is $\rho_s = 0.0007371 \text{ g/cm}^3$.

JENRE uses a different method entirely, where the number of particles per cell (with fractions) is computed over a geometric range. Particles are added on a cell by cell basis, with a random number generator determining the location of the particle within the cell. For cases where the number of particles per cell is not an integral number (probably a majority of cases), JENRE ensures that a fraction of the cells have an additional particle so that the total mass of the particles matches the desired specified mass-loading. For instance, if it was determined that there were 3.3 particles per cell, all cells would have at least 3 particles, and 30% of the cells would have a fourth particle. An example of that section of a configuration file is shown below:

```

group1.configuration
  cell_based_random
    diameter      27e-4
    temperature    300.0
    particles_per_virtual 2000
    cad intersection plane origin (305.0 0.0)
                                normal -1
                                plane origin (0.0 50.0)
                                normal (0.0 1.0)
    mass_loading 0.63
  end
end

```

The results are shown in Figure 4. The simulations were run both with and without particle heating. Surprisingly, adding particle heating substantially changes the solution, drawing more energy out of the shocked gases and further slowing the shock wave down. The simulations were run with 2000 virtual particles, and resulted in 6800 particles in DASLIB, and 7200 in JENRE, which amounts to about 2.85 particles/cell. With such a low number of particles per cell, the volume fraction becomes quite noisy for the DASLIB programs. This does not appear to be a problem with the JENRE code, which is the result of the different initialization schemes. Since the DASLIB codes and the Ref. 13 codes use essentially the same algorithms, we expect an almost perfect match between the two solutions, as is seen in Figure 4. JENRE results also are quite close to the solution in Ref. 13, but differences in the algorithm and small differences in the drag and heat transfer laws result in differences in the solution.

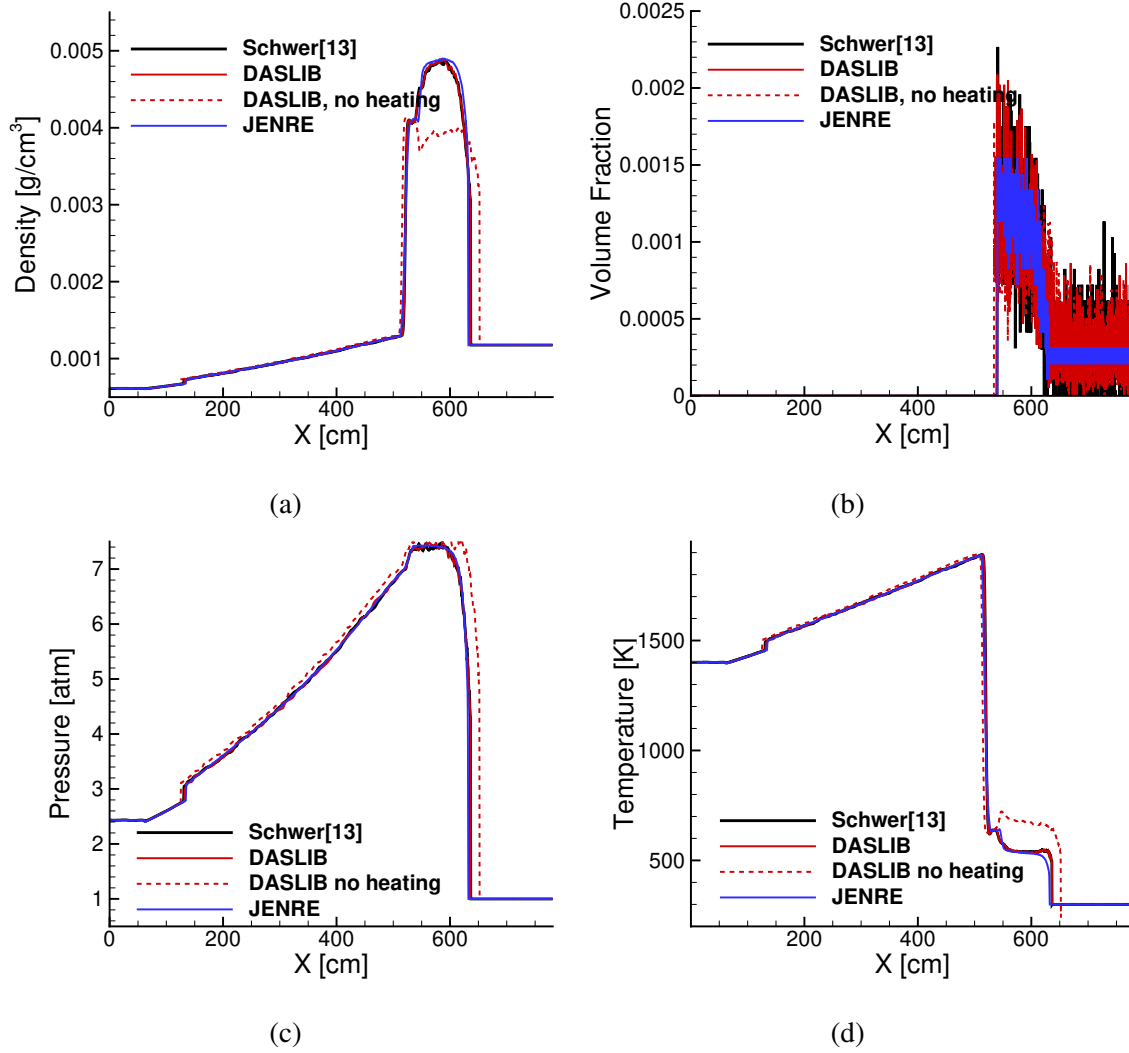


Figure 4 — Comparison of Case 4, Mach 2.42 driver conditions, 6 ms into the simulation with (solid) and without (dashed) particle heating.

5.2 Two-Dimensional Simulations with Particles

The final set of glass-particle simulations are a series of two-dimensional cases. Cases 1, 4, and 5 were run with the two-dimensional setup and compared with the original results in Ref. 13. The original simulations had a resolution of 0.5 cm in both the x and y direction, while the newer simulations kept the 0.2 cm resolution from the 1D simulations. Both sets of simulations have particle-heating enabled. The 2D cases had substantially more particles, with about 340,000 simulated particles for DASLIB and JENRE, due to the much larger geometry (now 100 cm in the y direction instead of just a unit length). Figure 5 shows the original solution from Ref. 13 for Case 4, while Figure 6 shows the same case with DASLIB and the finer resolution. All the same features appear in Figure 6 that are present in Figure 5, and several additional features. Because of the finer resolution in the gas-phase, but same number of particles for the dispersed-phase, the

noise from the particle solution does appear in the gas-phase solution. A more reasonable number of particles/cell, from 10-20, or a more uniform distribution such as the one in JENRE is needed to get a less noisy gas-phase solution. In spite of that, the results show a clear match between the original work in Ref. 13 and the current codes.

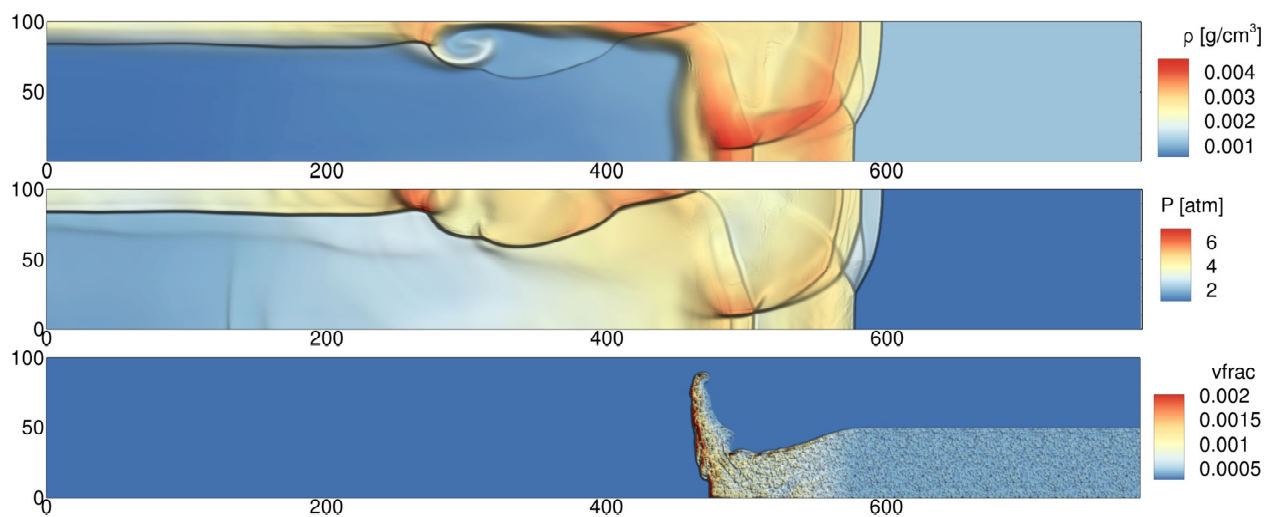


Figure 5 — Schwer[13] two-dimensional simulation for Case 4 at 6 ms. Resolution is 0.5 cm.

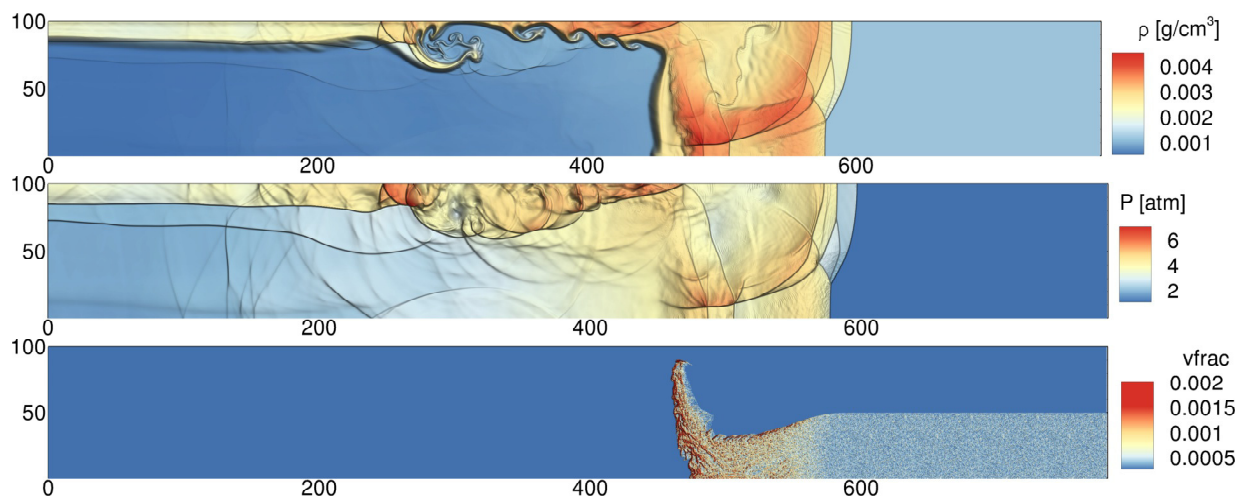


Figure 6 — DASLIB two-dimensional simulation for Case 4 at 6 ms. Resolution is 0.2 cm.

Figure 7 shows the DASLIB result for Case 5, again with the 0.2 cm resolution. Similar to the result in Ref. 13 (not shown), we see the development of a void region, where particles have been pushed from the center to other regions. For more discussion on the development of this zone, please refer to [13]. For both cases 4 and 5, the flow field predicted using JENRE was nearly identical to the DASLIB flow fields, and, for the sake of brevity, they are not shown in the figures.

These results show that the particle-tracking has been correctly implemented into both the DASLIB and JENRE codes.

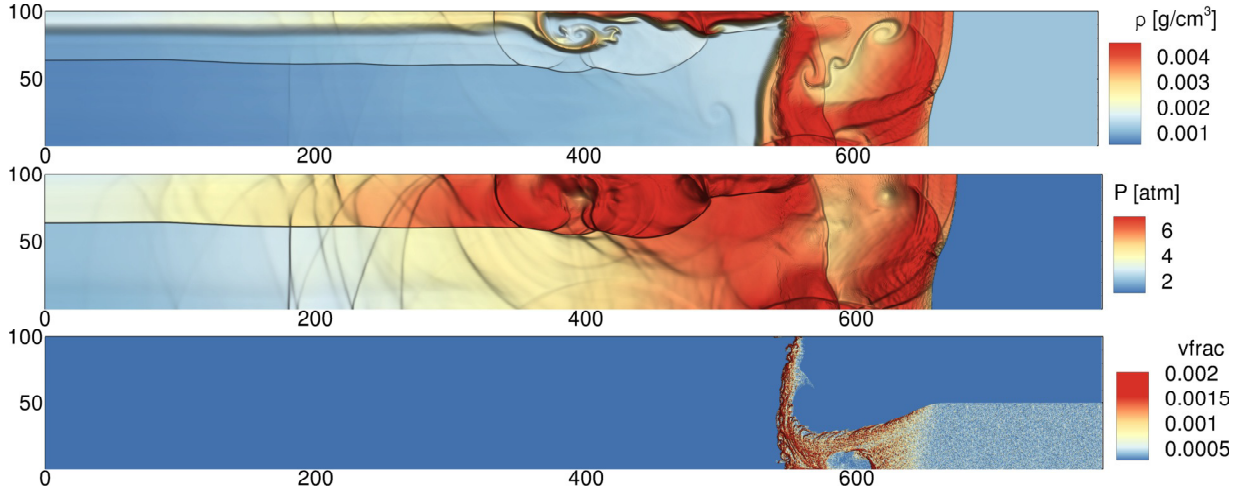


Figure 7 — DASLIB two-dimensional simulation for Case 5 at 6 ms. Resolution is 0.2 cm.

6. WATER DROPLET SIMULATIONS

The next set of simulations use the same geometry as in Figure 1, but the glass particles are replaced by water droplets. The water droplets vary from 25-30 μm and the mass loading is $\tilde{\rho}_s = 0.50$. In this case, vaporization does take place, so for DASLIB we use the multi-species `das_detonate` application to run the simulation. As before, we compare with previous work in Schwer [13]. In the original simulations, we used constant specific heats for each of the species, where $\gamma_{\text{air}} = 1.4$, $\gamma_{\text{steam}} = 1.327$, $C_{l,\text{water}} = 4.189 \times 10^7$ ergs/g, and the latent heat of vaporization at 298.15 K is $L_v = 2.3049 \times 10^{10}$ ergs/g. The viscosity and thermal conductivity are computed as above in equation (29). The vapor pressure is computed using the Antoine equation, equation (20), where the coefficients are $A_{\text{Ant}} = 11.70327$, $B_{\text{Ant}} = 3835.83$ K, and $C_{\text{Ant}} = -45$ K. The units for $\exp(A_{\text{Ant}})$ are atm. These coefficients produce an error of less than 2% over most of the data compiled in *Steam Tables* [33]. Near the critical temperature (647 K) this error increases slightly to under 5%. We also do computations using temperature dependent curve-fits for each of the species. The curve-fits are given in Appendix B.

In Figure 8, the saturation curve computed from the Antoine curve-fit above is compared with the saturation curve from the *Steam Tables*. We divide the curve into two plots, one for temperatures below the normal boiling temperature, and one for temperatures above T_{bn} . We also compare both the differential Clasius Clapeyron (eqn. 21) with the algebraic Clasius Clapeyron (eqn. 22) methods, where the latent heat of vaporization is calculated from the curve-fits at the temperature of interest. Results show that all four methods do fairly well with calculating saturation pressures below the normal boiling temperature. The algebraic Clasius Clapeyron equation accumulates a significant error at very low pressures due to changes in the latent heat of vaporization far away from T_{bn} , but otherwise does a remarkable job, indicating that the latent heat of vaporization does not vary radically at the lower temperatures. At the higher end, however, we see significant errors

in the Clasius-Clapeyron equation, whereas the Antoine curve-fit remains very accurate. Real gas effects cause the differential Clasius Clapeyron equation to be off substantially as we approach the critical point, while the large variation in latent heat of vaporization causes the algebraic Clasius Clapeyron to be off by an even larger amount. For the results for this paper, the temperatures for the droplets remains fairly low, so all the different methods should work fairly well.

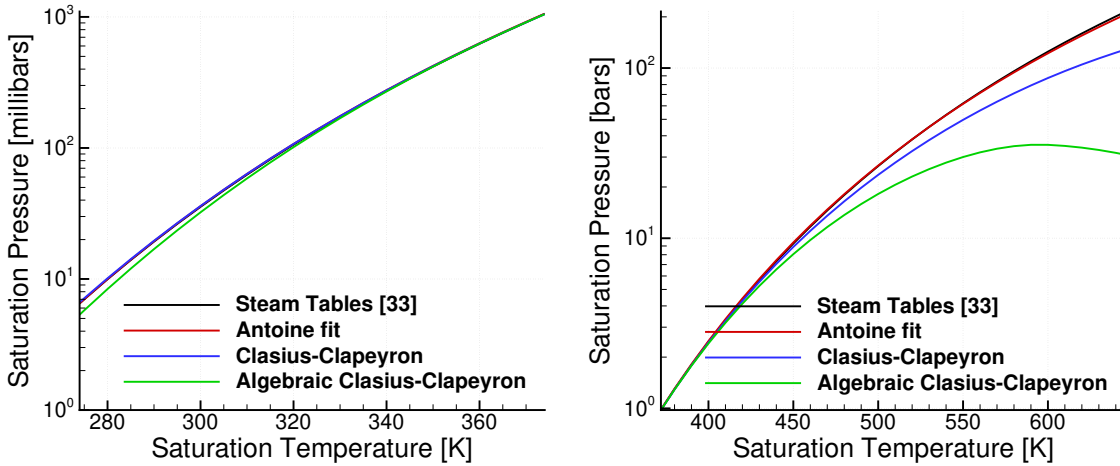


Figure 8 — Saturation pressure as function of temperature for water and steam.

Results for Case 4, 6 ms into the simulation is shown in Figure 9. Here we specify 1000 virtual particles per simulated particle, with the result that the number of simulated particles is 25,200, much more than the glass particles (both because the number of virtual particles is less and water density is less than the glass density). Comparisons are made with the original work[13], JENRE results, and DASLIB results using constant specific heats (as did the original work) and curve-fit specific heats. As with the previous glass particle results, we see that the DASLIB constant specific heat and the previous results are almost an exact match, while the JENRE results are very close to it and the variable specific heat is the outlier. Using the variable specific heat has the interesting effect of making the shock wave attenuation weaker and establishing higher temperatures in the driver gases. The reason for this is that specific heats tend to increase with higher temperatures, therefore, the driver energy associated with the variable specific heat case is higher than for the constant specific heat case (because we set the temperature of the driver), making the initial push stronger and the driver gases sustaining higher temperatures and pressures. This is born out in Figure 9. Vaporization of the water spray brings the temperature down in the shocked gases by about the same level. Ultimately, the shock waves are not strong enough to produce very different behavior between the constant and variable specific heat cases, especially within the shocked driven gases.

In addition to the 1D studies conducted in Ref. 13, we also did two-dimensional simulations for Cases 4 and 5, similar to the glass particle two-dimensional simulations. These are shown in Figure 10. The simulations look remarkably similar to Figure 7 with glass particles, with some differences. Vaporization tends to have two effects. It increases the density and pressure of the

gas-phase locally, and also decreases the temperature locally. Both the density and the temperature flood plots are interesting, as we can clearly see the shape of the particle cloud super-imposed on the temperature. As with the glass particle simulations, the flow fields predicted for the droplet simulations using JENRE match extremely well with those predicted using DASLIB.

7. DETONATION SIMULATIONS

The final set of simulations done for this report are detonation tube calculations with JP-10 droplets and oxygen. This was selected due to previous work by Cheatham and Kailasanath [9] in computing liquid droplet detonations. We use the `das_detonate` code, which has both vaporization and also gas-phase reaction models. The reaction is calculated using an induction-time parameter model. For the thermodynamics, the codes use curve-fits for oxygen (O_2), JP-10 ($C_{10}H_{16}$), and C-J detonation products, found by running the equilibrium CEA code [25]. The curve-fits are given in Appendix C.3.2. The induction-time parameter model uses a correlation from Davidson *et. al.* [34]:

$$t_{\text{ind}} = 3.47 \times 10^{-15} [O_2]^{-1.27} [JP\ 10]^{0.67} \exp[54,000/R_u T] \quad (34)$$

Because we are using the curve-fit thermodynamics, the heat-release for the detonation is not a parameter, and is simply found based on the enthalpy shifts in Appendix A. For the saturation pressure, we use the Antoine coefficients found in [35], where $A_{\text{Ant}} = 8.173667$, $B_{\text{Ant}} = 2783.0$ K, and $C_{\text{Ant}} = -117.037$ K. The characteristic heat-release time we use for this simulations is $\Delta t_{\text{hrst}} = 2 \times 10^{-7}$ s.

The saturation curve for JP-10 is shown in Figure 11. Here we compare the curve-fit defined above with the differential and algebraic Clasius Clapeyron curves. Similar to the water curves in Fig. 8, there is also wide variation between the Antoine equation and both Clasius Clapeyron relations, for similar reasons as for the wide variations in the water saturation curves. What is more surprising is that those variations persist for the lower end of the saturation curve, below the normal boiling temperature. Unlike the upper end of the saturation curve, the lower end should not be effected by real gas effects, and so the main differences are the variations in the latent heat of vaporization, or relatively poor fits for the Antoine equation. These differences should be addressed in further research, to get a more accurate and consistent representation of the saturation pressure.

For this work, because the droplets are being subjected to detonation waves, the droplets will most likely heat up rapidly to somewhere close to the normal boiling temperature, so the curve-fits should work okay.

7.1 Detonation Tube and Conditions

The detonation tube is very similar to the shock tubes computed above. The tube itself is much shorter, only 150 cm. The driver section is much smaller as well, only 1 cm wide, and the driver conditions are also more extreme, set at 50 atm and 3500 K. Immediately adjacent to the driver is a 29 cm driven section with a gaseous JP-10/oxygen mixture at an equivalence ratio of $\phi = 1$, followed by a 120 cm driven section with liquid JP-10 and oxygen. The gaseous section produces a stable detonation wave which then interacts with the liquid/gaseous fuel section. These simulations

use a finer resolution of 0.025 cm to better capture the induction zone. The resolution is still too course for the gaseous detonation, but captures the liquid-spray detonation induction zone fairly well. The geometry is shown in Figure 12.

7.2 Detonation Simulations

The first simulation completed for JP-10/oxygen was completely gas-phase. The simulation was done in order to verify that the gas-phase thermodynamics for the reactants and product are correct and recover the correct C-J wave speed, temperature, and pressure. We also wanted to verify that the driver conditions produced a strong enough shock wave to initiate the detonation. Results from this calculation have a wave-speed of 2300 m/s, which is extremely close to the correct C-J detonation wave-speed. The C-J pressure and temperature are more difficult to measure, because it requires knowing exactly where the C-J point is, however, approximately they are at $T_{CJ}/T_1 = 12.8$ and $P_{CJ}/P_1 = 36.4$, which are close to the CEA results of 13.0 and 39.1. The one-dimensional form of the detonation is shown in Figure 13 as the dashed line.

From the gas-phase detonation calculation we attempted a droplet-phase detonation. We filled the seeded region of the detonation tube with liquid JP-10 droplets. The droplets all have a diameter of 10-11 μm . The mass-loading is set to $\tilde{\rho}_s = 0.294$, which gives an equivalence ratio of $\phi = 1$ once all the droplets are evaporated. The number of virtual particles is again set to 2000, resulting in 83,500 particles (18 particles/cell). Pre-vaporization is minimal because the vapor pressure of JP-10 at 300 K is very low. The simulation does result in a detonation wave, where the pressure and temperatures are similar to the gaseous detonation and the detonation wave velocity is lower than the gas-phase C-J detonation. The detonation wave velocity is 2170 m/s. The induction region is much larger for the liquid-phase detonation, because the droplets now must be vaporized before creating the pool of radicals necessary to begin heat-release. The pressure spike at the start of the detonation wave (called the von Neumann spike) is also much larger and higher than for the pure gas-phase detonation wave, most likely due to vaporization adding to the density and pressure as it did with the water droplet simulations. The solutions are also noisier, which is an artifact of the particle-tracking spray computation.

Figure 14 shows the maximum pressure for each cell during the entire simulation for the liquid spray detonation and the gas detonation. These results show that there is a temporary failure of the detonation-wave after the wave hits the first droplets. It then slowly re-organizes and re-initiates, becoming stable by about 60 or 70 cm in the detonation tube (30 or 40 cm from the start of the liquid droplet region). The resultant maximum pressure is now much higher than the gas-phase as well.

Two dimensional simulations were also accomplished for the JP-10/oxygen spray detonation. These simulations used the same one-dimensional geometry, but extended the y direction by 10 cm, expecting fairly large detonation cells based on the induction zones found in the one-dimensional simulation. Resolution in the y direction was kept at 0.025 cm, resulting in a grid with 1.92 million cells. For the two-dimensional simulations we used only 10 virtual particles per simulated particle, results in about 40 droplets in each cell. This is a substantial calculation, but well within the limits of even small current HPC systems. The result for the solution are shown in Figure 15 and 16. Figure 15 shows the solution at the end of the computation, whereas Figure 16 shows the maximum pressure for each cell over the entire simulation. There are two main points to make from

these simulations. Interesting, we see cellular structures appear that are very similar to gaseous detonations cellular structure, except much larger. The second observation is that, although the overall equivalence ratio is $\phi = 1$, therefore all the JP10 and oxygen should be consumed, we still see a substantial amount of JP10 behind the detonation. There are a few reasons why this could be happening. The most likely reason is that the droplets are being pushed around in such a way that when they vaporize, they create regions that are fuel rich, and other regions that are oxygen rich, resulting in incomplete combustion. This would also cause the reduction in the detonation wave velocity as well.

To confirm this, we did a computation at a lean condition, $\phi = 0.77$. In Figure 17, we show the JP-10 density as well as the resulting cellular structure from the maximum pressure. We see the same type of behavior, where we have pockets of fuel persisting far behind the detonation wave itself, even though the detonation wave itself is fairly stable. The cellular structure becomes stretched out compared to the stoichiometric case due to the reduced heat-release, and the detonation wave velocity goes from 2170 m/s to 2030 m/s.

These demonstration calculations are not intended to provide enough detail to understand how spray detonations differ from gaseous detonations, but they do clearly show that there are some differences. Further research in this area will help better define those differences and how they affect the ability of RDEs to use liquid fuels.

8. CONCLUSIONS

This report detailed the implementation of particle tracking methods into two newer code bases at LCP&FD, the DASLIB and JENRE codes. The DASLIB libraries and applications are based on algorithms developed at LCP&FD over the last thirty years to look at a wide range of high speed flow-fields, including both basic detonation and detonation engine research. The JENRE code was developed recently to simulate jet noise, but has been extended to consider reacting flows and a range of other complex flow-fields. The implementations in both of these solvers were verified by comparing with older work done at LCP&FD. A series of one-dimensional and two-dimensional shock tube simulations with both glass particles and water droplets were computed and compared. The DASLIB codes were able to replicate the older results almost exactly, while the JENRE results were also very close, giving us confidence that the particle tracking is correctly implemented in both code bases.

One dimensional and two dimensional simulations of a detonation tube seeded with JP-10 were then conducted with the DASLIB code using an induction-time parameter model. We updated the thermodynamics for the JP-10 from earlier work, and used an Antoine equation to determine the saturation pressure based on surface droplet temperatures. Initial simulations were done with the gas-phase in vapor form to ensure the gas-phase thermodynamics and induction-time parameter model were properly working. This simulation gave us a detonation velocity of 2300 m/s, very close to the 2299 m/s from C-J analysis using similar thermodynamics. Additional one-dimensional simulations were run with JP-10 droplets, dropping the detonation wave velocity to 2170 m/s but otherwise producing a very similar flow-field. The final set of simulations were two-dimensional detonation simulations, and showed the formation of large cell like structures very similar to what is seen in gas-phase detonations. One striking difference is the presence of fuel gas downstream of

the detonation wave itself. This is present in fairly large amounts for stoichiometric JP-10/oxygen computations, and also to a much smaller extent for lean mixtures. This implies that a percentage of the JP-10 is not at conditions for quick combustion, and decouple from the shock front, causing a slow down in the detonation wave velocity.

A surprising aspect of this work is that even with the minimal models that are described here, we can still do spray detonation calculations. The results for the JP-10/oxygen detonation are very similar to the results from Cheatham [9], and we would expect to see similar trends as we varied the different parameters. The current modeling effort does not include breakup phenomena, although we expect that process to have a large effect on the detonability of a spray mixture over a range of operating conditions. It also does not include the more advanced drag or heat transfer models, nor have we included film effects or non-uniform droplet temperature effects. One of the primary reasons we have hesitated from drawing too many conclusions from the above detonation calculations is that the different sub-models need to be more fully explored to determine their importance for detonation calculations, before drawing conclusions while neglecting them.

One area that is clearly problematic is the saturation pressure curves for the liquid fuel. If we do not have an Antoine curve-fit but do have fairly good thermodynamics for the gas and liquid, we can approximate this curve using the differential Clausius-Clapeyron equation. The current curve-fits and thermodynamics used for this work show that there is some disagreement between the two for JP-10, and this may be an issue for other fuels as well. A better representation for this is needed since the saturation pressure plays such a critical role in the vaporization.

Further research will be focused on addressing these issues with the models, and extending our knowledge of liquid spray detonations to understand how they differ from gaseous detonations and under what conditions they are stable. At that point, we can begin to use these models to develop reduced order models for detonation engine calculations.

ACKNOWLEDGMENTS

This work is supported by the Office of Naval Research through NRL 6.1 Computational Physics Task Area.

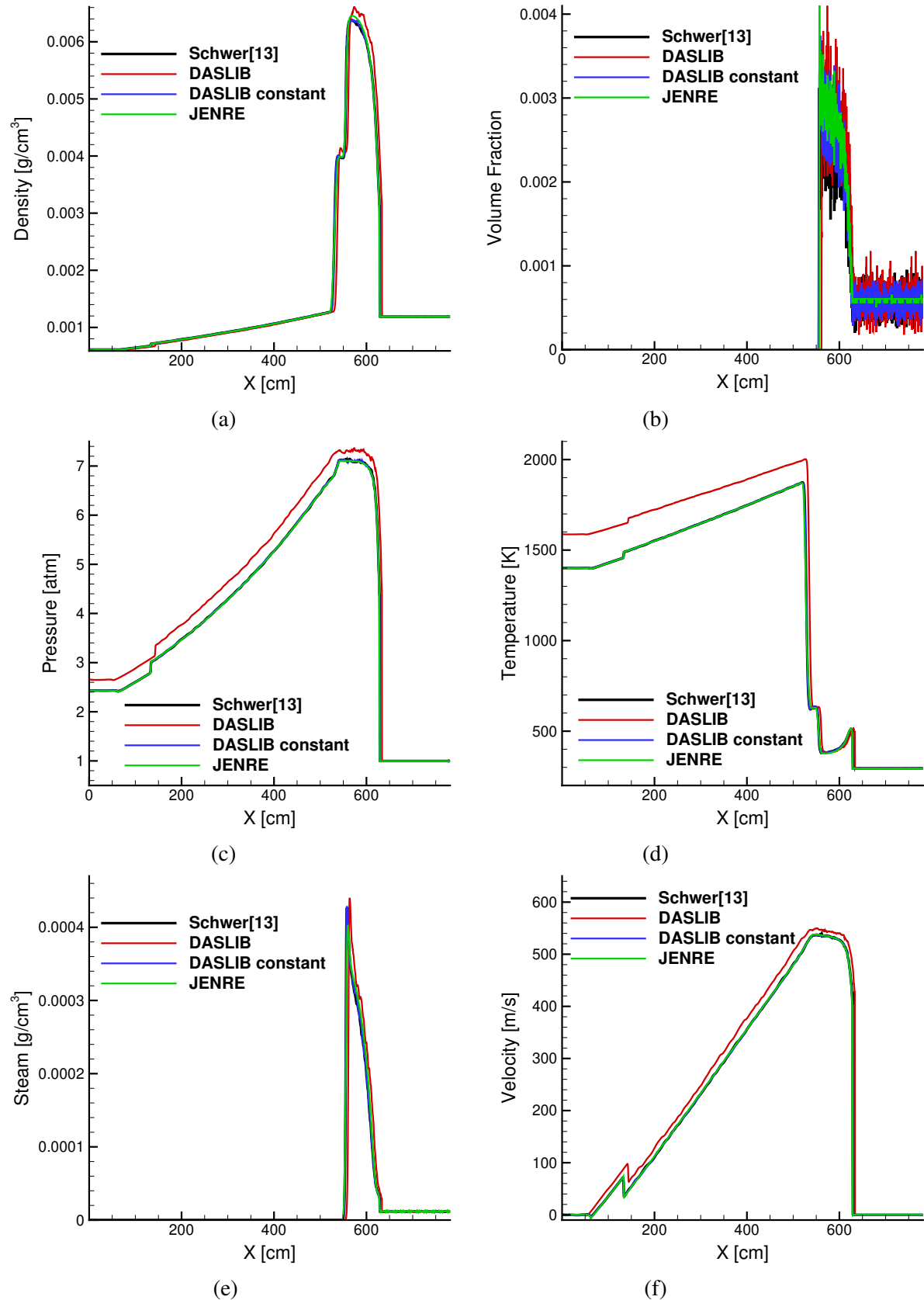


Figure 9 — Comparison of Case 4, Mach 2.42 driver conditions, 6 ms into the simulation.

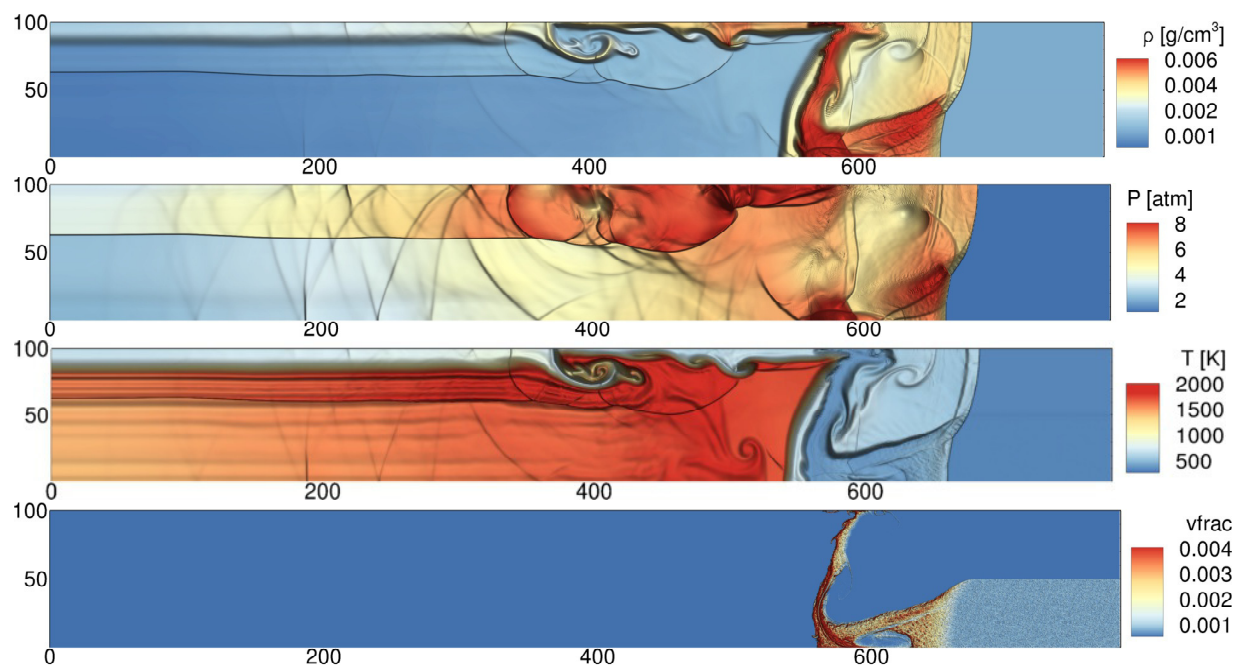


Figure 10 — Two dimensional simulation of water droplets for Case 5, 6 ms into the simulation.

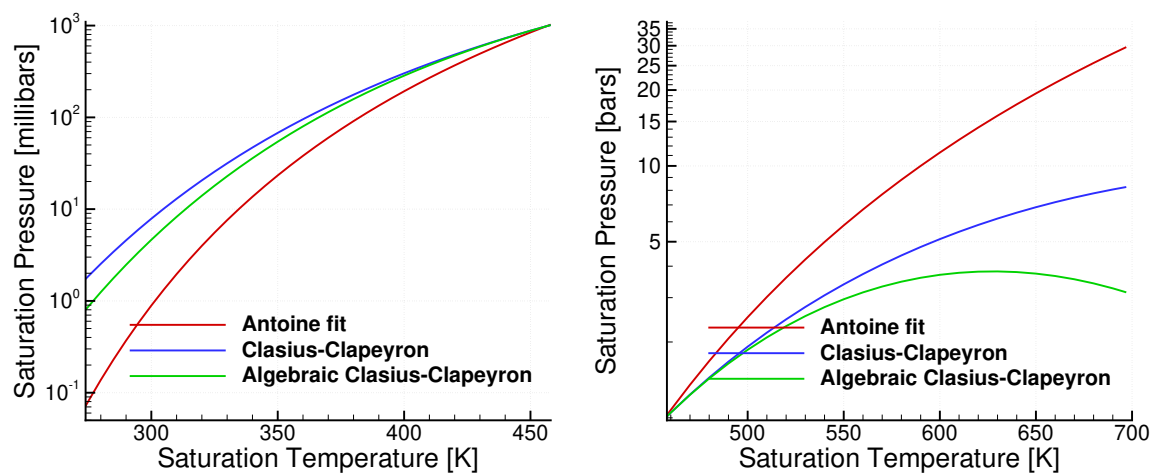


Figure 11 — Saturation pressure as a function of temperature for JP-10.

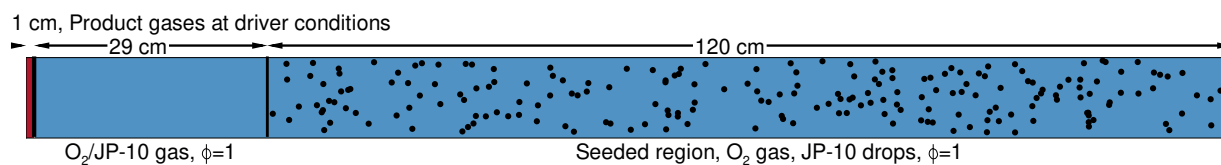


Figure 12 — Geometry of the JP-10 liquid detonation tube.

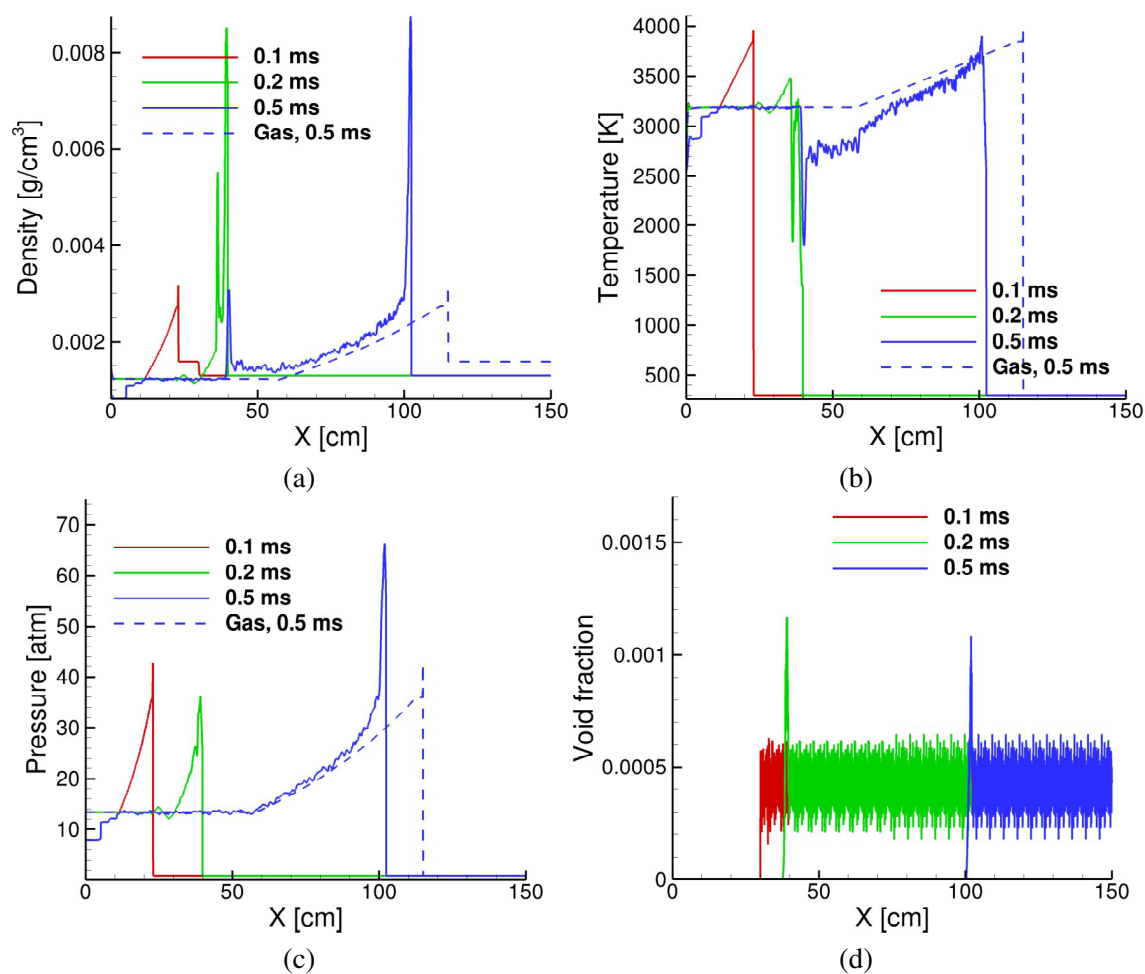


Figure 13 — Comparison of gaseous and liquid-droplet JP-10/oxygen detonation simulations.

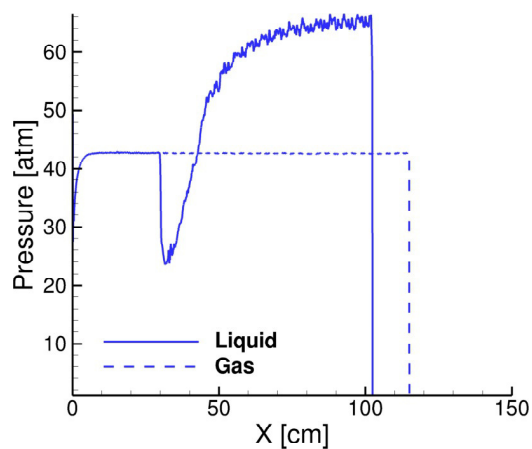


Figure 14 — The maximum pressure for each cell at the end of the simulation.

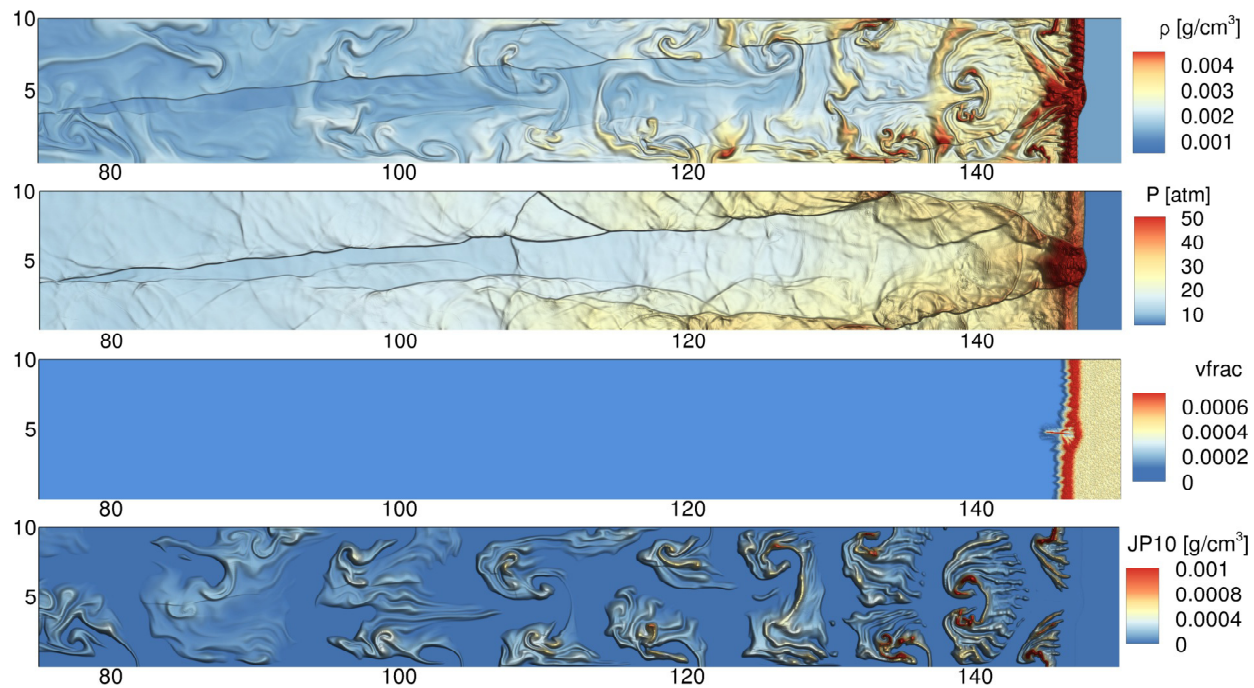


Figure 15 — Two-dimensional liquid spray JP-10/oxygen detonation simulation.

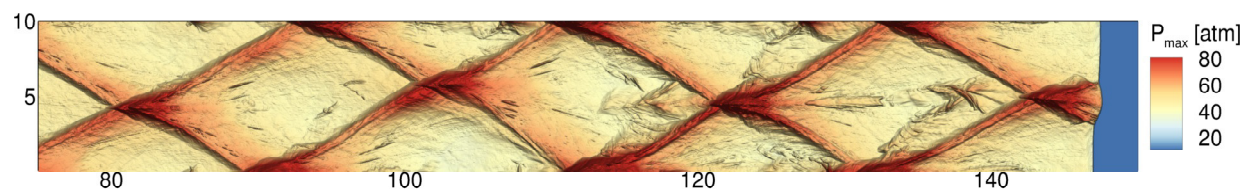
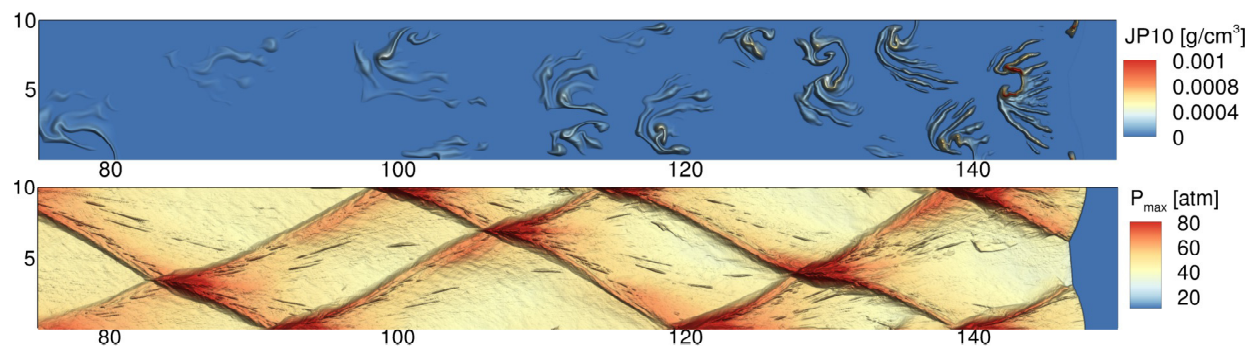


Figure 16 — The maximum pressure for each cell at the end of the simulation.

Figure 17 — Two dimensional liquid spray JP-10/oxygen detonation with $\phi = 0.77$.

REFERENCES

1. Zhou, R., Wu, D., Wang, J., "Progress of continuously rotating detonation engines," *Chinese Journal of Aeronautics*, Vol. 29, Iss. 1, pp 15-29, 2016.
2. Cocks, P.A.T., Holley, A.T., and Rankin, B.A., "High Fidelity Simulations of a Non-Premixed Rotating Detonation Engine," AIAA Paper 2016-0125, AIAA Science and Technology Forum, 2016.
3. Uemura, Y., Hayashi, A.K., Asahara, M., Tsuboi, N., and Yamada, E., "Transverse wave generation mechanism in rotating detonation," *Proceedings of the Combustion Institute*, Vol. 34, Iss. 2, pp 1981-1989, 2013.
4. Kailasanath, K. and Schwer, D.A., "High-Fidelity Simulations of Pressure-Gain Combustion Devices Based on Detonations," *J. Prop. Power*, Vol 33, No. 1, pp 153-162, 2017.
5. Strakey, P., Ferguson, D., Sisler, A., and Nix, A., "Computationally Quantifying Loss Mechanisms in a Rotating Detonation Engine," AIAA Paper 2016-0900, AIAA Science and Technology Forum, 2016.
6. Paxson, D.E., "Numerical Analysis of a Rotating Detonation Engine in the Relative Reference Frame", AIAA Paper 2014- 0284, AIAA Science and Technology Forum, 2014.
7. Kaemming, T., Fotia, M.L., Hoke, J. and Schauer, F. "Thermodynamic Modeling of a Rotating Detonation Engine Through a Reduced-Order Approach," *Journal of Propulsion and Power*, Vol. 33, No. 5, pp. 1170-1178, 2017.
8. Nordeen, C.A., Schwer, D.A., Schauer, F., Hoke, J., Barber, Th., and Cetegen, B., "Thermodynamic model of a rotating detonation engine," *Combustion, Explosion and Shock Waves*, Vol. 50, No. 5, 2014, pp 568,577.
9. Cheatham, S. and Kailasanath, K., "Numerical modelling of liquid-fuelled detonations in tubes," *Combustion Theory and Modelling*, Vol. 9, pp. 23-48, 2005.
10. Shen, H., Wang, G., Liu, K., and Zhang, D., "Numerical Simulation of Liquid-Fueled Detonations by an Eulerian-Lagrangian Model," *International Journal of Nonlinear Sciences and Numerical Simulation*, Vol. 13, No. 2, pp. 177-188, 2012.
11. Khmel, T., and Fedorov, A., "Physical and Mathematical Model of Detonation in Aluminum Gas Suspensions with Regard for Transition Processes of Nanosized Particle Flow, Heat Transfer and Combustion," *AIP Conference Proceedings*, Vol. 1893, Paper 030144, 2017.
12. Kindracki, J., "Experimental research on rotating detonation in liquid fuel-gaseous air mixtures," *Aerospace Science and Technology*, Vol 43, pp. 445-453.
13. Schwer, D.A., and Kailasanath, K., "Direct comparison of particle-tracking and sectional approaches for shock driven flows," *International Journal of Spray and Combustion Dynamics*, Vol. 1, Iss. 1, pp 732-741, 2009.
14. Schiller, L. and Naumann, A.Z., *Über die Grundlegenden Berechnungen bei der Schwerkraftaufbereitung*, *Vereines Deutscher Ingenieure*, Vol. 77, pp. 318-320, 1933.

15. Tedeschi, G., Gouin, H., and Elena, M., "Motion of tracer particles in supersonic flows," *Exper Fluids*, Vol 26, p 288, 1999.
16. Abraham, F. F., "Functional dependence of drag coefficient of a sphere on Reynolds number," *Physics of Fluids*, Vol. 13, pp. 2194-2195, 1970.
17. Ranz, W.E. and Marshall, W.R., "Evaporation from droplets," *Chem. Engrg. Prog.*, Vol. 48, No. 3, p 141, 1952.
18. Sivier, S., Loth, E., Baum, J., and Löhner, R., "Eulerian-Eulerian and Eulerian-Lagrangian methods in two phase flow," 13th International Conference on Numerical Methods in Fluid Dynamics, pp. 473-477, 1993.
19. Boris, J.P. and Book, D.L. "Flux Corrected Transport I. SHASTA, A fluid transport algorithm that works," *J. Comput. Phys.*, Vol 11, No. 1, p 38, 1973.
20. Boris, J.P., Landsberg, A.M., Oran, E.S., and Gardner, J.H., "LCPFCT—A Flux- Corrected Transport algorithm for solving generalized continuity equations." NRL Technical Report NRL/MR/6410—93-7192, Naval Research Laboratory, 1993.
21. Agrawal, G., Sussman, A. and Saltz, J., "An integrated runtime and compile-time approach for parallelizing structured and block structure applications." *IEEE Transactions on Parallel and Distributed Systems*, Vol 6, No 7, p 747, 1995.
22. Landsberg, A.M. and Boris, J.P., AIAA Paper No. 97-1982, 1997.
23. Schwer, D.A. and Kailasanath, K., "Towards Non-premixed Injection Modeling of Rotating Detonation Engines," AIAA Paper 2015-3782, AIAA Propulsion & Energy Forum 2015, 27-29 July, 2015, Orlando, FL.
24. Schwer, D.A. and Kailasanath, K., "Towards an Assessment of Rotating Detonation Engines with Fuel Blends," AIAA Paper 2017-4942, AIAA Propulsion and Energy Forum, 10-12 Jul 2017, Atlanta, GA.
25. Gordon, S. and McBride, B.J., "Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications," NASA Reference Publication 1311, 1994.
26. The MPI Forum, "MPI: A Message-Passing Interface Standard," July 2011, retrieved from <http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf>.
27. Karypis, G. and Schloegel, K. "ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library, Version 4.0," August 2011, retrieved from <http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview>.
28. Hoberock, J. and Bell, N., "Thrust: A Parallel Template Library," 2011, Version 1.4.0.
29. Bell, N., and Hoberock, J., "Thrust: A Productivity-Oriented Library for CUDA," *GPU Computing Gems: Jade Edition*, Morgan Kaufmann, 2011, pp. 359-372.
30. Hartmann, R., "Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations," *International Journal for Numerical Methods in Fluids*, Vol. 51, No. 9-10, pp. 1131–1156, 2006.

31. Johnson, R.F., Schwer, D.A., Kercher, A., Corrigan, A., Kailasanath, K., Kessler, D., and Gutmark, E., "Flow Characteristics of a Recirculating Flameless Combustor Configuration," AIAA Paper 2018-0136, AIAA Science and Technology Forum, 2018.
32. Sommerfeld, M., "The unsteadiness of shock waves propagating through gas-particle mixtures." *Exp. Fluids*, Vol 3, p 197, 1985.
33. Keenan, J.K., Keys, F.G., Hill, P.G., and Moore, J.G., *Steam Tables*, John Wiley and Sons, New York, 1969.
34. Davidson, D.F., Horning, D.C., Herbon, J.T. and Hanson, R.K., "Shock tube measurements of JP-10 ignition," *Proc. Combust. Inst.*, Vol. 28, pp 1687-1692.
35. Guo, Y., Yang, F., Xing, Y., Li, D., Fang, W., and Lin, R., "Bubble-Point Vapor Pressure Measurement for System JP-10 and Tributylamine by an Inclined Ebulliometer," *Energy & Fuels*, Vol. 22, pp. 510-513, 2008.

Appendix A

ENTHALPY SHIFT FOR SPECIES ENTHALPIES

For an ideal gas mixture, the internal energy for a mixture with N_s species:

$$\rho e_t = \sum_{i=0}^{N_s} c_i h_i(T) - P \quad (\text{A1})$$

where c_i are species concentrations, P is the pressure, and $h_i(T)$ is given by curve-fits:

$$h_i(T) = \sum_{j=0}^M a_{i,j} T^j \quad (\text{A2})$$

Using this system, we recover the correct (or as nearly correct as the curve-fits provide) enthalpy of formation at standard temperature. That is,

$$h_i(T_{\text{ref}}) = \Delta h_{f,i}^o \quad (\text{A3})$$

where $T_{\text{ref}} = 298.15$ K. We, however, do not use these enthalpies *as is* in the DASLIB codes. Instead, we define the internal energy, ρe_t , as

$$\rho e_t = \sum_i^{N_s} c_i \tilde{h}_i - P. \quad (\text{A4})$$

where $\tilde{h}_i = h_i - \tilde{a}_i$. We want to set $\tilde{h}_i(T_{\text{ref}}) = C_{p,i}(T_{\text{ref}})T_{\text{ref}}$. Thus, to get $\tilde{a}_i$, we substitute in:

$$\tilde{h}_i(T_{\text{ref}}) = h_i(T_{\text{ref}}) - \tilde{a}_i \quad (\text{A5})$$

$$C_{p,i}(T_{\text{ref}})T_{\text{ref}} = \Delta h_{f,i}^o - \tilde{a}_i \quad (\text{A6})$$

$$\tilde{a}_i = \Delta h_{f,i}^o - C_{p,i}(T_{\text{ref}})T_{\text{ref}} \quad (\text{A7})$$

This creates an energy that more closely corresponds to the $\rho e_t = p/(\gamma - 1)$ definition for the ideal, thermally perfect case. The downside to this is that we now need to incorporate the energy difference $\tilde{a}_i$ into the reaction term. Energy addition from reactions is simply:

$$\Delta(\rho e_t) = - \sum_{i=1}^N \dot{w}_i \tilde{a}_i \quad (\text{A8})$$

Appendix B

SAMPLE INPUT FILES

B.1 Two-Dimensional Glass Particle Simulation

B.1.1 DASLIB input file (*sommglass.inp*)

```
#
# das_shocktube application
#
Geometry
  Dimensions      2
  Linear
  X-control-points
    Point      0  0.5  1.1
    Point     781  0.5  1.1
  Y-control-points
    Point      0  0.5  1.1
    Point     100  0.5  1.1
End

Model
  Rgas              287e4
  Gamma             1.4
  Particle-tracking
    Two-way-coupling
    Particle-density 2.5
    Viscosity-parameters 1.458e-5 110.0
    Prandtl-number   0.75
End

Initialize
  Ambient
    Pressure      1.013e6
    Temperature    300
  End
  Driver
    Pressure      16.208e6
    Temperature    2400
  End
  Box
    Location      100 25 # location is center of box
    Size          200 50
  End
End

Initialize-particles
  Number-bins      1
  Number-virtual-particles 2000
  Diameters        26.99e-4 27.01e-4
```

```

Volume-distribution      1.0
Mass-loading             0.0007371
Box
  Location               543 25
  Size                   476 50
End
End

Simulation
  File                   output
  Write-variables        rho pre tmp vfrac rvx rvy mach
  Write-particles
  Difl                   0.995
  Dt                     1e-7
  Dtmin                  2e-8
  Dtmax                  1e-7
  Time-per-frame         1e-4
  Frames-per-restart     10
  Frames-per-output      10
  Frames                 60
End

```

B.1.2 JENRE Configuration File

```

diagnostics
  log_status             step 10
  dump_field             time 0.159158
  dump_sampler xyplane time 0.159158
                        plane extents (1562 200)
                        origin (0 0)
  bottom (7.81 0)
  left (0 1.00)
  scalars Density Pressure Temperature Mach VelocityX group1VolumeFraction DensityH2O end
end

particles
  inertial glass group1 coupled end
end

group1.initialization
  cell_based_random diameter 2.7e-5
                        temperature 300.0
                        particles_per_virtual 100000
                        cad intersection plane origin (3.05 0.)
                                                normal -1
                        plane origin (0. 0.5)
                                                normal (0 1)
                        mass_loading 0.63
end

domain.default
  slip_wall
end

domain.default.particles
  group1 specular

```

```

end

field_reference_state
  Pressure    101300
  Velocity    30
  Temperature 300
  MassFractions Air 1.000000
                  H2O 0.000000
                  end_list
end

initialization
  cad
  ambient Pressure    3241600
            Velocity    0
            Temperature 2400
            MassFractions Air 1.000000
                              H2O 0.000000
                              end_list
  upstream Pressure    101300
            Velocity    0
            Temperature 300
            MassFractions Air 1.000000
                              H2O 0.000000
                              end_list
  union plane origin 2.00
            normal -1
            plane origin (0 0.5)
            normal (0 -1)
end

solver
  step_end 50000
  cfl 0.1
end

lagrangian_rk
  levels 1
  solver_dt
end

coupled_particle_fem_controls
  coupling_type transient
  n_lagrangian_steps_per_fem_step 1
  n_iterations 1
end

solver.artificial_viscosity
  ambient cdiff 0.25
end

solver.fail_safe_limiting
  MachMax 20
  TemperatureMin 100.0
  TemperatureMax 5000.0
end

```

B.2 Two-Dimensional Water Droplet Simulation

B.2.1 DASLIB Input File (*sommwater.inp*)

```
#
# das_detonate application
#
#
Geometry
  Dimensions      2
  Linear
  X-control-points
    Point         0   0.2  1.1
    Point         781 0.2  1.1
  Y-control-points
    Point         0   0.2  1.1
    Point        100 0.2  1.1
End

Model
  Thermal-model   dascp1
  Thermal-data    "water.therm"
  Mixture         Air H2O_g Error
  Temperature-range 273 3500
  Droplet-tracking
    Two-way-coupling
    Species       H2O_l
    Gas-species   H2O_g
    Liquid-density 1.0
    Saturation-data "water.satprop"
    Viscosity-parameters 1.458e-5 110.0
    Prandtl-number 0.75
    Schmidt-number 1 1 1
End

Initialize
  Ambient
    Pressure      1.013e6
    Temperature    300
    Air           1
  End
  Driver
    Pressure      16.208e6
    Temperature    2400
    Air           1
  End
  Box
    Location      100 25
    Size          200 50
  End
End

Initialize-droplets
  Number-bins     1
  Number-virtual-particles 1000
  Diameters       25e-4 30e-4
  Volume-distribution 1.0
```

```

Mass-loading      0.00059192
Box
  Location        543 25
  Size            476 50
End
End

Simulation
  File            output
  Write-variables  rho rvx rvy tmp pre H2O_g vfrac mach
  Dt              1e-8
  Dtmin           1e-8
  Dtmax           5e-8
  Difl            0.995
  Time-per-frame  1e-4
  Frames-per-output 10
  Frames-per-restart 10
  Frames          60
End

```

B.2.2 DASLIB Thermodynamic File (*water.therm*)

```

# Thermodynamics for doing water/air calculations
# Curve-fits are done down to triple point of water (273 K)
#
# Name          Form.    Phase  Mweight    Triple  Tcrit  Pcrit  Temperature ranges
# Each line following the name of a species represents curve-fits for each
# temperature range. The curve fits
# For each curve fit, a0 ... a8
#
#   Cp = a0 + a1 T + a2 T^2 + a3 T^3 + a4 T^4 + a5 T^5
#   H  = a0 T + a1/2 T^2 + a2/3 T^3 + a3/4 T^4 + a4/5 T^5 + a5/6 T^6 + a6
#   S  = a0 ln T + a1 T + a2/2 T^2 + a3/3 T^3 + a4/4 T^4 + a5/5 T^5 + a7
#
#
# Water Mixture
#   H2O          1.00000000
# "H2O_l"        H2O          liquid  18.01528    273        626        220        273.1    600.0
# -1.320638E+02 1.862739E+00 -9.679462E-03 2.477094E-05 -3.127822E-08 1.565252E-11 -2.831109E+04 3.443677E+02
#
# Steam Mixture mole fractions:
#   H2O          1.00000000
# "H2O_g"        H2O          gas       18.01528    0          0          0          273.0    3500.0
# 3.970059E+00 -4.193932E-04 2.560506E-06 -1.453988E-09 3.390954E-13 -2.934623E-17 -3.026975E+04 1.123357E-01
#
# Air Mixture mole fractions:
#   O2           0.21000000
#   N2           0.78000000
#   Ar           0.01000000
# Air            "O2/N2/Ar"    gas       28.96968    0          0          0          273.0    3500.0
# 3.385619E+00 2.401222E-05 1.228833E-06 -9.106907E-10 2.578912E-13 -2.595150E-17 -1.019665E+03 3.992740E+00
#
# Oxygen mole fractions:
#   O2           1.00000000
# O2             O2           gas       31.99880    0          0          0          273.0    3500.0
# 2.999177E+00 2.014700E-03 -1.097679E-06 2.918398E-10 -2.478856E-14 -9.235178E-19 -9.746186E+02 7.022315E+00
#
# Inert Mixture mole fractions:
#   N2           0.98734177
#   Ar           0.01265823
# Inert          "N2/Ar"      gas       28.16447    0          0          0          273.0    3500.0

```

```

3.488132E+00 -5.043105E-04 1.846227E-06 -1.229797E-09 3.329020E-13 -3.259285E-17 -1.031606E+03 3.188399E+00
#
# Error Mixture mole fractions:
#   O2          0.21000000
#   N2          0.78000000
#   Ar          0.01000000
Error      "O2/N2/Ar"    gas    28.96968      0      0      0      273.0    3500.0
3.385619E+00 2.401222E-05 1.228833E-06 -9.106907E-10 2.578912E-13 -2.595150E-17 -1.019665E+03 3.992740E+00

```

B.2.3 DASLIB Constant Property Thermodynamic File (*water-const.therm*)

```

# Thermodynamics for doing water/air calculations with constant properties
# Curve-fits are done down to triple point of water (273 K)
#
# Name          Form.      Phase  Mweight      Triple  Tcrit  Pcrit  Temperature ranges
# Each line following the name of a species represents curve-fits for each
# temperature range. The curve fits
# For each curve fit, a0 ... a8
#
#   Cp = a0 + a1 T + a2 T^2 + a3 T^3 + a4 T^4 + a5 T^5
#   H  = a0 T + a1/2 T^2 + a2/3 T^3 + a3/4 T^4 + a4/5 T^5 + a5/6 T^6 + a6
#   S  = a0 ln T + a1 T + a2/2 T^2 + a3/3 T^3 + a4/4 T^4 + a5/5 T^5 + a7
#
# Air Mixture mole fractions:
#   O2          0.21000000
#   N2          0.78000000
#   Ar          0.01000000
Air          "O2/N2/Ar"    gas    28.97      0      0      0      273    3500.0
3.5          0.0 0.0 0.0 0.0 0.0 -1043.525      0.0000
#
# Steam Mixture mole fractions:
#   H2O          1.00000000
Steam        H2O          gas    18.015      0      0      0      273    3500.0
4.058        0.0 0.0 0.0 0.0 0.0 -30298.00      0.0000
#
# Water Mixture
#   H2O          1.00000000
Water        H2O          liquid 18.015      273      647      220      273      647
9.077        0.0 0.0 0.0 0.0 0.0 -36788.51      0.0000
#
# Error Mixture mole fractions:
#   O2          0.21000000
#   N2          0.78000000
#   Ar          0.01000000
Error      "O2/N2/Ar"    gas    28.97      0      0      0      250.0    3500.0
3.5          0.0 0.0 0.0 0.0 0.0 -1043.525      0.0000

```

B.2.4 DASLIB Saturation Property File (*water.satprop*)

# Name	Type	Pcrit[atm]	Tcrit[K]	AA	BB[K]	CC[K]	STens0[dynes/cm]	T0[K]
H2O_1	Antoine	220	626	11.70327	3835.83	-45	73	293

B.2.5 JENRE Configuration File

```

diagnostics
  log_status      step 10
  dump_field      time 0.159158
  dump_sampler xyplane time 0.159158

```

```

        plane extents (1562 200)
        origin (0 0)
    bottom (7.81 0)
    left (0 1.00)
    scalars Density Pressure Temperature Mach VelocityX group1VolumeFraction DensityH2O end
end

particles
    droplet H2O group1 coupled end
end

group1.initialization
    cell_based_random diameter 2.7e-5
        temperature 300.0
        particles_per_virtual 100000
        cad intersection plane origin (3.05 0.)
            normal -1
            plane origin (0. 0.5)
            normal (0 1)
        mass_loading 0.5
end

domain.default
    slip_wall
end

domain.default.particles
    group1 specular
end

field_reference_state
    Pressure 101300
    Velocity 30
    Temperature 300
    MassFractions Air 1.000000
        H2O 0.000000
    end_list
end

initialization
    cad
    ambient Pressure 3241600
        Velocity 0
        Temperature 2400
        MassFractions Air 1.000000
            H2O 0.000000
        end_list
    upstream Pressure 101300
        Velocity 0
        Temperature 300
        MassFractions Air 1.000000
            H2O 0.000000
        end_list
    union plane origin 2.00
        normal -1
        plane origin (0 0.5)
        normal (0 -1)

```

```

end

solver
  step_end 50000
  cfl 0.1
end

lagrangian_rk
  levels 1
  solver_dt
end

coupled_particle_fem_controls
  coupling_type transient
  n_lagrangian_steps_per_fem_step 1
  n_iterations 1
end

solver.artificial_viscosity
  ambient cdiff 0.25
end

solver.fail_safe_limiting
  MachMax 20
  TemperatureMin 100.0
  TemperatureMax 5000.0
end

```

B.3 JP-10 Liquid Droplet Detonation File

B.3.1 DASLIB Input File (*detliq.inp*)

```

#
# das_detonate application
#
#
Geometry
  Dimensions      2
  Linear
  X-control-points
    Point      0  0.025  1.1
    Point     150  0.025  1.1
  Y-control-points
    Point      0  0.025  1.1
    Point     10  0.025  1.1
End

Model
  Thermal-model  dascp1
  Thermal-data   "../jp10det.therm"
  Mixture        JP10_g Oxygen Prod Error
  Temperature-range 273 4000
  JP10-Oxygen-induction-model
    JP10_g      -0.0666667
    Oxygen      -0.9333333

```

```

    Prod          1.6161590
    Fuel           JP10_g
    Oxidizer       Oxygen
    Heat-scale     1.0
Droplet-tracking
  Two-way-coupling
  Species         JP10_l
  Gas-species     JP10_g
  Liquid-density  0.932
  Saturation-data "../satprop.dat"
  Viscosity-parameters 1.458e-5 110.0
  Prandtl-number  0.75
  Schmidt-number  1 1 1 1
End

Initialize
  Ambient
    Pressure      1.013e6
    Temperature    300
    JP10_g        1
    Oxygen        14
  End
  Driver
    Pressure      50.0e6
    Temperature    3500
    Prod          1
  End
  Box
    Location      0.05  5
    Size          0.1   10
  End
  Ellipse
    Location      1.4   4.0
    Radius        0.25
  End
  Ellipse
    Location      1.6   0.4
    Radius        0.2
  End
  Liquid
    Pressure      1.013e6
    Temperature    300
    Oxygen        1
  End
  Box
    Location      90    5
    Size          120   10
  End
End

Initialize-droplets
  Number-bins      1
  Number-virtual-particles 10
  Diameters        10e-4 11e-4
  Volume-distribution 1.0
  Mass-loading      0.0003952956
  Box

```

```

        Location      90      5
        Size          120     10
    End
End

Transducer
    File              "press.dat"
    Type              Pressure
    Point             10.0     5
    Point             20.0     5
    Point             40.0     5
End

Transducer
    File              "temp.dat"
    Type              Temperature
    Point             10.0     5
    Point             20.0     5
    Point             40.0     5
End

Simulation
    File              output
    Write-variables    rho tmp pre vfrac hrel JP10_g Oxygen Prod premax
    Dt                 1e-8
    Dtmin              2e-9
    Dtmax              1e-8
    Steps-per-diagnostics 100
    Time-per-frame     1e-5
    Frames-per-restart 10
    Frames-per-output  1
    Frames             74
End

```

B.3.2 DASLIB Thermodynamic File (*jp10det.therm*)

```

# Thermodynamics for doing JP10/oxygen calculations
# Curve-fits are done down to triple point of water (273 K)
#
# Name          Form.    Phase    Mweight    Triple    Tcrit    Pcrit    Temperature ranges
# Each line following the name of a species represents curve-fits for each
# temperature range. The curve fits
# For each curve fit, a0 ... a8
#
#    Cp = a0 + a1 T + a2 T^2 + a3 T^3 + a4 T^4 + a5 T^5
#    H  = a0 T + a1/2 T^2 + a2/3 T^3 + a3/4 T^4 + a4/5 T^5 + a5/6 T^6 + a6
#    S  = a0 ln T + a1 T + a2/2 T^2 + a3/3 T^3 + a4/4 T^4 + a5/5 T^5 + a7
#
# JP10/Oxy Detonation mixture
# Detonation Velocity: 2299.8 m/s
# P/P1:                39.098
# T/T1:                13.032
#
# Detonation product mole fractions:
# CO                   0.27482649
# CO2                  0.13767826
# H                    0.04603276
# HO2                  0.00033002
# H2                   0.04546273

```

```

# H2O 0.20061204
# O 0.06509391
# OH 0.12147729
# O2 0.10848651
Prod "Products" gas 24.09900 0 0 0 273.0 4000.0
3.306263E+00 1.137439E-03 3.871835E-08 -1.999259E-10 6.006192E-14 -5.598032E-18 -1.333788E+04 3.429610E+00
#
# JP10 gaseous properties
"JP10_g" C10H16 gas 136.23404 0 0 0 273.0 4000.0
-1.230220E+01 1.280529E-01 -8.764174E-05 3.198449E-08 -5.936589E-12 4.386334E-16 -1.175610E+04 7.874349E+01
#
# Oxygen properties
Oxygen O2 gas 31.99880 0 0 0 273.0 4000.0
2.969213E+00 2.173119E-03 -1.358823E-06 4.725150E-10 -7.977703E-14 5.150553E-18 -9.707508E+02 7.153757E+00
#
# Air Inerts mixture
# N2 0.98734177
# Ar 0.01265823
Inert "N2/Ar" gas 28.16447 0 0 0 273.0 4000.0
3.389785E+00 -2.001122E-05 1.080731E-06 -7.152041E-10 1.796774E-13 -1.596824E-17 -1.017993E+03 3.629651E+00
#
# Air mixture for Error
# N2 0.78000000
# Ar 0.01000000
# O2 0.21000000
Error "O2/N2/Ar" gas 28.96968 0 0 0 273.0 4000.0
3.301374E+00 4.408946E-04 5.678883E-07 -4.654275E-10 1.250902E-13 -1.152292E-17 -1.008056E+03 4.370169E+00
#
# JP10 liquid properties
# C10H16 1.00000000
"JP10_l" C10H16 liquid 136.23404 0 698 37.33 273.0 3500.0
3.951250E+00 7.375000E-02 0.0 0.0 0.0 0.0 -1.960061E+04 0.0

```

B.3.3 DASLIB Saturation Property File (*water.satprop*)

# Name	Type	Pcrit[atm]	Tcrit[K]	AA[atm]	BB[K]	CC[K]	STens0[dynes/cm]	T0[K]
#								
Water	Antoine	220	626	11.70327	3835.83	-45	73	293
JP10_l	Antoine	37.33	698	8.173667	2783.0	-117.037	33	273