ALLSPD-3D
Version 1.0a

K.-H. Chen
Ohio Aerospace Institute
Brook Park, Ohio

B. Duncan
NYMA, Inc.
Brook Park, Ohio

D. Fricker
Vehicle Propulsion Directorate
U.S. Army Research Laboratory
Lewis Research Center
Cleveland, Ohio

J. Lee and A. Quealy
NYMA, Inc.
Brook Park, Ohio

April 1996
DISTRIBUTION POLICY

NASA policy, as outlined in NASA Management Instruction (NMI) 2210.2B, paragraph 6, subparagraph a(5), allows the release of preliminary or development versions of computer codes only if the recipient assists in the development process. Organizations receiving the code will, therefore, be expected to inform the Internal Fluid Mechanics Division of NASA Lewis Research Center of problems encountered in using the code and the documentation. We expect that you will run the example cases to confirm the successful installation of the code and to become familiar with the operation of the code.

Release of the code is restricted to domestic organizations whose major computational activity is conducted within the United States. In addition, the use of the code must be limited to the recipient, and further distribution of the code by the recipient is prohibited. Neither NASA Lewis Research Center, nor any of its employees, assumes any responsibility or legal liability for the accuracy or completeness of the code or represents that its use would not infringe on privately owned rights.

If the results of the code’s use or incorporation with other code(s) are presented in oral or written form, please give credit for its development to the Internal Fluid Mechanics Division of NASA Lewis Research Center.

The code and documentation are still in development. Input from you on your experience with the code and its documentation will aid in their development. If you find errors, or have questions or comments, please contact the ALLSPD team via email at allspd@lerc.nasa.gov or contact David M. Fricker at 216-433-5960 or Kuo-Huey Chen at 216-962-3092.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
Foreword

Aircraft engine companies in the U.S. have been using reacting flow CFD computer codes for many years as a design tool for new combustor designs and for the analysis of flow inside existing combustors to solve design problems. In spite of this record of successful use of CFD codes for combustors, the engine companies all agree that there is a need for combustor codes with capabilities much greater than those currently in use. Generally, current codes are limited in accuracy, are time-consuming in establishing a grid system which represents the complex geometries of a combustor, present difficulties in obtaining a stable solution (limited robustness), and require large execution times on mainframe supercomputers for practical problems. At the same time, the down-sizing that has occurred within these companies has left them with very limited resources to independently pursue an improved combustor code. They envision a partnership with the U.S. Government (NASA, Army, Air Force, etc.) to be a more cost-effective approach for producing improved combustor code capability that can be used by all within the U.S. without compromising proprietary, company-owned combustor designs. We at Lewis Research Center are pursuing this approach in the following way.

Efficient and accurate CFD algorithms have been developed for compressible flow applications over the last several years. These techniques have been modified for computational efficiency at the low Mach numbers which are typically found in combustors. Added to this numerical algorithm are state-of-the-art physical models for liquid fuel spray, complex turbulence, and finite rate chemistry for combustion. This Workshop marks the formal release of ALLSPD-3D to the U.S. combustion community. Concurrent with the ALLSPD effort, we are pursuing even more promising CFD approaches such as unstructured grids, finite element flow solvers, and adaptive re-meshing. The hallmark of these longer term efforts is the initiation of the National Combustor Code (NCC), a working partnership with several U.S. companies, NASA, and other agencies to jointly develop the next generation code. The first version of the NCC is expected to be available in 1998.

We believe that the ALLSPD-3D code being released at this time will be a major step forward in computational capability for its users. It should provide a user-friendly platform for executing the analyses of complex reacting flow problems with state-of-the-art physical models. Greater physical model fidelity and an improved gridding methodology are being pursued through the NCC partner-
ship and will be available in the future. In the meantime, it is our desire that ALLSPD-3D provide computational capability to the U.S. industry until the fruits of the NCC ripen and are harvested.

The ALLSPD combustor code has been in development at Lewis since 1991. An initial 2-D version of the code was released in 1993. Some of the major features of the present 3-D code are a stochastic fuel spray model, a k-ε turbulence model, a library of finite rate combustion models, internal obstacles, multiple grid zone capability, and parallel processor execution. ALLSPD-3D has a graphical user interface to increase the comfort level of the user and has been run on supercomputers, a variety of scientific workstations, and a cluster of workstations. The code has successfully computed a wide range of reacting and non-reacting flow problems of increasing geometric complexity. The full validation of ALLPSD-3D will be an interactive process with the users as they gain experience with the code. The Lewis Research Center will work with the ALLSPD-3D users as they perform calculations. We are looking forward to a productive working relationship with you, the "customers," of our code development effort!

Dr. Edward J. Mularz
Chief, Engine Components Division
U.S. Army Research Laboratory - Vehicle Propulsion Directorate
NASA Lewis Research Center
November 14, 1995

(Dr. Mularz, as Chief of the Aerothermochemistry Branch at NASA Lewis Research Center, initiated the ALLSPD Code effort and guided its development until the recent move to his new position.)
# Table of Contents

Foreword ................................................................. i
List of Figures .......................................................... vii
Abstract .................................................................. xi
Nomenclature ............................................................... xiii
I : Introduction .......................................................... 1
II : Mathematical Formulations ........................................ 7
    Gas-Phase Equations ............................................. 7
    All-Mach-Number Formulation .................................. 10
    Liquid-Phase Equations .......................................... 12
    Droplet Motion Equations ....................................... 12
    Droplet Mass and Heat Transfer Equations ................. 13
    Droplet Internal Temperature Equations ................. 14
    Chemical Source Term ........................................... 16
    Equilibrium Rate Kinetics Model ............................ 18
    Two Equation Model of Turbulence ......................... 20
    Transport Property Model ...................................... 22
    Thermodynamic Model .......................................... 24
    Turbulent Chemistry Model .................................... 25
III : Numerical Methods ................................................. 27
    LU Scheme ........................................................ 31
    Mass Conservation ............................................... 34
    Spray Source Terms .............................................. 35
    Stochastic Process ............................................... 35
    Determination of Spray Time Step ............................ 36
    Interaction Between Two Phases ............................. 40
    Pseudo-Time Step ................................................ 40
    Boundary Conditions .............................................. 41
    Wall Boundary Condition ....................................... 42
Species Continuity Equations ........................................... 44
Dilution Holes or Transpiration Cooling (or Bleed) .......... 45
Upstream ........................................................................... 45
Downstream ....................................................................... 46
Periodic .............................................................................. 46
Symmetry ............................................................................ 47
Far Field ............................................................................. 47
Singularity ........................................................................... 47

IV : Input/Output Files ...................................................... 49
Input/Output ....................................................................... 49
Input Data Files .................................................................... 50
  Grid Data: grid.dat .......................................................... 50
  Multi-Block Data: gridlink.dat ......................................... 51
Flow Data: allspd.dat ......................................................... 54
User Defined Initial Conditions: uvwt.dat ....................... 59
Property Data: prop.dat ...................................................... 61
User Defined Spray Data: spray.dat .................................... 62
Output Data Files ............................................................... 63
  Plot3d Data Files ............................................................. 63
  Spray Data Files ............................................................. 64
  Restart File ....................................................................... 65
  Residual File .................................................................... 65
Typical Chemistry Input File Structure ......................... 66
  One-Step Global Chemistry Mode .................................... 67
  Simple Two-Step Global Chemistry Model .................... 68
  Typical Detailed Chemistry Input file ............................ 70

V : Graphical User Interface(GUI) ...................................... 75
  Installation ....................................................................... 75
General Overview ............................................................. 77
File ................................................................................. 79
Grid .................................................................................. 80
Model ............................................................................... 80
  Model/File ...................................................................... 80
  Model/Manipulate Grid .................................................. 81
  Model/Problem Description ............................................ 81
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model/Properties</td>
<td>82</td>
</tr>
<tr>
<td>Model/Boundary Conditions</td>
<td>89</td>
</tr>
<tr>
<td>Model/Help</td>
<td>91</td>
</tr>
<tr>
<td>Solve</td>
<td>91</td>
</tr>
<tr>
<td>Solve/File</td>
<td>91</td>
</tr>
<tr>
<td>Solve/Control</td>
<td>91</td>
</tr>
<tr>
<td>Solve/Output</td>
<td>93</td>
</tr>
<tr>
<td>Solve/Solution</td>
<td>93</td>
</tr>
<tr>
<td>Solve/Help</td>
<td>95</td>
</tr>
<tr>
<td>View</td>
<td>96</td>
</tr>
<tr>
<td>Help</td>
<td>96</td>
</tr>
<tr>
<td>Known Bugs</td>
<td>96</td>
</tr>
<tr>
<td>Future Plans</td>
<td>97</td>
</tr>
<tr>
<td>VI : Code Structure and Hardware Implementations</td>
<td>99</td>
</tr>
<tr>
<td>General Structure</td>
<td>99</td>
</tr>
<tr>
<td>Program Execution (sans GUI)</td>
<td>108</td>
</tr>
<tr>
<td>Parallel Implementation</td>
<td></td>
</tr>
<tr>
<td>Theory</td>
<td>111</td>
</tr>
<tr>
<td>Implementation</td>
<td>111</td>
</tr>
<tr>
<td>Notes About Coding Details</td>
<td>113</td>
</tr>
<tr>
<td>Compilation and Run-time Details</td>
<td>114</td>
</tr>
<tr>
<td>VII : Examples</td>
<td>117</td>
</tr>
<tr>
<td>3-D Non-Reacting Transition Duct Flow</td>
<td>117</td>
</tr>
<tr>
<td>Axisymmetric Swirl-Can Combustor Flow with Spray</td>
<td>122</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>129</td>
</tr>
<tr>
<td>References</td>
<td>131</td>
</tr>
<tr>
<td>Appendix A</td>
<td>137</td>
</tr>
<tr>
<td>Appendix B</td>
<td>141</td>
</tr>
<tr>
<td>ALLSPD3D V1.0 -I/O GUIDE</td>
<td>143</td>
</tr>
</tbody>
</table>
List of Figures

FIGURE 1. Partially Decoupled Procedure .................................................. 28
FIGURE 2. Vectorized Oblique Plane ........................................................... 34
FIGURE 3. Read file format for grid.dat ...................................................... 50
FIGURE 4. Typical gridlink.dat file .............................................................. 51
FIGURE 5. Typical Listing of the allspd.dat Input File ................................ 54
FIGURE 6. Typical prop.dat File .................................................................. 61
FIGURE 7. ALLSPD-3D GUI Main Menu Window ....................................... 79
FIGURE 8. ALLSPD-3D GUI Model Menu Window ..................................... 80
FIGURE 9. ALLSPD-3D GUI Solve menu window ...................................... 91
FIGURE 10. ALLSPD-3D Flow Chart ......................................................... 100
FIGURE 11. ALLSPD-3D Flow Chart (cont’d) ............................................ 101
FIGURE 12. ALLSPD-3D Flow Chart (cont’d) ............................................ 102
FIGURE 13. Coordinates ............................................................................ 105
FIGURE 14. Two Dimensional Planar and Axisymmetric Configurations .... 106
FIGURE 15. The Transition Duct Grid at Three Streamwise Stations .......... 118
FIGURE 16. The gridlink.dat Input File for the Transition Duct ................. 118
FIGURE 17. The allspd.dat Input File for the Laminar Transition Duct ....... 119
FIGURE 18. The prop.dat Input file for the Transition Duct ....................... 119
FIGURE 19. Velocity Vectors for the Laminar Transition Duct ................. 120
FIGURE 20. Convergence History for the Laminar and Turbulent Transition Duct 120
FIGURE 21. The allspd.dat Input File for the Turbulent Transition Duct ...... 121
FIGURE 22. Velocity Vectors for the Turbulent Transitions Duct ............... 121
FIGURE 23. Velocity Profile Comparison .................................................... 122
FIGURE 24. The gridlink.dat Input File for the Swirl Can Combustor .......... 123
FIGURE 25. The allspd.dat Input File for the Swirl Can Combustor .......... 124
FIGURE 26. The prop.dat Input File for the Swirl Can Combustor .............. 125
FIGURE 27. Convergence History for the Swirl Can Combustor ................. 126
FIGURE 28. Grid for the Swirl Can Combustor .......................................... 127
FIGURE 29. Temperature Colored Velocity Vectors ............................................. 127
List of Tables

TABLE 1. ALLSPD-3D Main Features ......................................................... 6
TABLE 2. Turbulence Model Parameters ................................................... 21
TABLE 3. Detailed Information about gridlink.dat File ............................... 52
TABLE 4. Detailed Layout of allspd.dat File .............................................. 55
TABLE 5. Detailed Information about prop.dat File .................................... 61
TABLE 7. Typical Two-step Chemistry Input File ...................................... 68
TABLE 8. Detailed Chemistry Input File ................................................... 70
TABLE 9. The make Commands for Supported Platforms ............................ 108
TABLE 10. The pregrid Command Line Options ...................................... 109
TABLE 11. The allsp3d3d Command Line Options .................................. 110
TABLE 12. Reference Conditions for the Transition Duct ........................... 118
TABLE 13. Reference Conditions for the Swirl Can Combustor ....................... 123
TABLE 14. Swirl Can Combustor Grid Block Sizes .................................. 126
TABLE 15. Sample GUI Database Entry ................................................ 141
TABLE 16. Description of GUI Database Entries .................................... 142
Abstract

This report describes the ALLSPD-3D computer program for the calculation of three-dimensional, chemically reacting flows with sprays. ALLSPD-3D employs a coupled, strongly implicit solution procedure for turbulent spray combustion flows. ALLSPD-3D utilizes generalized curvilinear coordinates with both multi-block and flexible internal blockage capabilities to handle complex geometries. The ALLSPD-3D algorithm, which employs preconditioning and eigenvalue rescaling techniques, is capable of providing efficient solution of flows over a wide range of subsonic Mach numbers. The chemistry treatment in the code is general enough that an arbitrary number of reactions and species can be defined by the users. In addition, for general industrial combustion applications, the code provides both dilution and transpiration cooling capabilities. A stochastic spray model and an efficient method for treating the spray source terms in the gas-phase equations are used to analyze evaporating spray droplets. Although written for three-dimensional flows, the code can be used for two-dimensional and axisymmetric flow computations as well. The code is written in such a way that it can be run on various computer platforms (supercomputers, workstations, and parallel processors) while the GUI (Graphical User Interface) should provide a user-friendly tool for setting up and running the code.
Nomenclature

Symbols

\begin{align*}
A, B, C & \quad \text{Convective Jacobians} \\
A_0 & \quad \text{Eddy Viscosity Modeling Parameter, Equation (40)} \\
A_f, A_b & \quad \text{Finite Reaction Rate Pre-exponents, Equation (25)} \\
a_1, a_2, a_3 & \quad \text{Turbulence Model Damping Coefficients, Equation (39)} \\
C_j & \quad \text{Species Concentration} \\
C_{e1}, C_{e2} & \quad \text{Turbulence Model Constants, Table 1} \\
C_\mu & \quad \text{Eddy Viscosity Constant, Equation (39)} \\
\mathbf{CR}_{ik} & \quad \text{Difference Between } v'_{ik} \text{ and } v''_{ik}, \text{Equation (26)} \\
\mathbf{CS} & \quad \text{Species Symbol, Equation (23)} \\
D_{ij} & \quad \text{Diffusion Coefficients}[m^2/s], \text{Equation (44)} \\
E_{b,f} & \quad \text{Activation Energy for Backward and Forward Reactions} \\
G & \quad \text{Gibbs Free Energy} \\
g & \quad \text{Specific Gibbs Free Energy} \\
H & \quad \text{Source Vector} \\
H & \quad \text{Total Enthalpy } (h+(u^2+v^2+w^2)/2) \\
h & \quad \text{Mixture Enthalpy [Joules/kg]} \\
h_i & \quad \text{Specific Enthalpy of } i^{th} \text{ species [Joules/kg]} \\
\bar{h} & \quad \text{Convective Heat Transfer Coefficient} \\
I & \quad \text{Identity Matrix} \\
K & \quad \text{Finite Rate Reaction Constants} \\
k & \quad \text{Turbulent Kinetic Energy} \\
LHS & \quad \text{Left Hand Side} \\
l & \quad \text{Length Scale [m]} \\
MW_i & \quad \text{Molecular Weight of } i^{th} \text{ Species}
\end{align*}
\( m \) Mass Flow Rate [kg/s]
\( m_p \) Evaporation Rate [kg/s/m²]
\( N_r \) Number of Chemical Reactions
\( N \) Number of Chemical Species
\( p \) Pressure, [N/m²]
\( p_{atm} \) 1.01325x10⁵ [N/m²]
RHS Right Hand Side
\( p_k \) Turbulent Production, Equation (36)
\( Pr \) Prandtl Number
\( r_p \) Spray Particle Radius.
\( R_u \) Universal Gas Constant
\( R_t \) Turbulent Reynolds Number \((k/\nu e)\)
\( R_k \) Turbulent Reynolds Number \((k^{1/2}y_n/v)\)
\( R_p \) \( p_{atm}/R_u \)
\( S_{ij} \) Strain Rate Tensor, Equation (40)
\( T \) Temperature
\( T_u \) Turbulence Intensity
\( T_{ijk} \) Source Jacobian
\( u,v,w \) Mean Velocity Components
\( U_\tau \) Friction Velocity \(((\tau_{w}/\rho)^{1/2})\)
\( U_{ref} \) Reference Velocity
\( x,y,z \) Coordinates
\( X_i \) Mole Fraction
\( y_n \) Nearest Distance to a No-Slip Surface
\( y^+ \) Non-Dimensional Distance
\( Y_i \) Species Mass Fraction
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta$</td>
<td>Kinetic energy per unit mass($u^2+v^2+w^3$)</td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>Specific Dissipation</td>
</tr>
<tr>
<td>$\delta_{ij}$</td>
<td>Kronecker Delta</td>
</tr>
<tr>
<td>$\Gamma$</td>
<td>Preconditioning Vector</td>
</tr>
<tr>
<td>$\kappa$</td>
<td>Thermal Conductivity, [Joules/(m-s-K)]</td>
</tr>
<tr>
<td>$\Lambda$</td>
<td>Low Reynolds Number Terms</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Viscosity [kg/m/s]</td>
</tr>
<tr>
<td>$\mu_t$</td>
<td>Eddy Viscosity [kg/m/s]</td>
</tr>
<tr>
<td>$\nu$</td>
<td>Kinematic Viscosity [m$^2$/s]</td>
</tr>
<tr>
<td>$\nu'^{ik}, \nu''^{ik}$</td>
<td>Stoichiometric Coefficients for $i^{th}$ Species in $k^{th}$ Reaction</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Viscous Diffusion Coefficient, Table 1</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Shear Stress</td>
</tr>
<tr>
<td>$\tau_{ij}$</td>
<td>Reynolds Stresses, Equation (37)</td>
</tr>
<tr>
<td>$\omega_{ij}$</td>
<td>Rate of Change of Species Concentration [kg of $i^{th}$ species/(m$^3$-s)], Equation (24)</td>
</tr>
</tbody>
</table>

Subscripts and Superscripts

- $a, b, c$ With Respect to the Directions
- eff Effective Quantities
- inj With Respect to Injector Holes, and Slots
- $k$ With Respect to Turbulent Kinetic Energy
- $m$ Group Index
- $p$ With Respect to $p^{th}$ Particle Group
- $n$ Pseudo Time Index
- ref With Respect to Reference Condition
- $t$ With Respect to Turbulence
- $v$ With Respect to Viscous Terms
- $^\wedge$ Transformed Variable
- $\infty$ Freestream
- $0$ At the Inlet
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e)</td>
<td>With Respect to Specific Dissipation</td>
</tr>
<tr>
<td>(\kappa)</td>
<td>With Respect to Conductivity</td>
</tr>
<tr>
<td>(\mu)</td>
<td>With Respect to Viscosity</td>
</tr>
</tbody>
</table>
I: Introduction

It has been about two decades\(^1\) since the U.S. Government sponsored a major Computational Fluid Dynamics (CFD) code development effort focused on the analysis of reacting flows inside combustors of gas turbine engines or liquid rocket engines. During the ensuing years, many advances in CFD have been realized, especially in the area of compressible flows. Applying these advanced techniques in a new reacting flow code has the potential for significant improvements in accuracy, efficiency, and robustness over the current design codes being used in the industry. For the past few years, NASA Lewis Research Center has undertaken the development of a new combustor analysis code, ALLSPD-3D, which employs state-of-the-art computational technology and physical models. During this development, NASA Lewis will provide from time to time a current version of the code to the aerospace industry, and will solicit feedback from users on the code's shortcomings and suggestions for further improvements. (Please direct any questions or comments regarding the ALLSPD-3D code to allspd@lerc.nasa.gov.)

The current ALLSPD-3D code has evolved from the two-dimensional version released in June, 1993. Besides extension to three dimensions, the new code features several additional improvements, including a user-friendly graphical user interface (GUI), multi-platform capability (supercomputers, workstations, and parallel processors), improved turbulence and spray models, generalized chemistry, a new solution procedure, and more general boundary conditions. In this report, the fundamental structure of the ALLSPD-3D code, numerics, physical models, and I/O procedure are described in detail.

The numerics of the ALLSPD-3D code are based on modern compressible
algorithms to take advantage of advances made in CFD during the past two decades. It is well known that numerical methods developed for compressible flows are often ineffective at low Mach numbers. There are two reasons\textsuperscript{2,6} for this difficulty. First, the system's eigenvalues become stiff at low flow velocities. In theory, this can be circumvented by using implicit numerical schemes. In practice, however, a large approximate factorization error in multiple dimensions limits an optimal CFL number beyond which convergence slows down. At low Mach numbers, there are large disparities among CFL numbers and eigenvalues in three directions leads to further uncertainties. Therefore, the optimal CFL numbers can be difficult to determine. Second, the pressure term in the momentum equations becomes singular as the Mach number approaches zero, yielding a large round-off error. This behavior smears the pressure variation and often produces inaccurate solutions. Following the approach in Shuen et. al.\textsuperscript{6} and Chen,\textsuperscript{7} these difficulties are circumvented in the ALLSPD-3D code by using the pressure decomposition and the eigenvalue rescaling techniques. The results of these treatments show that the convergence characteristics of the code can indeed formulated to be independent of the flow Mach number.\textsuperscript{6}

Chemically reacting flows exhibit another class of numerical difficulties because of the wide range of time scales involved in the calculation. To avoid this chemical stiffness problem, the chemical source terms are treated fully implicitly. This measure is equivalent to preconditioning the time derivative terms of the species conservation equations so that all chemical and convective processes proceed at approximately the same numerical rate.

As pointed out in Shuen et. al.,\textsuperscript{6} the Navier-Stokes computation of a general three-dimensional turbulent combustion flow has been a tremendous task in the past and even with today's supercomputers and high end workstation clusters, it is still very difficult to accomplish such a demanding job. To understand the magnitude of such computations, it is useful to analyze a typical three-dimensional reacting flow problem with detailed chemistry. The compressible Navier-Stokes equations are usually required because of the wide density vari-
ations resulting from the combustion process despite the low flow speed in most practical applications. In addition, the turbulent flowfield must be modeled for most cases with only few exceptions. Thus, with the standard $k$-$\varepsilon$ equations to model the turbulence, seven equations are required to solve for turbulent non-reacting flows. For reacting flows with $N$ species considered, the total number of equations to be solved for a three-dimensional turbulent reacting flow is $7+(N-1)$, and the number of species ($N$) to completely describe a typical hydrocarbon fuel reaction is generally greater than 40.\textsuperscript{10} Coupling this number of equations to be solved with a grid suitable to describe a typical combustor geometry (250,000+ cells) quickly exceeds the capabilities of existing computer technology. It is definitely impractical to include so many species and couple with the flow equations in terms of today’s (and in the near future’s) computer technology. In the past, ways have been sought to reduce the number of species (reduced mechanism) for some particular fuels in order to form a more tractable CFD problem.\textsuperscript{9} Unfortunately, although significantly reduced, the number of species is still between 10 and 20 depending on the particular application. Further reduction below this level is unlikely because the reduction will also lead to inadequate description of the combustion physics.\textsuperscript{10}

The numerical issues relating to solving the governing equations for turbulent reacting flows are robustness, CPU time requirements, implicitness, stability, equation coupling, and memory. Some of these issues are very closely related, such as robustness, implicitness, stability and equation coupling. During the past two decades, the development of numerical algorithms and the usage of various solvers in CFD have been more or less dependent on the availability and accessibility of computer resources. The trend has evolved from explicit, decoupled schemes to strongly implicit and fully coupled strategies. The pros and cons among various schemes depend on the problems of interest and sometimes the vectorization and/or parallelization on computer platforms as well. Nevertheless, strongly implicit, fully coupled schemes generally give better stability, robustness and convergence, at least for steady state calculations; however, they usually take more CPU time per iteration compared to explicit schemes. For combustion flow computations, equation coupling (at least for species equa-
tions) is inevitable due to the strong coupling nature of the chemical source terms in a detailed chemistry treatment.

In order to compromise between the degree of coupling and the total memory requirement, the partially decoupled numerical scheme in developed by Chen and Shuen\textsuperscript{11} has been adopted for ALLSPD-3D. This procedure divides the whole set of equations into three groups. The first group is the five flow equations, the second group is the two turbulent equations and the third group is the species equations. Each group of equations is solved in a coupled manner and the subsequent sweep through the three groups completes one iteration. This arrangement provides greater flexibility in various types of flow calculations while maintaining the strongly coupled and implicit manner within the existing combustion code structure. The numerical issues relating to this partially decoupled process are described in Chen and Shuen\textsuperscript{11}.

The solver in the ALLSPD-3D code has been modified to include the LU scheme which, through our tests,\textsuperscript{7} is thought to be more suitable for solving large sets of coupled three-dimensional reacting flow equations.

A recently developed low Reynolds number $k$-$\varepsilon$ model\textsuperscript{12} is used to represent the effects of turbulence. In this model, the $k$-$\varepsilon$ equations are solved all the way to the wall and the conventional wall function treatment of near wall regions is not needed. Some variations of implementing this model, especially the near wall treatment, are also available as an option to alleviate the fine grid requirements near walls (which is typically a heavy burden associated with this type of turbulence model for three-dimensional calculations). Detailed treatment of the turbulent model is discussed later in this report. Although turbulence is included, the mean flow quantities are still used in the chemistry calculations (laminar chemistry) in the current version of the code. A simple eddy breakup model (EBM)\textsuperscript{13} is available as an option to model the turbulence effect on combustion.

The liquid phase is treated by solving Lagrangian equations of motion and transport for the life histories of a statistically significant sample of individual droplets. This involves dividing the droplets into $n$ groups (defined by position,
velocity, temperature, and diameter) at the fuel nozzle exit and then computing their subsequent trajectories in the flow. The spray model used in this study is based on a dilute spray assumption which is valid in regions where the droplet loading is low. The liquid fuel is assumed to enter the combustor as a fully atomized spray comprised of spherical droplets. The present model does not account for the effects due to droplet breakup and coalescence processes which might be significant in a dense spray situation. A stochastic process is included in the spray calculation to account for the effect of turbulence dispersion on the spray trajectories. Also the temperature within each droplet is non-uniform and a vortex model is used to compute the internal temperature distribution.

Most combustors in aero-propulsive engines involve very complex geometries. To enhance geometrical flexibility, the ALLSPD-3D code is written in generalized curvilinear coordinates and is equipped with multi-block grid and internal blockage capabilities. These features allow the users to handle complex geometries. Some of the main features of ALLSPD-3D are listed in Table 1
TABLE 1. ALLSPD-3D Main Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D/2D/ axisymmetric geometries</td>
<td></td>
</tr>
<tr>
<td>Finite-difference, compressible flow formulation with low Mach number</td>
<td>Preconditioning (current version is intended only for subsonic flow simulations and uses central-differencing for convective and viscous terms on right and left-hand sides)</td>
</tr>
<tr>
<td>Strongly coupled, implicit steady-state algorithm</td>
<td></td>
</tr>
<tr>
<td>Generalized finite rate chemistry</td>
<td></td>
</tr>
<tr>
<td>Generalized property and chemical reactions database</td>
<td></td>
</tr>
<tr>
<td>Eddy breakup model for turbulence-chemistry closure</td>
<td></td>
</tr>
<tr>
<td>Low-Reynolds number k-ε turbulence model</td>
<td></td>
</tr>
<tr>
<td>Spray combustion (a stochastic, separated flow model with vortex model for droplet internal temperature distribution)</td>
<td></td>
</tr>
<tr>
<td>Dilution and transpiration cooling capabilities</td>
<td></td>
</tr>
<tr>
<td>Serial &amp; parallel versions (using PVM for message passing language) on various platforms (IBM, SGI, SUN, HP workstations, CRAY)</td>
<td></td>
</tr>
<tr>
<td>GUI for problem setup and post-processing</td>
<td></td>
</tr>
<tr>
<td>Various boundary conditions (multiple inlets/outlets, dilution holes, transpiration holes, periodic, symmetry, far-field, adiabatic or isothermal walls, centerline singularity)</td>
<td></td>
</tr>
</tbody>
</table>
II : Mathematical Formulations

2.1 Gas-Phase Equations

The three-dimensional, unsteady, compressible, density-weighted, time-averaged Navier-Stokes equations and species transport equations for a chemically reacting gas of \( N \) species written in generalized non-orthogonal coordinates can be expressed as

\[
\frac{\partial}{\partial \tau} \tilde{Q} + \frac{\partial}{\partial \xi} \left( \tilde{E} - \tilde{E}_V \right) + \frac{\partial}{\partial \eta} \left( \tilde{F} - \tilde{F}_V \right) + \frac{\partial}{\partial \zeta} \left( \tilde{G} - \tilde{G}_V \right) = \tilde{H}_c + \tilde{H}_i
\]

(1)

where the vectors \( \tilde{Q}, \tilde{E}, \tilde{F}, \tilde{G}, \tilde{E}_V, \tilde{F}_V, \tilde{G}_V, \tilde{H}_c \) and \( \tilde{H}_i \) are defined as

\[
\begin{align*}
\tilde{Q} &= \frac{1}{j}Q \\
\tilde{E} &= \frac{1}{j} (\xi_x Q + \xi_y E + \xi_z F + \xi_z G) \\
\tilde{E}_V &= \frac{1}{j} (\xi_x E + \xi_y F + \xi_z G) \\
\tilde{F} &= \frac{1}{j} (\eta_x Q + \eta_y E + \eta_z F + \eta_z G) \\
\tilde{F}_V &= \frac{1}{j} (\eta_x E + \eta_y F + \eta_z G) \\
\tilde{G} &= \frac{1}{j} (\zeta_x Q + \zeta_y E + \zeta_z F + \zeta_z G) \\
\tilde{G}_V &= \frac{1}{j} (\zeta_x E + \zeta_y F + \zeta_z G) \\
\tilde{H}_c &= \frac{1}{j} H_c \\
\tilde{H}_i &= \frac{1}{j} H_i
\end{align*}
\]

(2)

In the above expressions, \( \tau, \xi, \eta, \xi \) and \( \zeta \) are the time and generalized spatial coordinates, and \( \xi_x, \eta_x \) and \( \zeta_x \) are the grid speed terms. The variables \( \xi_x, \xi_y, \xi_z, \eta_x, \eta_y, \eta_z, \zeta_x, \zeta_y, \zeta_z \) and \( \zeta_x \) are the metric terms and \( J \) is the transformation Jacobian. The vectors \( Q, E, F, G, E_V, F_V, G_V, H_c \) and \( H_i \) in the above definitions are
\[
Q = \left( \rho, \rho u, \rho v, \rho w, \rho E_t, \rho k, \rho e, \rho Y_1 \ldots \rho Y_{N-1} \right)^T
\]
\[
E = \left( \rho u, \rho u^2 + p, \rho u v, \rho u w, \left( \rho E_t + p \right) u, \rho k u, \rho e u, \rho u Y_1 \ldots \rho u Y_{N-1} \right)^T
\]
\[
F = \left( \rho v, \rho u v, \rho v^2 + p, \rho v w, \left( \rho E_t + p \right) v, \rho k v, \rho e v, \rho v Y_1 \ldots \rho v Y_{N-1} \right)^T
\]
\[
G = \left( \rho w, \rho u w, \rho v w, \rho w^2 + p, \left( \rho E_t + p \right) w, \rho k w, \rho e w, \rho w Y_1 \ldots \rho w Y_{N-1} \right)^T
\]
\[
E_v = \left( 0, \tau_{xx}, \tau_{xy}, \tau_{xz}, \alpha_1 \tau_{xx} + \kappa \tau_{xy} + \nu_1 \tau_{xz} + q_{x1} \tau_{xx} + q_{x1} \tau_{xy} + q_{x1} \tau_{xz} + q_{x1} \tau_{yz} + q_{x1} \tau_{xz} + q_{x1} \tau_{zz} \right)^T
\]
\[
F_v = \left( 0, \tau_{yx}, \tau_{yy}, \tau_{yz}, \alpha_1 \tau_{yx} + \kappa \tau_{yy} + \nu_1 \tau_{yz} + q_{y1} \tau_{yx} + q_{y1} \tau_{yy} + q_{y1} \tau_{yz} + q_{y1} \tau_{zz} \right)^T
\]
\[
G_v = \left( 0, \tau_{zx}, \tau_{zy}, \tau_{zz}, \alpha_1 \tau_{zx} + \kappa \tau_{zy} + \nu_1 \tau_{zz} + q_{z1} \tau_{zx} + q_{z1} \tau_{zy} + q_{z1} \tau_{zz} + q_{z1} \tau_{yy} + q_{z1} \tau_{zz} \right)^T
\]
and the source term vectors \( H_C \) and \( H_I \) are
\[
H_C = \left( 0, 0, 0, 0, 0, 0, S_k, S_e + \Lambda, \omega_1 \ldots \omega_{N-1} \right)^T
\]
\[
H_I = \left( \sum_p n \dot{m}_p, \sum_p n \dot{m}_p \dot{u}_p - \frac{4\pi}{3} \rho r^3 \int_p \frac{3}{p} \rho \frac{du_p}{dt}, \sum_p n \dot{m}_p \dot{v}_p - \frac{4\pi}{3} \rho r^3 \int_p \frac{3}{p} \rho \frac{dv_p}{dt}, \right.
\]
\[
\sum_p n \dot{m}_p \dot{w}_p - \frac{4\pi}{3} \rho r^3 \int_p \frac{3}{p} \rho \frac{dw_p}{dt}, \sum_p n \dot{m}_p \dot{h}_s - \frac{4\pi}{3} \rho r^2 \int_p \frac{2}{p} \rho \frac{dh_s}{dt}, \frac{\partial T}{\partial \Delta T}, 0, 0,
\]
\[
\sum_p n \dot{m}_p \dot{0}_p, 0, 0, \ldots, 0, 0 \right)^T
\]
where \( \rho, p, u, v, w, \kappa, \varepsilon \) and \( Y_i \) represent the density, pressure, Cartesian velocity components, turbulent kinetic energy, dissipation rate of turbulent kinetic energy and species mass fraction, respectively; \( E_t = e + 1 / 2 (u^2 + v^2 + w^2) \) is the total internal energy with \( e \) being the thermodynamic internal energy; and \( \omega_i \) is the rate of change of species \( i \) due to chemical reactions. The \( \omega_i \) in the chemical
source term will be described later in Section 2.4. The quantities related to the source term in the turbulent equations are described in Section 2.6. The normal and shear stresses, energy, species, and turbulent diffusion fluxes are given by

\begin{align*}
q_{xi} &= \rho D_{in} \frac{\partial Y_i}{\partial x} \\
\tau_{xx} &= 2 \mu_e \frac{\partial u}{\partial z} - \frac{2}{3} \mu_e \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
q_{yi} &= \rho D_{in} \frac{\partial Y_i}{\partial y} \\
\tau_{yy} &= 2 \mu_e \frac{\partial v}{\partial z} - \frac{2}{3} \mu_e \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \\
q_{zi} &= \rho D_{in} \frac{\partial Y_i}{\partial z} \\
\tau_{zz} &= 2 \mu_e \frac{\partial w}{\partial z} - \frac{2}{3} \mu_e \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right)
\end{align*}

(6)

\begin{align*}
\tau_{xy} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x} \\
\tau_{yx} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial y} \\
\tau_{zx} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial z} \\
\tau_{xy} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial y} \\
\tau_{yx} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial x} \\
\tau_{zx} &= \left( \mu + \frac{\mu_t}{\sigma_k} \right) \frac{\partial k}{\partial z}
\end{align*}

\begin{align*}
q_{se} &= k_e \left( \frac{\partial T}{\partial x} \right) + \rho \sum_{i=1}^{N} h_i D_{in} \frac{\partial Y_i}{\partial x} \\
q_{se} &= k_e \left( \frac{\partial T}{\partial y} \right) + \rho \sum_{i=1}^{N} h_i D_{in} \frac{\partial Y_i}{\partial y} \\
q_{se} &= k_e \left( \frac{\partial T}{\partial z} \right) + \rho \sum_{i=1}^{N} h_i D_{in} \frac{\partial Y_i}{\partial z}
\end{align*}

where \( T, \mu_e, \mu_t, \mu \) and \( k_e \) are the temperature, effective viscosity, turbulent viscosity, molecular viscosity and the effective thermal conductivity, respectively. The vector \( \mathbf{H}_i \) in Equation (1) represents the source term that accounts for the interactions between the gas and liquid phases. The \( n \) is the number of droplets in the \( p \)th characteristic group of droplets. The \( \bar{m}_p \) is the evaporation rate of each particle group; \( \rho_p \) is the liquid density. The \( r_p \) is the droplet radius. The \( u_p, v_p \) and \( w_p \) are the particle velocities for the \( p \)th particle group. The \( h_{\rho} \) is the enthalpy of fuel vapor at the droplet surface, and \( \bar{H} \Delta T \) is the convective heat...
transfer between two phases. Detailed discussions about the liquid phase equations will be described in Section 2.3. In reacting flow calculations, the evaluation of thermo-physical properties is of vital importance. In this code, the values of \( C_p, k, \) and \( \mu_i \) for each species are determined by fourth-order polynomials in temperature, as described in Section 2.7 and Section 2.8.

2.2 All-Mach-Number Formulation

As noted earlier, the two main difficulties that render the compressible flow algorithms ineffective at low Mach numbers are the round-off error caused by the singular pressure gradient term in the momentum equations (the pressure term is of order \( 1/M^2 \) while the convective term is of order unity in the non-dimensional momentum equations) and the stiffness caused by the wide disparities in eigenvalues. To circumvent these two problems regarding the low Mach number calculations, a time preconditioning term has been applied to rescale the system eigenvalues and the pressure variable is decomposed into a constant reference pressure (\( p_{\text{ref}} \)) and a gauge pressure (\( p_g \)). This approach has been successfully applied to solve chemical non-equilibrium viscous flows with and without sprays by Shuen et al.\(^6\) and by Chen and Shuen.\(^7\) After including the all-Mach-number formulation, the resulting Navier-Stokes equations in a conservative form are

\[
\Gamma \frac{\partial}{\partial \tau} (\hat{\mathbf{Q}}) + \frac{\partial}{\partial \xi} \left( E - \hat{\mathbf{E}} \right) + \frac{\partial}{\partial \eta} \left( \tilde{\mathbf{F}} - \hat{\mathbf{F}} \right) + \frac{\partial}{\partial \zeta} \left( \tilde{\mathbf{G}} - \hat{\mathbf{G}} \right) = \hat{\mathbf{H}}_{\xi} + \hat{\mathbf{H}}_{\eta} \tag{7}
\]

where the primitive variable vector \( \hat{\mathbf{Q}} \) and the preconditioned matrix are given as

\[
\hat{\mathbf{Q}} = \left( p_g, u, v, w, h, k, e, Y_1...Y_{N-1} \right)^T \tag{8}
\]
\[ \Gamma = \begin{bmatrix} \frac{1}{\beta} & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\ u & \rho & 0 & 0 & 0 & 0 & \ldots & 0 \\ v & \rho & 0 & 0 & 0 & 0 & \ldots & 0 \\ w & 0 & \rho & 0 & 0 & 0 & \ldots & 0 \\ \frac{H}{\beta} - 1 & \rho u & \rho v & \rho w & \rho & 0 & \ldots & 0 \\ k & 0 & 0 & 0 & \ldots & \rho & \ldots & 0 \\ e & 0 & 0 & 0 & \ldots & \rho & \ldots & 0 \\ Y_1 & 0 & 0 & 0 & \ldots & \rho & \ldots & 0 \\ Y_2 & 0 & 0 & 0 & \ldots & \rho & \ldots & 0 \\ \cdots & \cdots & 0 & \ldots & \rho & 0 \\ \cdots & \cdots & 0 & \ldots & \rho & 0 \\ \frac{Y_{N-1}}{\beta} & 0 & 0 & 0 & 0 & 0 & \ldots & \rho \end{bmatrix} \]  

where \( \tau^* \) is the pseudo-time, \( h = e + p/\rho \) is the specific enthalpy of the gas mixture. The eigenvalues (in the \( \xi \)-direction) in pseudo-time can be obtained from the matrix \( \Gamma^{-1}A \), where \( A \) is the Jacobian \( \partial(\vec{\xi})/\partial\vec{\xi} \). The eigenvalues for a real gas are

\[ \lambda = \left( U, U, \ldots, \frac{1}{2} \left[ U \left( 1 + \frac{\beta}{c^2} \right) \pm \sqrt{U^2 \left( 1 - \frac{\beta}{c^2} \right) + 4\beta (\alpha_1^2 + \alpha_2^2 + \alpha_3^2)} \right] \right) \]  

where \( c \) is the speed of sound, \( \alpha_1 = \xi_{x'}, \alpha_2 = \xi_{y'}, \alpha_3 = \xi_{z'}, \) and \( U \) is the contravariant velocity component defined as \( U = \alpha_1 u + \alpha_2 v + \alpha_3 w \). To obtain well-conditioned eigenvalues, the scaling factor \( \beta \) is taken to be \( \beta = u^2 + v^2 + w^2 \); such that all the eigenvalues will have the same order of magnitude. The definition of
vectors in Equation (7) is identical to those in Equation (1) except that the pressure terms in the momentum equations are replaced by the gauge pressure.

### 2.3 Liquid-Phase Equations

#### 2.3.1 Droplet Motion Equations

The liquid phase is treated by solving Lagrangian equations of motion and transport for the life histories of a statistically significant sample of individual droplets. This involves dividing the droplets into \( n \) groups (defined by position, velocity, temperature and diameter) at the fuel nozzle exit and then computing their subsequent trajectories in the flow. The spray model used in this study is based on a dilute spray assumption which is valid in regions where the droplet loading is low. The liquid fuel is assumed to enter the combustor as a fully atomized spray comprised of spherical droplets. The present model does not account for the effects due to droplet breakup and coalescence processes, which might be significant in a dense spray situation. The Lagrangian equations governing the droplet motion are

\[
\begin{align*}
\frac{dx_p}{dt} &= u_p \\
\frac{du_p}{dt} &= \frac{3}{16} \frac{C_D \mu_s Re_p}{\rho_p r_p^2} (u_p - u_s) \\
\frac{dy_p}{dt} &= v_p \\
\frac{dv_p}{dt} &= \frac{3}{16} \frac{C_D \mu_s Re_p}{\rho_p r_p^2} (v_p - v_s) \\
\frac{dz_p}{dt} &= w_p \\
\frac{dw_p}{dt} &= \frac{3}{16} \frac{C_D \mu_s Re_p}{\rho_p r_p^2} (w_p - w_s)
\end{align*}
\]

where the particle Reynolds number, \( Re_p \), and the drag coefficient, \( C_D \), are defined as
\[ Re_p = \frac{2r_p \rho_d}{\mu_g} \left[ (u_g - u_p)^2 + (v_g - v_p)^2 + (w_g - w_p)^2 \right] \]

\[ C_D = \begin{cases} 
\frac{24}{Re_p} \left( 1 + \frac{Re_p^{2/3}}{6} \right) & \text{for } Re_p < 1000 \\
0.44 & \text{for } Re_p \geq 1000 
\end{cases} \] (12)

The subscript \( g \) represents the gas-phase quantities and \( p \) represents the liquid-phase (or "particle") quantities. Equation (11) and Equation (12) are used to calculate the new droplet positions, and the new droplet velocities at the new locations. A second-order Runge-Kutta scheme was used to integrate Equation (11).

### 2.3.2 Droplet Mass and Heat Transfer Equations

As described by Faeth,\(^\text{14}\) the following correlations were used to approximate the mass and heat transfer coefficients for a single isolated spherical droplet,

\[ \frac{\dot{m}^*_p}{\rho D_f} = N_s (1 + B) \] (13)

\[ \frac{\bar{h} r_p}{k} = \frac{N_p \ln (1 + B) L e^{-1}}{[ (1 + B) L e^{-1} - 1]} \] (14)

where \( \bar{h} \) is the heat transfer coefficient, \( d_p \) is the droplet diameter, \( D_f \) is the fuel mass diffusivity, \( k \) is the thermal conductivity of fuel vapor and \( \dot{m}^*_p \) is the fuel mass evaporation rate per unit area. The \( N_s \) and \( N_p \) are defined as,

\[ N_s = 1 + \frac{0.276 Re_p^{1/2} Pr^{1/3}}{\left[ 1 + \frac{1.232}{Re_p Pr^{3/4}} \right]^{1/2}} \]

\[ N_p = 1 + \frac{0.276 Re_p^{1/2} Sc^{1/3}}{\left[ 1 + \frac{1.232}{Re_p Sc^{3/4}} \right]^{1/2}} \] (15)
where $Sc$ and $Le$ are the Schmidt and Lewis numbers, respectively. The Spalding number, $B$, in Equation (13) and Equation (14) is defined as,

$$B = \frac{Y_{fsp} - \bar{Y}_g}{1 - Y_{fsp}}$$

(16)

where $Y_{fsp}$ is the fuel vapor mass fraction at the surface of the droplet and $\bar{Y}_g$ is the mean fuel mass fraction of the ambient gas. In the present study, $Y_{fsp}$ is obtained from the following equation,

$$Y_{fsp} = \frac{X_{fsp} W_f}{X_{fsp} W_f + (1 + X_{fsp}) W_a}$$

(17)

where $W_a$ is the molecular weight of gas mixture excluding fuel vapor, $W_f$ is the molecular weight of fuel and $X_{fsp}$ is the mole fraction of fuel. The $X_{fsp}$ is obtained from the assumption of Raoult’s law. Based on this assumption, the mole fraction at the droplet surface is equal to the ratio of the partial pressure of fuel vapor to the total pressure. For the present spray calculation, the partial pressure of fuel vapor is computed using the empirical correlation in Reid.\(^{17}\)

$$\ln\left(\frac{p_{x}}{p_c}\right) = \frac{1}{1-x}\left(P_{v_1}x + P_{v_2}x^{1.5} + P_{v_3}x^3 + P_{v_4}x^6\right)$$

(18)

where $x=1-T/T_c$, $T_c$ and $P_c$ are the critical temperature and critical pressure of the fuel vapor, respectively. The constants, $P_{v_1}$, $P_{v_2}$, $P_{v_3}$, and $P_{v_4}$ are obtained from Reid.\(^{17}\)

2.3.3 Droplet Internal Temperature Equations

As a single droplet enters a hot environment, the small portion of the droplet near the surface will heat up quickly while the center core of the droplet remains “cold.” The heat will be conducted and convected to the droplet’s interior as the droplet penetrates further into the hot ambient gas. Eventually, the temperature
within the droplet will become nearly uniform before the end of its lifetime. To solve this transient process occurring within the droplet is not trivial. In the past, certain approximations were usually made to alleviate this computational burden while retaining the solution integrity. The simplest approximation is the uniform temperature model. This model assumes that the thermal conductivity of the fuel is infinite. Of course, this is not valid at the beginning of the heating-up process of the droplet. Another model considers the heat diffusion inside the droplet, yielding a conduction model. The temperature distribution within the droplet is obtained by solving the one-dimensional heat conduction equation subject to the convective boundary conditions at the droplet surface. The conduction model completely neglects the convective phenomena within the droplet which might occur due to significant shear forces at the surface induced by high slip velocities. Tong and Sirignano\textsuperscript{15} developed a vortex model for the internal temperature of a single droplet which accounts for the convective effect of the Hill's vortex formed inside the droplet. Tong and Sirignano's vortex model is used in ALLSPD-3D code to obtain the internal temperature distribution of the droplet. The equations governing the internal temperature distribution based on this model are

\[
\frac{\partial T_p}{\partial t} = \frac{17 k_l}{C_p \rho_l r_p^2} \left[ \frac{\partial^2 T_p}{\partial \alpha^2} + \left( 1 + C(t) \alpha \right) \frac{\partial T_p}{\partial \alpha} \right] \quad \text{and} \quad C(t) = \frac{3}{17} \frac{(C_p \rho_l r_p)}{k_l} \frac{dr_p}{dt} \quad (19)
\]

where the value of \( \alpha \) varies between 0 and 1 with \( \alpha = 0 \) referring to the vortex center and \( \alpha = 1 \) referring to the droplet surface. The initial and boundary conditions for Equation (19) are

\[
\begin{align*}
t &= t_{\text{inj}} \\
T_p &= T_{\text{inj}} \\
\frac{\partial T_p}{\partial \alpha} &= 1 \left( \frac{C_p \rho_l}{k_l} \right) \frac{r_p \partial T_p}{\partial t} \\
\alpha &= 0 \\
\frac{\partial T_p}{\partial \alpha} &= \frac{3}{16} \left( \frac{\partial T_p}{\partial t} \right)_s \\
\alpha &= 1
\end{align*}
\]

(20)
where $\partial T_p/\partial \alpha$ is obtained from the energy balance at the droplet surface,

$$\left(\frac{\partial T_p}{\partial r}\right)_s = \frac{1}{k_i} \left[ h \Delta (T - \dot{m}_p h_{fg}) \right]$$

(21)

where $\dot{m}_p$ and $h$ are calculated from Equation (13) and Equation (14), respectively, $h_{fg}$ is the latent heat of the fuel and $\Delta T = T_s - T_{ps}$, where $T_{ps}$ is droplet surface temperature and $\tilde{T}_s$ is the mean gas temperature evaluated as,

$$\tilde{T}_s = \frac{1}{3} T_s + \frac{2}{3} T_{ps}$$

(22)

An implicit scheme is used to solve Equation (19) subject to the initial and boundary conditions. Second-order central difference is used for the spatial differential terms and a first-order difference is used for the time term. These treatments render a scalar tridiagonal algebraic system and is solved by the Thomas algorithm.\(^{18}\)

### 2.4 Chemical Source Term

For a set of $N_r$ elementary reactions involving $N$ species, the reaction equations can be written in the following general form,

$$\sum_{i=1}^{N} v_{ik}^r CS_i \rightarrow \sum_{i=1}^{N} v_{ik}^w CS_i \quad , k = 1,2,3 ... Nr$$

(23)

where $v_{ik}^r$ and $v_{ik}^w$ are the stoichiometric coefficients for species $i$ appearing as a reactant in the $k^{th}$ forward and backward reactions, respectively. The law of mass action states that the rate of change of concentration of the $i^{th}$ species is defined by the movement of each species summed over all of the contributing reactions. Therefore, the rate of change in species concentration is expressed as follows,

\[ \]
\[ \omega_i = MW_i \sum_{k=1}^{N_r} \left[ CR_{ik} \left( K_f(k) \prod_{j=1}^{N} C_j^{\nu jk} - K_b(k) \prod_{j=1}^{N} C_j^{\nu' jk} \right) \right] \] (24)

The forward and backward reaction rate constants \((K_f, K_b)\) are computed from Arrhenius' relationship with phenomenologically developed forms. The forward and backward reaction rate constants of the \(k^{th}\) reactions are defined as,

\[ K_b(k) = A_b(k) T^{\alpha_b(k)} \exp \left( \frac{E_b(k)}{R_u T} \right) \]

\[ K_f(k) = A_f(k) T^{\alpha_f(k)} \exp \left( \frac{E_f(k)}{R_u T} \right) \] (25)

\[ CR_{ik} = \left( \nu''_{ik} - \nu'_{ik} \right) \] (26)

The influence of other bodies on a particular chemical reaction is modeled using a typical third body efficiency correction,

\[ K_f(k) = K_f(k) \sum_{i=1}^{N} \text{Third}_i(k) C_i \] (27)

\[ K_b(k) = K_b(k) \sum_{i=1}^{N} \text{Third}_i(k) C_i \]

It is noted that, because of the vastly different time scales that may be involved in the elementary reactions and the flow, and the strong dependence of the source terms on temperature (exponentially) and density \((p^2\) or \(p^3\), depending on the order of reaction), the set of equations may become very stiff for most flow conditions of interest. To mitigate this problem, chemical source terms are treated in an implicit fashion in the ALLSPD-3D code to improve numerical
stability. Details for the construction of the source term Jacobian are described in Appendix A.

2.5 Equilibrium Rate Kinetics Model

There is a large database of finite-rate chemistry data where only one of the two required reaction rate constants is reported. The chemical rate information in the secondary direction, either forward or backward direction, can be determined from an equilibrium assumption. Typically, the equilibrium constant \( K_p \) for a reaction can be found in many standard chemistry references as a function of temperature. However, it is convenient to determine \( K_p \) as a part of the solution. This can be done by taking advantage of the large thermodynamic information available in the NASA Lewis database.\(^{16}\) The equilibrium rate information can be determined for a large number of hydrocarbon related species using the following procedure.

In an equilibrium state, it is known that the partial pressure ratio of the species component of an ideal mixture is equal to the equilibrium constant \( (K_p) \).\(^{34}\) In equilibrium \( \Delta G \)\( _T, p \) is zero and therefore only the Gibbs free energy of formation for each species is needed to determine the equilibrium constant. For a simple reaction, a \( \text{CS(a)} + b \text{CS(b)} \leftrightarrow c \text{CS(c)} + d \text{CS(d)} \), the equilibrium constant in terms of partial pressures (nondimensionalized by atmospheric pressure) is defined as,

\[
    K_p = \frac{[P_a]^a [P_b]^b}{[P_c]^c [P_d]^d} = \exp \left( \frac{-\Delta G^0}{R_a T} \right)
\]

(28)

where \( \Delta G^0 \) is defined as standard free energy change. This relationship can be extended to any general chemical reaction using the standard free energy change expressed in terms of molar Gibbs free energy \([\text{kcal/kg-kmole}]\). The molar Gibbs free energy is defined by the following expression,
\[ \Delta fG_r^0 = \sum_{n=1}^{N} \nu_{nr} \Delta g_{nr}^0 - \sum_{n=1}^{N} \nu_{nr} \Delta g_{nr}^0 \]  \hspace{1cm} (29)

where the partial molar Gibbs free energy of formulation of \( n \)-th species, \( \Delta g_{nr}^0 \), consistent with the polynomial representations of Gordon and McBride is,

\[ \Delta g_{n}^0(T) = h - Ts = C_{p_n}^0 \left( 0 \right) \left( 1 - \ln(T) \right) T - C_{p_n}^1 \frac{T^2}{2} - C_{p_n}^2 \frac{T^3}{6} \]

\[- C_{p_n}^3 \frac{T^4}{12} - C_{p_n}^4 \frac{T^5}{20} + L_{p_n}^5 \frac{T^6}{12} \]  \hspace{1cm} (30)

\[ \Delta fG_r^0(T) = \sum_{n=1}^{N} \text{CR}_{nr} \Delta g_{n}^0(T) \]  \hspace{1cm} (31)

\[ K_{P_r} = \exp \left( \frac{-\Delta fG_r^0(T)}{R_u T} \right) = \prod_{i=1}^{N} \frac{\nu_{ir}^*}{\nu_{ir}^*} \left( \frac{1}{p_{am}} \right) \text{CR}_{ir} \]  \hspace{1cm} (32)

The equilibrium constant in terms of partial pressures is converted to the molar based constant required for Equation (24) using the following expression,

\[ K_{C_r} = K_{P_r} \left( \frac{1}{R_u T} \sum_{i=1}^{N} \text{CR}_{ir} \right) \]  \hspace{1cm} (33)

Once the forward reaction rate constants are known, the backward reaction rate constant is,

\[ K_{br} = K_{fr} / K_{C_r} \]  \hspace{1cm} (34)
2.6 Two Equation Model of Turbulence

In general, there are several types of low-Reynolds number k-ε models and the major distinguishing feature of these models is the handling of the no-slip wall singularity that exists in the destruction term of the specific dissipation equation. The most common practice is to use a modified form of the transport equations, where a fictitious dissipation($\tilde{\varepsilon}$) rather than the real dissipation(ε) is used. The fictitious dissipation is usually defined as $\tilde{\varepsilon} = \varepsilon - \varepsilon_{\text{wall}}$. There are three types of low-Reynolds number k-ε models. The first type uses flowfield characteristics, such as curvature of the turbulent kinetic energy($\varepsilon_{\text{wall}} = 2\nu(d^2k/dy^2)^2$), to satisfy the near wall asymptotic behavior, while at same time avoiding the singularity caused by the no-slip boundary condition. The early k-ε model formulation proposed by Launder and Sharma\textsuperscript{24} is typical of this type of formulation. The second type of low-Reynolds number k-ε model, proposed by Chien\textsuperscript{25} uses the wall normal distance($\varepsilon_{\text{wall}} = 2\nu k/y_n^2$) and Van-Driest type damping to develop the necessary near wall characteristics. The third type of low-Reynolds number formulation makes no attempt to modify the transport equation; rather, finite near-wall characteristics are used as boundary conditions. The low Reynolds number model proposed by Shih\textsuperscript{12} is typical of the models that avoid the zero boundary condition and therefore the singularity. All three employ additional wall damping to satisfy the near wall physics.

In practice, the low-Reynolds number k-ε models are easier to implement numerically than high Reynolds number models with wall function boundary conditions for complex geometries because of the simple boundary conditions. However, all of the turbulence models have limitations imposed by the numerical implementation and physics.\textsuperscript{28} Therefore, the two-equation model solver in ALLSPD-3D has been designed to be generic and modular.

Presently, the low-Reynolds number model developed by Shih, et al.\textsuperscript{12} of CMOTT has been incorporated into ALLSPD-3D as the default turbulence model. However, the modular design of the numerical implementation allows
the exact form of the low Reynolds number characteristics to be changed with only minor modifications. Furthermore, several additional numerical options are incorporated to address the numerical stiffness caused by the high near-wall grid resolution. The typical form of the source term required for the Shih et. al. model is given by the following,

\[
\begin{bmatrix}
S_k \\
S_\varepsilon \\
C_{\varepsilon} \\
\end{bmatrix} = \begin{bmatrix}
P_k - \rho \varepsilon \\
\rho \varepsilon \\
\end{bmatrix} + \Lambda
\]

where the exact turbulent production, \(P_k\), is,

\[P_k = \overline{\rho \tau_{ij} \left[ \frac{\partial u_i}{\partial x_j} \right]}\]

\[\overline{\rho \tau_{ij}} = \mu_t \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] - \frac{2}{3} \sigma_k \sigma_{ij}\]

Here a linear stress-strain relationship is used to model the Reynolds’ stress components. The low-Reynolds number corrections required for the source term are given by the following expression,

\[
\Lambda = \begin{bmatrix}
0 \\
-2\mu_t (\nabla u)^2
\end{bmatrix}
\]

The modeling constants \(f_\varepsilon, C_{\varepsilon1},\) and \(C_{\varepsilon2}\) are summarized in Table 2.

<table>
<thead>
<tr>
<th>Model</th>
<th>(C_{\varepsilon1})</th>
<th>(C_{\varepsilon2})</th>
<th>(C_{\mu})</th>
<th>(\sigma_\kappa)</th>
<th>(\sigma_\varepsilon)</th>
<th>(f_\varepsilon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shih et. al.</td>
<td>1.44</td>
<td>1.92</td>
<td>0.09</td>
<td>1.00</td>
<td>1.30</td>
<td>1-0.22exp(-R_c^2/36)</td>
</tr>
</tbody>
</table>

**TABLE 2. Turbulence Model Parameters**
The eddy viscosity is defined by

\[ \mu_i = C_{\mu_i} \left( 1 - \exp \left( a_1 R_i + a_2 R_i^2 + a_3 R_i^3 \right) \right) \rho \frac{k^2}{\varepsilon} \]  

(39)

and the modeled damping coefficients are \( a_1 = -1.7 \times 10^{-3} \), \( a_2 = -1 \times 10^{-9} \), and \( a_3 = -5 \times 10^{-10} \).

The eddy viscosity constant can also be varied as a function of the mean shear as suggested by Shih. The formulation of a variable eddy viscosity constant has been developed to satisfy realizability criteria and has been shown to be more accurate for non-homogeneous flows and wall bounded flow situations.\(^{12}\)

\[ C_\mu = \min \left( 0.09, \frac{2/3}{A_0 + (k/\varepsilon) \sqrt{2 S_i S_j}} \right) \quad \text{where} \quad S_i = \left\{ \frac{1}{2} \left[ \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right] - \frac{1}{3} \left[ \frac{\partial u_k}{\partial x_i} \delta_{ij} \right] \right\} \]  

(40)

### 2.7 Transport Property Model

All of the transport properties used in ALLSPD-3D have been derived from the database of the physical property information developed by Gordon and McBride.\(^{16}\) However, in order to reduce complexity, property curves for the diffusion and conductivity coefficients are converted into single forth order polynomial curves. The typical diffusion property curves for the \( i \)\(^{th} \) species are defined by the following expressions,

\[ \mu_i = \sum_{k=0}^{4} C_{\mu_i}^{(k)} T^k \quad \kappa_i = \sum_{k=0}^{4} C_{\kappa_i}^{(k)} T^k \]  

(41)

The \( C_{\mu_i}^{(k)} \) constants have the dimensional property of \([\text{kg/(m-s)/K}^k]\), and \( C_{\kappa_i}^{(k)} \) constants have the dimensional property of \([\text{Joules/(m-s-K)/K}^k]\). Once
the individual species viscosities are found, the mixture viscosity and conductivity are computed using Wilke's mixing rule,

\[ \mu = \sum_{i=1}^{N} \mu_i \left( \frac{1}{X_i} \sum_{j=1}^{N} X_j \phi_{ij} \right) \quad \kappa = \sum_{i=1}^{N} \kappa_i \left( \frac{1}{X_i} \sum_{j=1}^{N} X_j \phi_{ij} \right) \]  

(42)

Here, the inter-collisional parameter(\(\phi\)) between species \(i\) and \(j\) is defined as

\[ \phi_{ij} = \left( 2 \sqrt{\frac{1}{8} \left( 1 + \frac{MW_i}{MW_j} \right)} \right)^{-1} \left( 1 + \frac{MW_i}{MW_j} \frac{\mu_i}{\mu_j} \right)^2 \]  

(43)

The inter-species diffusion, between the species \(i\) and \(j\), is modeled using the Chapman-Enskog\(^{34}\) formula for dilute gases,

\[ D_{ij} = 0.0018583 \frac{1}{\sqrt{WM_i} + \sqrt{WM_j}} \left( \frac{T_{1.5}}{P\Omega_{ij}} \right) \]  

(44)

where the effective collision integral factor, \(\Omega_{ij}\), and the effective temperature, \(T_D\), are approximated by

\[ \Omega_{ij} = \frac{1}{T_{D0.145}} + \frac{1}{(T_D + 0.5)^2} \]

\[ T_D = \frac{T}{\sqrt{T_e(i) T_e(j)}} \quad \sigma_{ij} = \frac{1}{2} (\sigma_i + \sigma_j) \]  

(45)

Once, the individual species diffusivity properties have been determined, the overall diffusion coefficients are computed by lumping all of the species together into a mixture and then computing the binary diffusion between the \(i^{th}\) species and the mixture. An approximate form of the diffusion coefficient is as follows,
\[ D_{im} = \left( 1 - X_i \right) \left( \sum_{j \neq i}^{N} \frac{X_j}{D_{ij}} \right) \] (46)

Once, the binary diffusion coefficients, \( D_{ij} \), for all of the species combinations are known, the diffusion velocity of each species is found using Fick's law. The species diffusion velocity components, \( \hat{u}_{k}(Y_i) \), are defined as

\[ \hat{u}_{k}(Y_i) = -D_{im} \frac{\partial Y_i}{\partial x_k} \] (47)

The turbulent diffusion of momentum, mass and heat are modeled based on the assumption that the eddy diffusivity is directly proportional to the turbulent diffusivity by a constant, which is the turbulent Schmidt number (\( Sc_t \)) for mass diffusion and the turbulent Prandtl number (\( Pr_t \)) for thermal diffusion. Once combined with the mean properties, effective diffusion coefficients can be extracted and defined as follows,

\[ \mu_{\text{eff}} = \mu + \mu_t \quad \left( D_{im}\right)_{\text{eff}} = D_{im} + \frac{\mu_t}{\rho Sc_t} \] (48)

\[ \kappa_{\text{eff}} = \kappa + \frac{\mu_t C_{p_n}}{P_r t} \]

The turbulent Schmidt number and the turbulent Prandtl number are assumed to be equal to 0.90.

### 2.8 Thermodynamic Model

The mass-averaged constant pressure specific heat constant (\( C_{p_i} \)) is assumed to be a fourth order polynomial function of temperature. The curve fitted constants have been adopted from Gordon and McBride. Once the individual species specific heat (\( C_{p_i} \)) have been found, the mixture specific heat (\( C_{p_m} \)) can
be found by weighting over all of the species,

\[ C_{p_i} = \sum_{k=0}^{4} C_{p_i} (k) T^k \quad \quad \quad C_{p_m} = \sum_{i=1}^{N} C_{p_i} Y_i \tag{49} \]

The constants have the dimensional property of [Joules/kg/K^4]. The internal energy of the gas mixture, found by using mixture properties such as pressure, density and enthalpy, is defined as

\[ e = \sum_{i=1}^{N} Y_i h_i - \frac{E}{\rho} \quad h_i = h_f + \int_{T_{ref}}^{T} C_{p_i}dT \tag{50} \]

where \( h_f \) is an integration constant to satisfy the reference temperature condition of Gordon and McBride.\(^{16}\) Since the specific heats, \( C_{pi} \), are formulated as a function of temperature only, the enthalpy relationship can be easily integrated to yield a polynomial relationship between enthalpy and temperature for the gas mixture. Newton's iteration is used to find the temperature from the predicted internal energy of the gas mixture. The pressure of the gas mixture is determined from Dalton's law of partial pressure,

\[ p = \rho R_u T \sum_{i=1}^{N} \frac{Y_i}{MW_i} \tag{51} \]

2.9 Turbulent Chemistry Model

The influence of turbulent intermittency on the finite rate chemistry is modeled using a simple eddy dissipation concept. The simple formulation of Magnussen and Hjertager\(^{13}\) is employed where the intermittency of the turbulent flame is modeled in terms of rates of dissipation of eddies. Here the rate of eddy dissipation is related to the mean concentrations of fuel species and the turbulent
time scales, where the turbulent time scale is defined as the ratio of the turbulent kinetic energy to specific dissipation. The eddy dissipation concept offers one distinct advantage over the eddy break up model concept of Spalding\(^{36}\) in that the modeled term is directly related to the mean species quantities and does not require additional modeling of species variance terms(\(\sqrt{\bar{Y}_i^2}\)). This turbulent chemistry model is a first order correction that has been shown to be accurate in combustor applications.\(^{31}\)

The species source term modeling the mass transfer between the species shown in Equation (24) is replaced with the following,

\[
\bar{w}_i = C_{ebm} \rho Y_{ik}^{e} \quad \omega_i = \text{Min}(\omega_i^{\text{Laminar}}, \omega_i^{\text{fuel}}, \omega_i^{\text{oxy}})
\]  

(52)
III: Numerical Methods

Equation (7) is the final gas-phase governing equation to be solved numerically. As mentioned in the introduction, solving the coupled three-dimensional equations numerically with "many" species using a reasonable number of grid points (say, a quarter million) in a fully coupled manner is impractical. The main problem with using the fully coupled approach, like the one in Chen et al., is that it requires a large amount of storage for the left-hand-side (LHS) coefficient matrix. Although great effort has been expended to take advantage of several storage saving tactics, this problem has not been fully solved. As the number of species and the grid density increase, this problem becomes more difficult to resolve.

Although it is believed that the fully coupled, strongly implicit algorithm is the best choice (as far as the stability, robustness and convergence are concerned) for stiff problems typical of combustion flows, it is not very economical or affordable for the practical complex problems found in industry. Other partially implicit and decoupled (sequential) schemes, such as TEACH, can experience difficulties in handling reacting flows, especially with sprays. There is currently a wide spectrum of numerical schemes, whose applicability depends on the specific problem of interest. The advantages of a particular scheme may be the disadvantages of others. In the area of three-dimensional combustion flow simulation, however, the freedom of choice is very narrow.

The approach in ALLSPD-3D employs a fully coupled, strongly implicit algorithm with a slight modification to allow more efficient computation for three-dimensional reacting flows. The partially decoupled procedure described here is designed to accommodate more species and more grid points in three-
dimensional combustion calculations. It is believed that this approach represents a reasonable trade-off between various numerical approaches available for the computation of finite-rate combustion flows. As the growth of computer technology in terms of memory and speed progresses, the advantages of the current approach will become more pronounced.

In ALLSPD-3D, the physical time (second term in Equation(7)) term has been eliminated and the solution marches forward in pseudo-time to steady state. The fully coupled governing equations, Equation(7), are partially decoupled into three sets of equations, i.e., five flow equations, the \( \kappa \)-\( \epsilon \) turbulent equations and \((N-1)\) species equations. The numerical scheme used in this code solves each set of equations in a coupled manner and iteratively. This process is depicted in Figure 1.

The governing equations for each set are similar and have the following general form:

\[
\Gamma^{(m)} \frac{\partial}{\partial \tau} (\dot{\xi}^{(m)}) + \frac{\partial}{\partial \xi} \left( \dot{E}^{(m)} - \dot{E}_V^{(m)} \right) + \frac{\partial}{\partial \eta} \left( \dot{F}^{(m)} - \dot{F}_V^{(m)} \right) + \frac{\partial}{\partial \zeta} \left( \dot{G}^{(m)} - \dot{G}_V^{(m)} \right) = \ddot{H}_c^{(m)} + \ddot{H}_l^{(m)} + \ddot{D}_c^{(m)}
\]
where \( m=1 \) for flow equations, \( m=2 \) for turbulent equations, and \( m=3 \) for species equations. For example,

\[
\hat{Q}^{(1)} = \left( \frac{1}{\bar{\beta}}(p, u, v, w, k) \right)^T
\]

and

\[
\hat{Q}^{(2)} = \left( \frac{1}{\bar{\beta}}(k, \varepsilon) \right)^T \quad \hat{Q}^{(3)} = \left( \frac{1}{\bar{\beta}}(Y_1, \ldots, Y_{N-1}) \right)^T
\]

\[
\Gamma^{(1)} = \begin{bmatrix}
\frac{1}{\bar{\beta}} & 0 & 0 & 0 & 0 \\
\frac{u}{\bar{\beta}} & \rho & 0 & 0 & 0 \\
\frac{v}{\bar{\beta}} & 0 & \rho & 0 & 0 \\
\frac{w}{\bar{\beta}} & 0 & 0 & \rho & 0 \\
H & -1 & \rho u & \rho v & \rho w & \rho
\end{bmatrix}
\]

\[
\Gamma^{(2)} = \begin{bmatrix}
\rho & 0 \\
0 & \rho
\end{bmatrix} \quad \Gamma^{(3)} = \begin{bmatrix}
\rho & \ldots \\
\ldots & \rho & 0 \\
\ldots & \ldots & 0 \\
\ldots & \ldots & \ldots & \rho
\end{bmatrix}
\]

Other terms in Equation (53) are straightforward to obtain from Equation (7) and are not listed here. The spray source terms, \( \hat{H}^{(m)} \), used to model the mass transfer are described in detail in Section 3.3.
\[
H_l^{(1)} = \frac{1}{j} \left( \sum_{p} n_p \dot{m}_p + \sum_{p} n_p \dot{m}_u p \frac{4\pi}{3} \frac{du}{dt} p \sum_{p} n_p \dot{m}_v p \frac{4\pi}{3} \frac{dv}{dt} p \sum_{p} n_p \dot{m}_w p \frac{4\pi}{3} \frac{dw}{dt} p \sum_{p} n_p \dot{m}_h p \frac{4\pi}{3} \frac{2}{r} \sum_{p} n_p h\Delta T p \right)
\]

\[
H_l^{(2)} = \frac{1}{j} \left( \begin{array}{c}
\sum_{p} n_p \dot{m}_p \\
0 \\
0 \\
0 \\
\end{array} \right) \quad H_l^{(3)} = \frac{1}{j} \left( \begin{array}{c}
0 \\
0 \\
0 \\
0 \\
\end{array} \right)
\]

The \( \tilde{D}^{(m)} \) term that appears in Equation (53) represents a source term arising from the partially decoupled procedure and is expressed as,

\[
\tilde{D}_c^{(1)} = \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}, \quad \tilde{D}_c^{(2)} = \frac{1}{\beta \partial \tau} \left( \frac{p_s}{j} \right) \begin{bmatrix}
k \\
0 \\
0 \\
0 \\
0
\end{bmatrix}, \quad \tilde{D}_c^{(3)} = \frac{1}{\beta \partial \tau} \left( \frac{p_s}{j} \right) \begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
Y_4
\end{bmatrix}
\]
3.1 LU Scheme

After linearization and applying a first-order time differencing to the pseudo-time term, Equation (53) can be expressed in the following form,

\[
\left[ \Gamma^{(m)} - \Delta \tau \left( T^{(m)} + \frac{\partial}{\partial \xi} A^{(m)} + \frac{\partial}{\partial \xi} R^{(m)} \right) \right] + \left( \frac{\partial}{\partial \eta} + \frac{\partial}{\partial \eta} \right) R^{(m)} \frac{\partial}{\partial \eta} + \left( \frac{\partial}{\partial \xi} C^{(m)} \right) = -\Delta \tau (R^{(m)})^p
\]

where,

\[
(R^{(m)})^p = \left( \begin{array}{c} \frac{\partial E^{(m)} - \bar{E}_v^{(m)}}{\partial \xi} \\ \frac{\partial F^{(m)} - \bar{F}_v^{(m)}}{\partial \eta} \\ \frac{\partial G^{(m)} - \bar{G}_v^{(m)}}{\partial \xi} \\ -\bar{H}_c^{(m)} - \bar{H}_t^{(m)} - \bar{D}_c^{(m)} - \bar{D}_4^{(m)} \end{array} \right)
\]

where the superscript \( p \) denotes the previous iteration level, and \( S^{(m)} \) and \( T^{(m)} \) are the Jacobian matrices for chemical and turbulent source terms, respectively. \( A^{(m)}, B^{(m)} \) and \( C^{(m)} \) are the inviscid term Jacobians and \( R^{(m)}_\xi, R^{(m)}_\eta \) and \( R^{(m)}_\xi \) are the viscous Jacobians and \( \Delta \hat{Q}^{p+1} = \Delta \hat{Q}^{p} + \Delta \hat{Q}^{(m)} \). The expressions for these Jacobians are given in Appendix A. \( D^{(m)}_4 \) is the fourth order dissipation term that will be defined later. The lower-upper symmetric Gauss-Seidel (LU-SGS) scheme described here was originally developed by Yoon and Jameson for Euler and Navier-Stokes calculations without preconditioning. With the current time preconditioning, the LU-SGS scheme has to be modified to incorporate the preconditioning matrix. Consider Equation (53), for example. The LHS inviscid terms, which include the preconditioning matrix, need to be modified. For example, in the \( \xi \) direction,

\[
\frac{\partial A^{(m)}}{\partial \xi} = \frac{\partial}{\partial \xi} \left( \Gamma^{(m)} A^{(m)} \right) = \frac{\partial}{\partial \xi} \left( \Gamma^{(m)} A^{(m)} \right)
\]

where \( A^{(m)} = \left( \Gamma^{(m)} A^{(m)} \right) \). \( A^{(m)} \) is then split into two parts,
\[ A^{(m)} = \frac{1}{2} \left( \hat{A}^{(m)} + \alpha |\lambda_{A_{max}}| I \right) + \frac{1}{2} \left( \hat{A}^{(m)} - \alpha |\lambda_{A_{max}}| I \right) \]

\[ = \hat{A}^{(m) \ast} + \hat{A}^{(m)} \]

where \( |\lambda_{A_{max}}| \) is the absolute value of the maximum eigenvalue in the \( \xi \) direction at each grid point, \( \alpha \) is a parameter greater than 1, and \( I \) is the unit matrix. The term \( \partial A^{(m)} / \partial \xi \) in Equation (59) is then approximated by the following one-sided operator,

\[ \frac{\partial A^{(m)}}{\partial \xi} = \hat{\partial} \left( \Gamma^{(m)} \hat{A}^{(m) \ast} \right) + \hat{\partial}^{\ast} \left( \Gamma^{(m)} \hat{A}^{(m)} \right) \quad (60) \]

where

\[ \Gamma^{(m)} \hat{A}^{(m) \ast} = \frac{1}{2} \Gamma^{(m)} \left( \hat{A}^{(m) \ast} + \alpha |\lambda_{A_{max}}| \right) \]

\[ = \frac{1}{2} \left( \hat{A}^{(m) \ast} + \alpha |\lambda_{A_{max}}| \Gamma^{(m)} \right) \]

The operators, \( \partial / \partial \xi \) and \( \partial^{\ast} / \partial \xi \) in Equation (60) are the first order backward and forward differencing operators, respectively. Similar procedure can be applied to \( \partial B^{(m) / \partial \eta} \) and \( \partial C^{(m) / \partial \zeta} \). It can be easily shown that the effect of the above eigenvalue augmentation is equivalent to adding a second order central-differencing dissipation term (scaled by the local maximum eigenvalues) to the implicit operator. Therefore, the discretized governing equations for LU-SGS scheme can be written as follows,

\[
\begin{bmatrix}
\Gamma^{(m)} - \Delta \tau^{\ast} R^{(m)} + \left( \frac{\partial}{\partial \xi} A^{(m)} - \frac{\partial}{\partial \xi} R^{(m)} \right) \frac{\partial}{\partial \xi} + \left( \frac{\partial}{\partial \eta} B^{(m)} - \frac{\partial}{\partial \eta} R^{(m)} \right) \frac{\partial}{\partial \eta} \\
\left( \frac{\partial}{\partial \zeta} C^{(m)} - \frac{\partial}{\partial \zeta} R^{(m)} \right) \frac{\partial}{\partial \zeta} + D^{(m)} \right] \Delta Q^{(m)} = -\Delta \tau^{\ast} \left( R^{(m)} \right)_{p}
\end{bmatrix}
\]

where the second and fourth order dissipation terms are defined as,
\[ D_2^{(m)} = -0.5\alpha \left[ \frac{\partial^2}{\partial \xi^2} (|\lambda_A|\Gamma^{(m)}) + \frac{\partial^2}{\partial \eta^2} (|\lambda_B|\Gamma^{(m)}) + \frac{\partial^2}{\partial \zeta^2} (|\lambda_C|\Gamma^{(m)}) \right] \]

\[ D_4^{(m)} = \sigma \Gamma^{(m)} \left[ |\lambda_A| \frac{\partial^4}{\partial \xi^4} \hat{Q}^{(m)} + |\lambda_B| \frac{\partial^4}{\partial \eta^4} \hat{Q}^{(m)} + |\lambda_C| \frac{\partial^4}{\partial \zeta^4} \hat{Q}^{(m)} \right] \]

where \(|\lambda_A^{(m)}|, |\lambda_B^{(m)}|\) and \(|\lambda_C^{(m)}|\) are the absolute values of the maximum eigenvalues in the \(\xi, \eta\) and \(\zeta\) directions, respectively. The parameter \(\alpha\) and \(\sigma\) are used to define the strength of the dissipation. Here, the eigenvalues for the flow equations are identical to those previously defined. The eigenvalues for turbulent and species equations are simply the contravariant velocity components. The LU-SGS scheme utilizes the following factorization procedure to obtain the solution,

\[ M\Delta \hat{Q} = b \]  \hspace{2cm} (61)

\[ (L + D) D^{-1} (U + D) \Delta \hat{Q} = b \]  \hspace{2cm} (62)

Equation (62) can be solved in the following steps,

Step 1 \hspace{1cm} (L + D) \Delta \hat{Q}^* = b

Step 2 \hspace{1cm} (U + D) \Delta \hat{Q} = D(D\Delta \hat{Q}^*)

Step 3 \hspace{1cm} \Delta \hat{Q}^{p+1} = \Delta \hat{Q}^p + \Delta \hat{Q}

where \(M\) and \(b\) are the LHS and RHS of Equation (53), respectively. The \(L, U\) and \(D\) are the non-zero, off-diagonal lower, upper and diagonal part of matrix \(M\), respectively. The LU-SGS scheme can be vectorized along a constant \((i+j+k)\) oblique planes (see Figure 2). Therefore, the \(n\times n\) LHS coefficient blocks are stored only for the two-dimensional oblique plane. Here, \(n\) is the number of the equations to be solved.
3.2 Mass Conservation

The first set of equations in the partially decoupled procedure is straightforward to solve. There are no numerical differences between this set of equations and those in the fully coupled approach\(^7\) except that only five equations are solved. However, when the turbulent equations or species equations are solved, there is an extra source term, i.e., \(D_c^{(2,3)}\). These terms consist of a common factor, i.e., \(\left( \partial (p_g/\gamma) / \partial \tau^* \right) / \beta\), which has to be evaluated from the continuity equation in order to conserve mass in a numerical sense. Since the continuity equation can be written as,

\[
\frac{1}{\beta} \frac{\partial}{\partial \tau} \left( \frac{p_g}{J} \right) + \nabla \cdot (\rho \mathbf{V}) = \sum_p n_p \dot{m}_p
\]

The, \(\left( \partial (p_g/\gamma) / \partial \tau^* \right) / \beta\) term can be evaluated as:
\[
\frac{1}{\rho} \frac{\partial}{\partial t} \left( \rho \frac{\mathbf{v}}{f} \right) = -\nabla \cdot \left( \rho \mathbf{v} \right) + \sum_p n_p \dot{m}_p
\] (63)

3.3 Spray Source Terms

In Equation (53), the liquid-phase interaction with the Navier-Stokes equations is modeled as a source term which accounts for mass, momentum and energy exchanges between the two phases. In order to compute this source term, the liquid-phase governing equations described in Section 2.3 have to be integrated in time along each particle trajectory from a specified initial condition. As a liquid droplet begins its journey from the injection nozzle, its position, velocity, temperature, and size will change according to its interaction with the gas-phase flowfield. In contrast to the Eulerian approach for the gas-phase equations, the Lagrangian treatment for the liquid-phase equations requires interpolation of the flow quantities from the Eulerian grid to the particle positions and re-distribution of the spray source terms from the computed particle positions to the Eulerian grid for the gas-phase calculation. The overall computational time step is carefully chosen since the time step required for the liquid phase calculations is much smaller than that of the flow solver and because the present LU scheme allows the use of a large CFL number. A large number of particle groups are required to statistically represent the spray behavior. Therefore, it is very important to keep the numerical efficiency of the present implicit scheme from being severely degraded by the coupling (or interaction) between the gas and liquid phases. A strategy used in ALLSPD-3D code to overcome this problem, in steady state flows, is described in the following subsections.

3.3.1 Stochastic Process

In the present code, the stochastic separated flow (SSF) model is applied to account for the dispersion effects on spray characteristics. In this model, the gas-phase velocity fluctuations, \( u_g \), \( v_g \), and \( w_g \), are generated by randomly sampling
a Gaussian probability density distribution having a standard deviation of \( \sqrt{2k/3} \). The instantaneous velocities are then used to evaluate Equation (11). A fixed number of samplings is conducted for each group of particles. The final spray quantities are obtained by averaging the results (source terms, trajectories,..., etc.) among the total samples. Details of the SSF model can be found in Faeth\textsuperscript{14} and Shuen.\textsuperscript{21}

3.3.2 Determination of Spray Time Step

For spray combustion calculations, several time scales exist in the flow field which can differ by several orders of magnitude. The chemical reaction time scale is usually very small compared to the rate of evolution of the gas flow. This is also true for spray evaporation. To accurately calculate particle trajectories, size and temperature, the integration time step has to be small. This restriction is especially severe as the droplet becomes smaller and smaller toward the end of its lifetime. For an unsteady problem, the time step for the entire system will be controlled by the smallest time scale. For a steady state calculation, however, the time step for chemical reaction is usually not a problem in the present formulation due to the implicit treatment of the chemical source term. Experiences indicates that for single-phase combustion calculations, the same CFL number usually can be used for both non-reacting and reacting calculations with the present LU method. This ensures that the convergence properties for reacting flow calculations do not deteriorate using the present numerical algorithm. However, the time step for spray equations still remains small and has to be selected (computed) very carefully in order to obtain accurate spray results and to maintain stability. Here we describe the determination of the spray time step. The spray interactions with the gas flow are described in the next section.

The spray time step at any instant along its trajectory is determined from the following time step constraints: (1) droplet velocity relaxation time \( t_v \), (2) droplet life time \( t_l \), (3) droplet surface temperature constraint time \( t_s \), (4) local grid time scale \( t_g \) and (5) turbulent eddy-droplet interaction time \( t_p \). The final
spray time step ($\Delta t_{spr}$) is determined by taking the minimum of $t_r$, $t_f$, $t_s$, $t_g$ and $t_i$ to ensure the accuracy and stability of the spray calculations. A factor between 0.1 and 0.5 is used to multiply the selected time step in the current spray calculation. These time steps are described in more detail below.

Droplet Velocity Relaxation time ($t_r$)

The locally linearized droplet equations of motion, Equation (11), have an exact solution in terms of the local slip velocity with an exponential decay form. The time constant for the exact solution can be expressed as,

$$t_r = \frac{16}{3} \left( \frac{\rho_i}{\rho_g} \right) \left( \frac{r_p}{v} \right) \left( C_D R_e p \right)^{-1}$$  \hspace{1cm} (64)

Droplet Life Time ($t_l$)

To ensure that the drop size remains positive for the purpose of practical computations, the droplet lifetime at any instant of time is estimated by the following equation,

$$t_l = \frac{\rho_p r_p}{3 \dot{m}_p^*}$$  \hspace{1cm} (65)

Droplet Surface Temperature Constraint Time ($t_c$)

When the governing equation for the droplet internal temperature distribution, Equation (19), is solved, the temperature solution can become completely incorrect due to the use of an inappropriately large time step. This is particularly important for fuel with a low boiling temperature (close to room temperature). In order to ensure the success of the temperature calculation using Equation (19), the exact solution of the infinite conductivity model (also called the uniform temperature model) is used to estimate the time step for the present vortex model. The temperature equation for the droplet based on this model is
\[ \frac{dT_p}{dt} = \frac{6}{\rho_l C_v d_p} \left[ h(T - T_p) - \dot{m}_p h_{fs} \right] \]  

Equation (66) has an exact solution (after local linearization) of the form,

\[ \Delta T_p = A (1 - e^{-A^t}) \]  

where \( \Delta T_p \) is the droplet temperature change within the integration time step, and \( A, A' \) and \( B' \) are defined as,

\[ A = \left( T - T_p - \frac{B'}{A'} \right) \]

\[ A' = \frac{6}{\rho_l C_v d_p} \frac{h}{h_{fs}} \]

\[ B' = \frac{6}{\rho_l C_v d_p} \dot{m}_p h_{fs} \]

Therefore, a time scale can be obtained based on the above solution if a desired \( \Delta T_p \) is specified. This time scale is expressed as,

\[ t_s = \frac{\ln \left( \frac{1}{1 - \frac{\Delta T_p}{A'}} \right)}{A'} \]  

In ALLSPD-3D spray calculations, \( \Delta T_p = 3 \) K is specified for particles just leaving the injector, where they experience a sudden temperature jump, and \( \Delta T_p = 3.0 K \) for the rest of the spray calculation (toward the end of their lifetime).

**Local Grid Time Scale \( (t_s) \)**

A particle can travel across several grid cells and may experience sudden changes in local gas properties if the time step is too large. This not only causes
inaccuracy in the integration, but also increases the difficulty of locating the particle positions. Therefore, a time scale, \( t_y \), is computed to ensure that the distance a particle moves is less than one local cell size in one time step.

**Turbulent Eddy-Droplet Interaction Time \( t_i \)**

According to Shuen et al.,\(^\text{22}\) a particle is assumed to interact with an eddy for a time which is the minimum of either the eddy lifetime or the transit time required for the particle to cross the eddy. These times are estimated by assuming that the characteristic size of an eddy is the dissipation length scale given by,

\[
Le = \frac{C_\mu^{3/4} k^{3/4}}{\varepsilon}
\]  

(70)

and the eddy lifetime is estimated as,

\[
t_e = \frac{Le}{\bar{a}_3}
\]  

(71)

The transit time of a particle was found using the linearized equation of motion for a particle in a uniform flow,

\[
t_i = -\tau \ln \left[ 1 - \frac{Le}{\tau |\bar{a}_r - \bar{a}_p|} \right] \text{ where } \tau = \frac{8 \rho_p r_p}{3 \rho_s C_D |\bar{a}_r - \bar{a}_p|^2}
\]  

(72)

and \( |\bar{a}_r - \bar{a}_p| \) is the relative velocity at the start of the interaction. When \( Le > \tau |\bar{a}_r - \bar{a}_p| \), the linearized stopping distance of the particle is smaller than the characteristic length scale of the eddy, and then Equation(72) has no solution. In this case, the eddy has captured the particle and the interaction time is the eddy lifetime. Therefore,

\[
t_i = t_e \quad \text{if} \quad Le > \tau |\bar{a}_r - \bar{a}_p|
\]

\[
t_i = \min (t_e, t_i) \quad \text{if} \quad Le \leq \tau |\bar{a}_r - \bar{a}_p|
\]
3.3.3 Interaction Between Two Phases

For the gas-phase equations, Equation (53), the presence of the spray appears in the form of a source term, $H^{(m)}$. This source term represents the interchange of the mass, momentum and energy between the two phases. As particles are injected into the flow domain, their subsequent behavior (positions, velocities, size and temperature) is affected continuously by the neighboring gas properties and *vice versa*. Usually a large number of spray particles is desirable to accurately predict the spray behavior. However, this requires tremendous computational effort for the spray calculation alone. To minimize the computational time for the spray, the spray source term, $H^{(m)}$, need not be updated at every gas-phase iteration (pseudo-time marching). Usually the spray source term is updated every 10 to 20 iterations in ALLSPD-3D spray calculations. When the spray source term is updated, each group of particles is integrated either to the end of its lifetime or until it leaves the computational domain. It should be noted that, for the present steady state spray combustion calculation, the spray time step determined previously is independent of the pseudo-time used for the gas-phase equations, which is determined from the maximum CFL number obtainable according to the local eigenvalues. Therefore, this strategy maintains both the efficiency of the flow solver and the accuracy of the spray Lagrangian integration. This treatment of the gas-liquid interaction is different from those of Raju and Sirignano\textsuperscript{23} where time-accurate solutions are of their primary concern. In the present computation, it is assumed that when the droplets impinge on the walls they evaporate completely.

3.4 Pseudo-Time Step

In Equation (58), $\Delta \tau$ is the pseudo-time which is determined from the eigenvalue analysis. For the flow set of equations, the eigenvalues are (defined previously),
\[
\lambda_A = \frac{1}{2} \left( \frac{U_A}{c^2} \right)^2 \pm \frac{U_A^2}{c^2} \left( 1 - \frac{\hat{\beta}}{c^2} \right)^2 + 4 \hat{\beta} \left( \xi_x^2 + \xi_y^2 + \xi_z^2 \right)
\]
\[
\lambda_B = \frac{1}{2} \left( \frac{U_B}{c^2} \right)^2 \pm \frac{U_B^2}{c^2} \left( 1 - \frac{\hat{\beta}}{c^2} \right)^2 + 4 \hat{\beta} \left( \eta_x^2 + \eta_y^2 + \eta_z^2 \right)
\]
\[
\lambda_C = \frac{1}{2} \left( \frac{U_C}{c^2} \right)^2 \pm \frac{U_C^2}{c^2} \left( 1 - \frac{\hat{\beta}}{c^2} \right)^2 + 4 \hat{\beta} \left( \zeta_x^2 + \zeta_y^2 + \zeta_z^2 \right)
\]

(73)

where,

\[
U_A = \xi_x u + \xi_y v + \xi_z w,
\]
\[
U_B = \eta_x u + \eta_y v + \eta_z w,
\]
\[
U_C = \zeta_x u + \zeta_y v + \zeta_z w,
\]

and,

\[
\hat{\beta} = \max(\beta, U_{ref})
\]

The pseudo-time, \( \Delta \tau^* \) is then calculated by the following equation,

\[
\Delta \tau^* = CFL / (\lambda_A^2 + \lambda_B^2 + \lambda_C^2)^{1/2}
\]

(74)

where \( CFL \) is a user-input CFL number and \( U_{ref} \) is a reference velocity.

### 3.5 Boundary Conditions

Boundary conditions in ALLSPD-3D are applied implicitly. Certain boundary conditions make use of ghost cells which are invisible to the user. In particular, grid interface boundaries acquire information for their ghost cells from the neighboring blocks. The grid connection between blocks is assumed to be contiguous and the neighboring volumes should be approximately the same size.
3.5.1 Wall Boundary Condition

3.5.1.1 Mean Equations

The boundary condition applied to the mean equations assumes that the pressure gradient is zero at the walls. Since the grid may be non-orthogonal, the projected normal pressure gradient is found from the grid metrics. For example, on an i-face

\[
\frac{\partial p}{\partial n} = \frac{\partial z}{\partial nd \xi} + \frac{\partial \eta}{\partial nd \eta} + \frac{\partial \zeta}{\partial nd \zeta}
\]

A no-slip wall condition with either an adiabatic or an isothermal condition may be chosen. The adiabatic condition is imposed by setting the enthalpy gradient to zero; similar to the pressure condition described above. To specify a wall temperature, the static enthalpy is determined for the desired wall temperature. The species mass fractions used in this calculation are the lagged values resulting from the previous species iteration.

3.5.1.2.1 Dilution Holes

A special boundary condition has been defined for problems containing injection ports. These areas are assumed to be relatively small, and to have a known mass flow rate and static temperature. The pressure is found from a first order extrapolation from the interior domain. The first step in applying this boundary condition is to determine the actual injection area. This is done by taking the average of the magnitude of the area normal to the four corners for every grid cell within the prescribed bounds.

Next, the change in velocity is found by calculating the change in mass flow rate required to meet the imposed conditions assuming that the density is unchanged from the previous iteration.
\[ \Delta V = \frac{m_{inj}}{\rho A_{inj}} - (u)^p. \]

The velocity direction comes from the direction cosines prescribed by the user. The boundary condition for the energy equation is set by holding the injected fluid to a constant stagnation enthalpy. This is accomplished by calculating the static enthalpy for the prescribed injected temperature using the thermal properties from the previous iteration,

\[ \Delta h = h(T_{inj}) + \frac{1}{2} \left( \frac{m_{inj}}{\rho A_{inj}} \right)^2 - h_i^p \]

This differs from the upstream boundary condition in that the mass flow rate is prescribed rather than the velocity.

3.5.1.3.2 Transpiration Cooling (or Bleed)

Walls containing transpiration cooling or bleed have been handled in a manner similar to the injection boundary condition. However, for this boundary condition, only the mass flow is prescribed as described above. The conditions on pressure and temperature are the same as for a wall. In other words, the energy equation can be bounded by either an adiabatic wall or by an isothermal wall option.

3.5.1.4 Turbulence Model Equations

At the wall, the turbulent kinetic energy and turbulent dissipation are bounded by non-zero values determined by Shih, et.al. to maintain realizability,

\[ k_{bc} = 0.250 U_t^2 \quad \varepsilon_{bc} = 0.251 \frac{U^4}{\nu} \]  \hspace{1cm} (75)

The above constraint on \( \varepsilon \) is accurate for very near wall flows but may be difficult to satisfy without grid clustering near the walls. For this reason, an optional
boundary condition has been included to relax the grid resolution requirement near walls. For this condition, the gradient of turbulent dissipation given by Equation (75) is replaced with $\partial \varepsilon / \partial \eta = 0$.

### 3.5.1.5 Dilution Holes

The values for the injected turbulence field are found from the user specified turbulent intensity

$$
\Delta k = \left[ \frac{\frac{3}{2} u_{inj}^3}{\rho A_{inj}} - k^p \right]
$$

where $u_{inj}$ is the injected turbulent intensity, and $A_{inj}$ is the total injection area. The assumed injected dissipation is then calculated from the turbulent kinetic energy and a specified length scale. For a dilution hole, the assumed length scale is the square root of the injection area.

$$
\Delta \varepsilon = \left[ \left( \frac{\frac{3}{2} u_{inj}^3}{\rho A_{inj}} \right)^{\frac{3}{2}} \right] - \varepsilon^p
$$

### 3.5.1.6 Transpiration Cooling (or Bleed)

For this type of boundary condition, the turbulence quantities are taken from the closest interior grid point.

### 3.5.2 Species Continuity Equations

At a wall the gradient of the species mass fractions is set to zero.
3.5.3 Dilution Holes or Transpiration Cooling (or Bleed)

For these boundary conditions the composition of the injected fluid is prescribed by the user and remains unchanged.

3.5.4 Upstream

At an upstream boundary, the stagnation enthalpy is held constant. This is accomplished by retaining the initial values for the three velocity components, species mass fractions, and by holding the inlet temperature constant. Pressure is extrapolated from the interior points. The default extrapolation method is a second order difference. The turbulence quantities are held at their initial definitions as are the species mass fractions.

The default upstream conditions for the turbulent flow scalar quantities are generated using following expressions. The velocity scales and turbulent kinetic energy are computed based on local or reference velocity levels and turbulence intensity levels.

\[ k = \frac{3}{2} \left( T_u U_{ref} \right)^2 \] (76)

The specific dissipation is modeled using the following three formulations for attached flow situations. The first expression uses that the most fundamental form of the specific dissipation where it is expressed as a function of velocity and length scales. Here, the length scale is assumed to be the Karman's mixing length scale formulation.

\[ \varepsilon = 2.4 \frac{k^{3/2}}{L_{ref}} \quad L_{ref} = 0.4 y_n \] (77)

The second formulation uses the experimental correlation of Mellor.\textsuperscript{33} Here, the experimental data is expressed as a function of eddy viscosity profiles. Therefore, the following expression is used to find the specific dissipation,
\[ \varepsilon = \frac{k^2}{\frac{\mu_t}{\mu_l} \frac{\mu_l}{\mu_l} \rho} \] 
\[ \frac{\mu_t}{\mu_l} = \frac{(0.4y^+)^4}{\left(0.4y^+\right)^4 + 328} \] 

(78)

The third method is similar to Equation (77). Here, the length scale definition has been modified to yield the following.

\[ \varepsilon = \frac{k^2}{500 \gamma^2 t_{ref}} \] 

(79)

3.5.5 Downstream

The flow through the downstream boundary is assumed to remain subsonic. An exit pressure is imposed at the grid cell past the exit plane. For multiple block problems, different exit pressures may be set for each block, however care must be taken if the exit pressures vary greatly from the initial pressure (which is set to user inputted value of Pref over the exit face). If an exit pressure does require a significant departure from the initial value, a large pressure gradient at the exit can be avoided by slowly decreasing the exit pressure over a number of iteration.

The remaining flow quantities are extrapolated from the interior flow domain. A zero order extrapolation method is the default. An expert user flag is included in subroutine qplus1.f to change the order of this extrapolation if needed.

3.5.6 Periodic

For the periodic boundary condition, flow quantities are copied from the opposite face into the ghost cells, e.g., the information for the j=1 plane would come from the j=jmax plane. This boundary condition requires only two grid points in the periodic direction and is therefore the best choice for a pseudo-two-
dimensional problem.

3.5.7 Symmetry

The symmetry boundary condition copies the flow field from the neighboring interior point to the "ghost cell", e.g., the information for the j=0 plane would come from the j=2 plane. Since ALLSPD-3D requires a minimum of five grid points (including the ghost cells), this option requires the user to supply a minimum of three grid points.

3.5.8 Far Field

The same condition is used as for the downstream boundary condition.

3.5.9 Singularity

For axisymmetric grids, where a plane collapses to a single line, a special boundary condition has been implemented. This boundary must be applied for the k=1 boundary. The solution domain will not include the points along the singularity, so that the centerline values must be found by extrapolation from the interior points. A check is made for internal blockages at the centerline so that the appropriate wall boundary conditions (see above) can be applied to the turbulence equations.
IV : Input/Output Files

4.1 Input/Output

The input and output file structure is explored in this section. Four user or GUI developed input files are required to execute the ALLSPD-3D code. To run the ALLSPD-3D code, users need the grid file, grid.dat, block link file, gridlink.dat, property data file, prop.dat, and the flow data file, allspd.dat. A typical grid.dat read format is shown in Figure 3. For cases where users would like to provide the initial conditions for the flow variables within the whole domain, the user-defined initial condition file, uvwt.dat, has to be generated. The structure of these input files will be discussed along with several examples. The gridlink.dat file contains the grid size, connectivity and boundary condition information. A typical gridlink file is shown in Figure 4. A detailed example is presented in Table 3. A typical allspd.dat file is shown in Figure 5. Various options are summarized in Table 3. The ALLSPD-3D property input file, prop.dat, is arranged in three major groups: Thermal Property Information, Finite Rate Chemistry Information and Spray Property Information. They are arranged in top-down groups as shown in Figure 6. The requirements of each group's input information is summarized later. For liquid spray problems, users can provide their own spray information data file, spray.dat, instead of using the default spray particle distribution generator.

For the output files, the most important files are the PLOT3D format solution files which can be examined using the NASA PLOT3D plotting package or other post-processors which accept this file format. If a liquid spray is initiated, the particle trajectories, temperatures, sizes are also written to a single file.
for post-processing. A restart file for subsequent runs may also be generated and is usually needed for running large combustion problems.

4.2 Input Data Files

4.2.1 Grid Data: grid.dat

The grid is generated separately by the user and read into the code by the subroutine MESH. The default grid data file name is grid.dat. The format for the grid file is consistent with PLOT3D’s multi-block format. It is generated as a free formatted file and therefore is portable. There are two formats available for grid.dat. One is a non-blanked file, and the other is an Iblanked. The Iblanked file requires an extra set of Iblanked data (0 or 1 for each grid point) in addition to the usual x, y, and z data. For most users, a non-Iblanked format is preferred. The input flag for either Iblanked or non-Iblanked grid.dat file is in gridlink.dat that will be described later. The format for grid.dat is listed as follows.

For non-Iblanked format:

```fortran
read(iunit,*)nblk
read(iunit,*)(imax(n),jmax(n),kmax(n),n=1,nblk)
do n=1,nblk
   read(iunit,*)
   (((x(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
   (((y(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
   (((z(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n))
end do
```

For Iblanked format:

```fortran
read(iunit,*)nblk
read(iunit,*)(imax(n),jmax(n),kmax(n),n=1,nblk)
do n=1,nblk
   read(iunit,*)
   (((x(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
   (((y(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
   (((z(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
   (((iblk(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n))
end do
```

FIGURE 3. Read file format for grid.dat
4.2.2 Multi-Block Data: gridlink.dat

The multi-block connection information is defined in gridlink.dat and is read by the subroutine MESH. The input variables are described in Table 3.

<table>
<thead>
<tr>
<th>iblank</th>
<th>Number of block information</th>
<th>Number of cells per block</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>3</td>
<td>49</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>49</td>
</tr>
<tr>
<td>4</td>
<td>1 3 3 5 5 0.0</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>4</td>
<td>3 3 3 5 5 0.0</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>2</td>
<td>4 3 3 5 101325.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 0 0</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>3</td>
<td>1 0 0 0 0</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
<tr>
<td>2</td>
<td>0 0 0 0 0</td>
<td>0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Boundary Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>20 24 1 1 1 5 .001058e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>20 24 1 1 20 24 .002116e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>20 24 1 1 39 43 .001058e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>20 24 49 49 1 5 .001058e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>20 24 49 49 20 24 .002116e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>20 24 49 49 39 43 .001058e-1 325.0 .0010 .258 .965 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>15 16 1 1 1 43 .00345e-2 325.0 .0010 1.00 .0 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>15 16 49 49 1 43 .00345e-2 325.0 .0010 1.00 .0 0.0 0.0 .233 0.0 0.0 .76</td>
</tr>
<tr>
<td>10 0</td>
</tr>
<tr>
<td>4 12 1 1 8 16 .00826e-1 325.0 .001 .139 .990 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>4 12 1 1 28 36 .00826e-1 325.0 .001 .139 .990 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>4 12 49 49 8 16 .00826e-1 325.0 .001 .139 .990 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>4 12 49 49 28 36 .00826e-1 325.0 .001 .139 .990 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 1 1 1 5 .002077e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 1 1 20 24 .003939e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 1 1 39 43 .002077e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 49 49 1 5 .002077e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 49 49 20 24 .003939e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
<tr>
<td>23 27 49 49 39 43 .002077e-1 325.0 .001 .087 .996 0.0 0.0 .233 0.0 0.0 .767</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dilution Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

FIGURE 4. Typical gridlink.dat file
<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>bblank</td>
<td>The flag for non-blanked or blanked grid.dat</td>
</tr>
<tr>
<td></td>
<td>0 = non-blanked grid.dat</td>
</tr>
<tr>
<td></td>
<td>1 = blanked grid.dat</td>
</tr>
<tr>
<td>(imaxbl(nb),jmaxbl(nb), kmaxbl(nb),nb=1,nblk)</td>
<td>Number of grid points in i, j, and k direction, respectively, in the nbth block.</td>
</tr>
<tr>
<td>nbk</td>
<td>Total number of blocks</td>
</tr>
<tr>
<td>least(nb),lwest(nb), lnorth(nb),lsouth(nb), ltop(nb),lbot(nb),bcin(nb,1)</td>
<td>Block boundary control indices, specifying the boundary type in the east, west, north, south, top, and bottom faces of the nbth block.</td>
</tr>
<tr>
<td></td>
<td>The boundary types:</td>
</tr>
<tr>
<td></td>
<td>least(nb),lwest(nb),lnorth(nb),lsouth(nb), ltop(nb),lbot(nb),</td>
</tr>
<tr>
<td></td>
<td>= 1 upstream (inflow) (all six faces), = 2 downstream (outflow) (all six faces)</td>
</tr>
<tr>
<td></td>
<td>= 3 solid wall (all six faces), = 4 block interface (all six faces)</td>
</tr>
<tr>
<td></td>
<td>= 5 symmetry (only for north, south, top, and bottom faces)</td>
</tr>
<tr>
<td></td>
<td>= 6 periodic boundary (only for north/south, and top/bottom faces)</td>
</tr>
<tr>
<td></td>
<td>= 7 singularity (only for bottom face, i.e., k=1)</td>
</tr>
<tr>
<td></td>
<td>= 8 far-field (for external flows) (all six faces)</td>
</tr>
<tr>
<td></td>
<td>bcin(nb,1),bcin(nb,2),bcin(nb,3),bcin(nb,4), bcin(nb,5),bcin(nb,6)</td>
</tr>
<tr>
<td></td>
<td>are used to assign either pressure or temperature at each faces depending on the boundary types. Zero values are usually used except for those faces that are either wall or exit conditions. For wall boundary faces, zero (0.0) is used to assign an adiabatic wall and desired temperature (°K) is used to assign an isothermal wall. For exit faces, these are used to specify the ambient pressure (N/m²) that is to remain constant. The corresponding faces are:</td>
</tr>
<tr>
<td></td>
<td>bcin(nb,1) east face, bcin(nb,2) west face, bcin(nb,3) north face, bcin(nb,4) south face, bcin(nb,5) top face, bcin(nb,6) bottom face</td>
</tr>
<tr>
<td>lee(nb),lw(wb),lnn(nb), lss(nb),ltt(nb),lbb(nb) , nb=1,nblk</td>
<td>Block boundary neighboring control indices, indicating the identity of the neighboring block, such as:</td>
</tr>
<tr>
<td></td>
<td>lee(nb)=m - nbth block's east side borders nth block; zero if nth block's east side is not a block interface.</td>
</tr>
<tr>
<td></td>
<td>lnn(nb)=m - nbth block's north side borders nth block; zero if nth block's north side is not a block interface, etc...</td>
</tr>
<tr>
<td>intblk</td>
<td>Internal blockage flag</td>
</tr>
<tr>
<td></td>
<td>= 0 no internal blockage defined; skip the following input data and move directly to the next one for dilution holes (ndi).</td>
</tr>
<tr>
<td></td>
<td>=1 with internal blockage(s); provide the following input data for internal blockage(s) in each block.</td>
</tr>
<tr>
<td>Input Variable</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| nintblk(nb),intb(nn,nb), jinte(nn,nb),jintb(nn,nb), kinte(nn,nb),nn=1, nintblk(nb) | nintblk(nb): number of internal blockages in the nth block (≤ 50) (limit defined in parameter file, allspd.h). If zero, skip to the next block; if equal or greater than one (up to nintblk(nb)) enter following 
| intb(nn,nb), jinte(nn,nb): the beginning and ending i-index for the nth internal blockage in the nth block. jintb(nn,nb), jinte(nn,nb): the beginning and ending j-index for the nth internal blockage in the nth block. kinte(nn,nb): the beginning and ending k-index for the nth internal blockage in the nth block. |
| ndil(nb),isdlil(nb,nn), jedil(nb,nn),kedil(nb,nn), dilmass(nb,nn), diltemp(nb,nn), dilin(nb,nn), dilcosx(nb,nn), dilcosy(nb,nn), dilcosz(nb,nn), (dilys(nb,nn,nn), nx=1,nn=1,ndil(nb)) | Dilution holes specification. 
| ndil(nb): number of dilution holes in the nth block. If zero, skip to the next block.. 
| jedil(nb,nn), kedil(nb,nn): the beginning and ending i-index for the nth dilution hole in the nth block. 
| dilmass(nb,nn): the mass flow rate (kg/s) for the nth dilution hole in the nth block. 
| dilin(nb,nn): the temperature (K) for the nth dilution hole in the nth block. 
| dilcosx(nb,nn), dilcosy(nb,nn), dilcosz(nb,nn): the x, y, z-directional cosine for the angle of the cooling velocity for the nth dilution hole in the nth block. 
| dilys(nb,nn,nn), nx=1,nn=1: the mass fractions of each species for the nth dilution hole in the nth block. |
| sstrn(nb),ststrn(nb,nn), kstrn(nb,nn), jstrn(nb,nn), kstrn(nb,nn), trmmass(nb,nn), trncossx(nb,nn), trncosy(nb,nn), trncosz(nb,nn), (trys(nb,nn,nn),nx=1,nn=1,sstrn(nb)) | Transpiration cooling holes specification. 
| sstrn(nb): number of transpiration cooling holes in the nth block. If zero, skip to the next block.. 
| jstrn(nb,nn), kstrn(nb,nn): the beginning and ending i-index for the nth transpiration cooling hole in the nth block. 
| jstrn(nb,nn), kstrn(nb,nn): the beginning and ending j-index for the nth transpiration cooling hole in the nth block. 
| kstrn(nb,nn), kstrn(nb,nn): the beginning and ending k-index for the nth transpiration cooling hole in the nth block. 
| trmmass(nb,nn): the mass flow rate (kg/s) for the nth transpiration cooling hole in the nth block. 
| trncossx(nb,nn), trncosy(nb,nn), trncosz(nb,nn): the x, y, z-directional cosine for the angle of the cooling velocity for the nth transpiration cooling hole in the nth block. 
| trys(nb,nn,nn), nx=1: the mass fractions of each species for the nth transpiration cooling hole in the nth block. |
4.2.3 Flow Data: allspd.dat

The flow data and major control of the program execution is done in the allspd.dat and is read by the subroutine INPUT. Each input is described in Table 4.

\[ \begin{align*}
0 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \\
10 \

FIGURE 5. Typical Listing of the allspd.dat Input File
<table>
<thead>
<tr>
<th>Input Data</th>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 0         | lread             | = 0 a fresh run  
                         | = 1 a restart run                                                                                                                    |
| 10        | nmax              | Total number of iterations (pseudo-time steps)                                                                                          |
| 10        | npot              | Every npot number of iterations the plot3d files are generated, and these files are overwritten every next npot number of iterations.  |
| 10        | nsave             | Every nsave number of iterations the restart file is generated, and this file is overwritten every next nsave number of iterations.    |
| 1, 0, 0   | iturb, icmu,      | turbulence model switch,                                                                                                                |
|           | idisbc            | = 0 laminar flow  
                         | = 1 turbulent flow,  
                         | eddy viscosity coefficient  
                         | = 0 constant Cμ in turbulence model (=0.09)  
                         | = 1 variable Cμ in turbulence model  
                         | idisbc  
                         | = 0 Dirichlet wall boundary condition for ε equation.  
                         | = 1 Neumann wall boundary condition for ε equation. |
| 1         | icomb             | chemistry switch  
                         | = 0 non-reacting flow  
                         | = 1 reacting flow                                                                                                                     |
| 1         | iaartv            | Explicit artificial viscosity switch  
                         | = 0 no artificial damping  
                         | = 1 with artificial damping                                                                                                             |
| 8.000e-01, 3.000e-01 | vis4, vis2      | vis4 = 4th order damping coefficient  
                         | vis2 = 2nd order damping coefficient                                                                                                   |
| 1e6, 1e6, 1e6 | CFL               | cfl(1) = CFL number for the flow equations  
                         | cfl(2) = CFL number for the k-ε equations  
                         | cfl(3) = CFL number for the species equations                                                                                           |
| 1.05, 1.05, 1.05 | prefcac         | prefcac(1) = pre-conditioning factor for LU scheme for the flow equations  
                         | prefcac(2) = pre-conditioning factor for LU scheme for the k-ε equations  
                         | prefcac(3) = pre-conditioning factor for LU scheme for the species equation                                                          |
TABLE 4. Detailed Layout of allspd.dat File

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>nswitch</td>
<td>For some difficult flows, a smaller CFL number initially is helpful to start the calculation. The CFL number can then be ramped up as the flow develops. The CFL number is calculated using the following formula: &lt;br&gt;CFL=CFLxfactau &lt;br&gt;where the factor factau has initial value of facini (defined next) and is linearly increased to 1 over nswitch iterations.</td>
</tr>
<tr>
<td>0.1</td>
<td>facini</td>
<td>The initial value used in conjunction with the above nswitch. facini is usually set to 0.1.</td>
</tr>
<tr>
<td>1e-7</td>
<td>bond</td>
<td>The convergence criterion, say, 1.0e-7 for example.</td>
</tr>
<tr>
<td>1</td>
<td>nkecall</td>
<td>The turbulence equations are updated (called) every nkecall flow iterations. Say, if nkecall=10, the turbulence equations are updated every 10 flow iterations.</td>
</tr>
<tr>
<td>1</td>
<td>nscal0</td>
<td>The species equations are updated (called) every nscal0 flow iterations. Say, if nscal0=10, the turbulence equations are updated every 10 flow iterations.</td>
</tr>
<tr>
<td>1</td>
<td>uuvwrt</td>
<td>Initial condition controlling flag &lt;br&gt;-1 only inlet faces have non-zero flow specified by uin,vin, and win described next. &lt;br&gt;-0 uniform flow initialization for the entire domain except for the walls. The flow is defined by uin,vin, and win described next. &lt;br&gt;-1 user provides the initial flow condition for the entire domain. In this case the flow data defined here (uin,vin, win) will be used only as reference. The description for the data file, uuvwrt.dat, is in another section. &lt;br&gt;-2 this is used only for easy specification of the swirling inlet condition. The file name is the same as in uuvwrt=1 case that is uuvwrt.dat. The format of this file for this case will be described later.</td>
</tr>
<tr>
<td>62.5 0.0 0.0</td>
<td>uin, vin, win</td>
<td>Initial velocities. These are used to assign initial flow velocity for the entire domain or inlet only, depending on the value of uuvwrt above. They are used to calculate the inlet flow Reynolds number also. &lt;br&gt;uin = x component of velocity (m/s) &lt;br&gt;vin = y component of velocity (m/s) &lt;br&gt;win = z component of velocity (m/s)</td>
</tr>
<tr>
<td>307.0</td>
<td>Tref</td>
<td>Reference temperature (K) for the flow domain. This is used to assign initial flow temperature for the entire domain. It is used to calculate the inlet flow Reynolds number also. However, the initial temperature will be overwritten by the temperature defined in uuvwrt.dat if uuvwrt=1, or 2.</td>
</tr>
<tr>
<td>1.013250e+05</td>
<td>Pref</td>
<td>Reference pressure (N/m²) for the flow domain. This is used to assign initial flow pressure for the entire domain except along outlet faces. It is used to calculate the inlet flow Reynolds number also. However, the initial pressure will be overwritten by the pressure defined in uuvwrt.dat if uuvwrt=1, or 2.</td>
</tr>
<tr>
<td>Input Data</td>
<td>Input Variable</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>.00254</td>
<td>lref</td>
<td>A reference length scale (m) used to calculate the inlet Reynolds number and turbulence quantities.</td>
</tr>
</tbody>
</table>
| -1        | iebm           | Eddy break-up model for turbulence-combustion closure.  
= 0 no eddy break-up model  
= -1 with eddy break-up model |
| 0         | iequate        | A flag to compute the reverse/forward reaction constant in reacting flow calculation.  
= 0 skip the reverse/forward reaction constant calculation  
= 1 perform the reverse reaction constant calculation  
= -1 perform the forward reaction constant calculation |
| 150       | st1igt         | Ignition energy is added to the gas phase energy equation starting at the time step of st1igt. |
| 50000     | st2igt         | Ignition energy is turned off from the gas phase energy equation starting after the time step of st2igt. |
| 1         | noigt          | Number of ignition sources. (if = 0, skips to next input of toff) |
| 1, 5 31 343343 | ibkigt, iligt, jligt, ... kligt | ibkigt(no), iligt(no), jligt(no), jligt(no), jligt(no), jligt(no), kligt(no), kligt(no), no=1, noigt: Specification of the position of the ignition spark(s).  
ibkigt(no): The start index of the ignition location.  
iligt(no): The starting i-index of the ignition spark location.  
jligt(no): The starting j-index of the ignition spark location.  
kligt(no): The starting k-index of the ignition spark location. |
| 1.6000000e+03 | toff         | The cut-off temperature within the ignition region to turn off the energy sparks. |
| 5.0e+07   | sparkuser      | The strength of ignition spark (J/m^2-s). If zero, then the default strength in the code is used. |
| 0         | nsdata         | = 0 The spray data are to be generated in the following inputs.  
= 1 The spray data are to be provided through user-generated file, spray.dat. The format for spray.dat is described in another section. |
| 1         | ispray         | = 0 Single-phase flow calculation. (no liquid phase equations are solved)  
= 1 Two-phase flow calculation. If ispray=0 or nsdata=1, skip to the input for implot(i). |
| 1         | izomp          | Nozzle zone |
| 3 3 22 28 19 25 | ispray1, ispray2, jspray1, jspray2, kspray1, kspray2 | The spray particles will be issuing from the (i,j,k) points between ispray1, ispray2 points in i direction, between jspray1, jspray2 points in j direction, and between kspray1, kspray2 points in k direction. |
### TABLE 4. Detailed Layout of allspd.dat File

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 45.0 65.0</td>
<td>nangle, angle1, angle2</td>
<td>nangle = Number of angles between (angle1,angle2) from which the spray particles are issuing. angle1 = The lower angle (say 10 degree) for the spray particles to shoot out. (The angle currently is only in the x-z plane). angle2 = The upper angle (say 45 degree) for the spray particles to shoot out. (The angle currently is only in the x-z plane).</td>
</tr>
<tr>
<td>5.5 7.5</td>
<td>size1, size2</td>
<td>size1 = The smallest size of spray particles (in terms of μm in radius). size2 = The largest size of spray particles (in terms of μm in radius).</td>
</tr>
<tr>
<td>100</td>
<td>ntspr</td>
<td>The time step to initiate the spray subroutine.</td>
</tr>
<tr>
<td>20</td>
<td>ncalpr</td>
<td>The spray subroutine is called every ncalpr flow iterations (pseudo time steps)</td>
</tr>
<tr>
<td>0</td>
<td>inspr</td>
<td>Always set to 0 (reserved for later version use)</td>
</tr>
<tr>
<td>300.0</td>
<td>tspn</td>
<td>Liquid spray temperature(K)</td>
</tr>
<tr>
<td>4.67e-4</td>
<td>flowspry</td>
<td>Liquid spray mass flow rate (kg/s)</td>
</tr>
<tr>
<td>50.0</td>
<td>vispr</td>
<td>Liquid spray velocity</td>
</tr>
<tr>
<td>1 2 3 4 5 15 16 17 18 19 11 12 13 14 8</td>
<td>inplot</td>
<td>Plot file flags specify the flow variables to be written to the plot3d files. There are three files to be written out for plotting and/or post-processing purposes. The first file contains vector quantities, allspd.PFV, and the second and the third files contain all scalar quantities, allspd.PFS1 and allspd.PFS2. In plot3d, five variables are needed in each file for three-dimensional problems. ALLSPD-3D users are asked to specify 15 flow variables for these three output files. The inplot(i) has the following valid integer values. =1 density, =2 u momentum, =3 v momentum, =4 w momentum =5 static temperature, =6 gage pressure, =7 static pressure =8 turbulent Cp, =9 static enthalpy, =10 total enthalpy =11 turbulent kinetic energy, =12 turbulent dissipation =13 turbulent eddy viscosity/laminar viscosity =14 distance to the walls, =i+14 mass fraction of species i</td>
</tr>
<tr>
<td>5</td>
<td>nspr</td>
<td>Number species to read in Initial Condition</td>
</tr>
</tbody>
</table>
TABLE 4. Detailed Layout of *allspd.dat* File

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;C1H3O7&quot; 0.000000e+0</td>
<td>name, Ys(1)</td>
<td>Species name and species mass fraction</td>
</tr>
<tr>
<td>&quot;O2&quot; 2.330000e-01</td>
<td>name, Ys(2)</td>
<td>Species name and species mass fraction</td>
</tr>
<tr>
<td>&quot;CO2&quot; 0.000000e+00</td>
<td>name, Ys(3)</td>
<td>Species name and species mass fraction</td>
</tr>
<tr>
<td>&quot;H2O&quot; 0.000000e+00</td>
<td>name, Ys(4)</td>
<td>Species name and species mass fraction</td>
</tr>
<tr>
<td>&quot;N2&quot; 7.670000e-01</td>
<td>name, Ys(ns)</td>
<td>Species name and species mass fraction</td>
</tr>
</tbody>
</table>

4.2.4 User Defined Initial Conditions: *uvwt.dat*

When $iuvwt=1$ or $2$ in *allspd.dat* is specified, the code needs the *uvwt.dat* data file. This file is read by subroutine INIT.

For $iuvwt=1$: *uvwt.dat* file must be an unformatted file.

```
  do n=1,nblk
     ii=imaxbl(n)
     jj=jmaxbl(n)
     kk=kmaxbl(n)
     do k=1,kk
         do j=1,jj
             do i=1,ii
                  read(9) pp, u, v, w, t, tke, tep,(ys(nn), nn=1, ns)
             end do
         end do
     end do
  end do
```

where

- \( nblk \) = number of blocks
- \( \text{imaxbl}(n) \) = grid points in \( i \) direction in \( n \)th block
- \( \text{jmaxbl}(n) \) = grid points in \( j \) direction in \( n \)th block
- \( \text{kmaxbl}(n) \) = grid points in \( k \) direction in \( n \)th block
- \( pp \) = gauge pressure (p-pref)
- \( u \) = \( x \) component of velocity
- \( v \) = \( y \) component of velocity
- \( w \) = \( z \) component of velocity
- \( t \) = static temperature
- \( tke \) = turbulent kinetic energy
- \( tep \) = turbulent dissipation rate
- \( \text{ys}(nn), nn=1, ns \) = mass fraction for each species
For iuwt=2: uvwt.dat must be a free formatted file, only the inlet face data are defined in this file (say i=1).

This option is designed for the users who want to specify an inlet swirling profile that can be processed and fitted into the grid points at the inlet face. Right now, a flat inlet face in the west boundary (i=1 and x=constant) is assumed to apply this option. Users have only to provide a radial profile of the swirl cup, and an axisymmetric condition is assumed for the swirling flow at the inlet. If a user’s geometry (and grid) does not fit into the above swirling condition, the user can always use the iuwt=1 option to define the variables for the entire domain including, of course, the inlet swirling profiles.

\[
\begin{align*}
d\ n=1,nblk \\
read(iunit,*) \ irmax \\
read(iunit,*) \ rmax,y0,z0 \\
read(iunit,*) \ (ycup(nx),nx=1,ns) \\
do \ ii=1,irmax \\
read(iunit,*) \ rcup(ii),ucup(ii),vcup(ii),wcup(ii),tcup(ii),tkcup(ii) \\
end do \\
end do \\
\end{align*}
\]

where

\texttt{nblk} = number of blocks
\texttt{irmax} = number of radial points used to define a swirl profile
\texttt{rmax} = radius of the swirler cup (m).
\texttt{y0} = y coordinate in the center of the swirler cup (m)
\texttt{z0} = z coordinate in the center of the swirler cup (m)
\texttt{(ycup(nx),nx=1,ns)} = Mass fractions of the species
\texttt{rcup(ii)} = \( r/rmax \), where \( r \) is the radial distance from the center of the swirler cup
\texttt{ucup(ii)} = u-velocity (m/s)
\texttt{vcup(ii)} = v-velocity (m/s)
\texttt{wcup(ii)} = w-velocity (m/s)
\texttt{tcup(ii)} = temperature (K)
\texttt{tkcup(ii)} = turbulence intensity
4.2.5 Property Data: prop.dat

Property data, chemistry kinetics and liquid fuel property are read by subroutine PROFIN and the data file name is prop.dat.

<table>
<thead>
<tr>
<th>TABLE 5. Detailed Information about prop.dat File</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Variable</strong></td>
</tr>
<tr>
<td>heading:</td>
</tr>
<tr>
<td>nbf:</td>
</tr>
<tr>
<td>heading:</td>
</tr>
</tbody>
</table>
| specnm(nx),wm(nx),nx=1,ns: | specnm(nx) = species name (up to 80 characters).  
wm(nx) = molecular weight for each species.  
heading: | title of the property (up to 80 characters) |
TABLE 5. Detailed Information about prop.dat File

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>specmm(nx),cp9(nx),cp1(nx),cp2(nx),cp3(nx),cp4(nx),cp5(nx),cp6(nx)</td>
<td>coefficients for 4th order specific heat polynomial curve as function of temperature. The Cp curve is generated by a forth order polynomial defined in McBride. The specific heat, enthalpy and entropy are defined in Equation (30) and Equation (49) where cp5(nx) is a constant which combines the heat of formation of each species at 298.15 K and an integration constant for the enthalpy and cp9(nx) is an integration constant for the entropy. specmm(nx)=species name (up to 80 characters).</td>
</tr>
<tr>
<td>heading:</td>
<td>title of the property (up to 80 characters)</td>
</tr>
<tr>
<td>specmm(nx),eu0(nx),eu1(nx),eu2(nx),eu3(nx),eu4(nx)</td>
<td>coefficients for 4th order dynamic viscosity polynomial curve function of temperature</td>
</tr>
<tr>
<td>heading:</td>
<td>title of the property (up to 80 characters)</td>
</tr>
<tr>
<td>specmm(nx),ed0(nx),ed1(nx),ed2(nx),ed3(nx),ed4(nx)</td>
<td>coefficients for 4th order thermal conductivity polynomial curve function of temperature, Equation (41)</td>
</tr>
<tr>
<td>heading:</td>
<td>title of the property (up to 80 characters)</td>
</tr>
<tr>
<td>specmm(nx),sigm(nx)</td>
<td>species name, effective molecular collision area, Equation (45)</td>
</tr>
<tr>
<td>heading:</td>
<td>title of the property (up to 80 characters)</td>
</tr>
<tr>
<td>specmm(nx),edk(nx)</td>
<td>species name, effective temperature(normalized depth of potential energy for a species), Equation (45)</td>
</tr>
<tr>
<td><strong>Kinetic Information</strong>*</td>
<td>See Section 4.4</td>
</tr>
<tr>
<td>heading:</td>
<td>title of the property (up to 80 characters)</td>
</tr>
<tr>
<td>pcri,tcrl,boil-tmelt</td>
<td>Liquid phase critical property information</td>
</tr>
<tr>
<td>pcri-liquid phase critical pressure (N/m$^2$)</td>
<td></td>
</tr>
<tr>
<td>tcrl-liquid phase critical temperature (K)</td>
<td></td>
</tr>
<tr>
<td>boil-liquid phase boiling point temperature (K)</td>
<td></td>
</tr>
<tr>
<td>tmelt-liquid phase melting point temperature (K)</td>
<td></td>
</tr>
<tr>
<td>hfg,cppar,rlpar,rhohipar</td>
<td>hfg- latent heat of vaporization (J/kg)</td>
</tr>
<tr>
<td>cppar-liquid phase specific heat (J/kg-K)</td>
<td></td>
</tr>
<tr>
<td>rlpar-liquid phase conductivity (J/m-s-K)</td>
<td></td>
</tr>
<tr>
<td>rhohipar-liquid phase density (kg/m$^3$)</td>
<td></td>
</tr>
<tr>
<td>vp1,vp2,vp3,vp4</td>
<td>Critical pressure curve fit coefficients, Equation (18)</td>
</tr>
</tbody>
</table>

4.2.6 User Defined Spray Data: spray.dat

When nsdata=1 and ispray=1 are specified, the code expects users to supply the spray information from the file, spray.dat. The format is described as follows.
read(iunit,*) npgrp

do np=1,npgrp
  read(iunit,*)
  xpar0(np,1),xpar0(np,2),xpar0(np,3),
  velpar0(np,1),velpar0(np,2),velpar0(np,3),
  tsrp0(np),radpar0(np),numpar(np)
end do

where
npgrp = number of group of particles for spray
xpar0(np,1) = x coordinate of the initial spray location (m)
xpar0(np,2) = y coordinate of the initial spray location (m)
xpar0(np,3) = z coordinate of the initial spray location (m)
velpar0(np,1) = x component of the spray injection velocity (m/s)
velpar0(np,2) = y component of the spray injection velocity (m/s)
velpar0(np,3) = z component of the spray injection velocity (m/s)
tsrp0(np) = liquid temperature (K) for each group of spray
radpar0(np) = radius of each drop size (m)
numpar(np) = number of droplets per spray group (#/s)

4.3 Output Data Files

The ALLSPD-3D code generates up to eleven output data files. The input data is echoed out to out.dat. The files, allspd.PFV, allspd.PFS1, and allspd.PFS2 contain solutions in PLOT3D format. If spray was used in the calculation, the spray data files loss.dat, allspd.SPY are also generated. The restart information is located in the save.dat file. The convergence histories are written out to the res.dat and reske.dat files. In addition, the turbulence model restart information is written to the face.dat file and the pre-processed geometry information is written to the ynbl.dat file.

4.3.1 Plot3d Data Files

There are three plot3d files. One is the vector file, allspd.PFV, and the other two are scalar files, allspd.PFS1 and allspd.PFS2. The contents for each file depends on the user inputs specified in allspd.dat

63
read(iunit,*)nblk
read(iunit,*)(imax(n),jmax(n),kmax(n),n=1,nblk)
do n=1,nblk
read(iunit,*)fm0,angle,reno,time
read(iunit,*)
(((q1(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
(((q2(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
(((q3(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
(((q4(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n)),
(((q5(i,j,k),i=1,imax(n)),j=1,jmax(n)),k=1,kmax(n))
end do

where
fm0 = the free stream Mach number
angle = flow angle
reno = flow Reynolds number
time = time
q1 to q5 = are the flow variables defined in the allspd.dat

4.3.2 Spray Data Files

All spray files are generated and read within the subroutine SPRAY. The spray results are written into two files. The file count.dat contains a brief summary of the initial distribution of the spray particles groups. The file loss.dat contains a summary of the spray group time integration history. This file also contains the number of time integrations required for a fuel droplet to vaporize or to hit a particular boundary. The ALLSPD-3D spray model assumes that complete vaporization occurs when a drop hits a wall. It is also assumed that, when a droplet passes through the exit boundary, it is no longer part of the calculation (i.e., particle mass is lost). The current code does not continue to track the particle trajectory once it passes through the interface between blocks (if more than one block is used in the calculation). A true multi-block spray code will be released in the future. Fortunately, in many practical spray applications, the liquid particles evaporate rather quickly before hitting any interface boundary. The maximum time integration limit written out to the loss.dat file should be used to set the integration limit in the parameter statement, allspd.h, when
compiling. Users can also try to increase \textit{nmax} to a large number. The spray data are written into \textit{allspd.SPY} and its format is as follows.

\begin{verbatim}
write(iunit, *) npgrp
do np=1,npgrp
write(iunit, *) np,npmark(np)
write(iunit, *)
xplot(npidx, 1),xplot(npidx, 2),xplot(npidx, 3),
rplot(npidx),tplot(npidx),dplot(npidx),npidx=1,npmark(np)
end do
\end{verbatim}

where
\begin{itemize}
\item \texttt{npgrp} = number of group of particles for spray
\item \texttt{npmark(np)} = the last number of time integration for each group of particle
\item \texttt{xplot(npidx, 1)} = x-coordinate of spray trajectory (m)
\item \texttt{xplot(npidx, 2)} = y-coordinate of spray trajectory (m)
\item \texttt{xplot(npidx, 3)} = z-coordinate of spray trajectory (m)
\item \texttt{rplot(npidx)} = radius of the particle along its trajectory
\item \texttt{tplot(npidx)} = droplet surface temperature along its trajectory (K)
\item \texttt{dplot(npidx)} = integration time step along its trajectory (s)
\end{itemize}

4.3.3 Restart File

The primary restart file is called \textit{save.dat}. This file, and the files \textit{face.dat} and \textit{ynbl.dat}, are the only files required to restart a previous ALLSPD-3D run. They contain all necessary data to restart the program. It should be noted that the \textit{save.dat} files generated by serial mode and parallel mode execution are not compatible. Therefore, users should not try to restart a run in serial mode with the restart file that is generated by a parallel version, and \textit{vice versa}. The file \textit{save.dat} is generated in subroutine SAVE and is read by subroutine RESTART. The file \textit{face.dat} is generated in subroutine SAVE and is read in by subroutine INIT. The file \textit{ynbl.dat} is generated in subroutine PREGRID and is read in by subroutine MESH.

4.3.4 Residual File

The code writes the convergence history to both the computer screen and files named \textit{res.dat} and \textit{reske.dat}. The residual shown on the computer screen is the
L2 norm of the total residual for flow variables and the residuals shown in the 
res.dat contain residual for different flow variables. The write statement is 
\texttt{write(iunit,*) itertot, rescon, res\_p, res\_u, res\_v, res\_w, res\_h} where

\texttt{itertot}: iteration number
\texttt{rescon}: L2 norm of residual for flow variables
\texttt{res\_p}: residual for pressure
\texttt{res\_u}: residual for u velocity
\texttt{res\_v}: residual for v velocity
\texttt{res\_w}: residual for w velocity
\texttt{res\_h}: residual for energy
The reske.dat contains residual for turbulent variables, i.e., $\kappa$ and $\varepsilon$.

4.4 Typical Chemistry Input File Structure

The most important rule to chemistry inputs is that N$_2$ must be the last spe-
cies in the input file. For turbulent chemistry model calculations, the first spe-
cies and second species must be fuel and air, respectively.
### 4.4.1 One-Step Global Chemistry Mode

**TABLE 6. Typical Single-step Chemistry Input File**

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>c methanol chemistry</code></td>
<td>Title</td>
</tr>
<tr>
<td><code>1</code></td>
<td>Number of reactions</td>
</tr>
<tr>
<td><code>c nr ar(nr) ear(nr) tfr</code></td>
<td>Heading: Title for Forward reaction rates</td>
</tr>
<tr>
<td><code>1 4.6e11 -15098.1 0.0</code></td>
<td>Reaction number, Pre-exponent, Activation Temperature(K), Temperature Correction Factor</td>
</tr>
<tr>
<td><code>c nr br(nr) ebr(nr) tbr</code></td>
<td>Heading: Title for Backward reactions</td>
</tr>
<tr>
<td><code>c 1 ch3oh + 1.5o2 -&gt; co2 + 2h2o</code></td>
<td>Title, helpful to write the reaction mechanism (blank lines are acceptable)</td>
</tr>
<tr>
<td><code>1 0.0 0.0</code></td>
<td>Reaction number, Pre-exponent, Activation Temperature(K), Temperature Correction Factor (note this is forward only reaction and GUI will not generate this line!)</td>
</tr>
<tr>
<td><code>c nr xkstp(nr,ns) ns-&gt;</code></td>
<td>Heading: Forward Stoichiometric Coefficients</td>
</tr>
<tr>
<td><code>1 11 5000</code></td>
<td>Reaction number, Forward Stoichiometric Coefficients (There should be npe of them)</td>
</tr>
<tr>
<td><code>c nr xkstp(n,ns) ns-&gt;</code></td>
<td>Heading: Backward Stoichiometric Coefficients</td>
</tr>
<tr>
<td><code>1 00 120</code></td>
<td>Reaction number, Backward Stoichiometric Coefficients (There should be npe of them)</td>
</tr>
<tr>
<td><code>c nr third0(nr) third(nr,ns) ns-&gt;</code></td>
<td>Heading: Title for Correction Mechanism (Note, this input is to used to change the power of the reaction constant)</td>
</tr>
<tr>
<td><code>1 -1.0 1.0 1.0 0.0 0.0 0.0</code></td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in forward direction (for the npe species)</td>
</tr>
<tr>
<td><code>1 -1.0 0.0 0.0 0.0 0.0 0.0</code></td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in Reverse direction (for the npe species)</td>
</tr>
</tbody>
</table>

**Note**

Third0(1) = -1, Global type kinetics
Third0(1) = 0, Modified kinetics with fractional power
Third0(1) >0 detailed kinetics with 3-body corrections

\[ K_d = A \exp(E_a/RT) T^T (ch3oh)^1 [O2]^1 \]

\[ K_d = 0.0 \]

Mechanism of Westbrook and Dryer\(^{37}\)
### 4.4.2 Simple Two-Step Global Chemistry Model

#### TABLE 7. Typical Two-step Chemistry Input File

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>cTwo-step propane chemistry</td>
<td>Title</td>
</tr>
<tr>
<td>2</td>
<td>Number of reactions</td>
</tr>
<tr>
<td>c 1 c3h8 + 7/2 o2 -&gt; 3co + 4h2o</td>
<td>Title, helpful to write the reaction mechanism (blank lines are acceptable)</td>
</tr>
<tr>
<td>c 2 co + 5/2 o2 &lt;-&gt; co2</td>
<td></td>
</tr>
<tr>
<td>c nr ar(nr) ear(nr) tfrr</td>
<td>Heading Title for Forward reaction rates</td>
</tr>
<tr>
<td>1 5.62e09 -15098.1 0.0</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction</td>
</tr>
<tr>
<td>2 2.24e12 -20140.0 0.0</td>
<td></td>
</tr>
<tr>
<td>c nr br(nr) ebr(nr) thr</td>
<td>Heading Title for Backward reactions</td>
</tr>
<tr>
<td>1 0.0e00 0.0 0.0</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction factor (reaction one is a global reaction and therefore does not have any reverse reactions)</td>
</tr>
<tr>
<td>2 5.00e08 -20140 0.0</td>
<td></td>
</tr>
<tr>
<td>c nr xkstp(nr,ns) ns-&gt;</td>
<td>Heading for Forward Stoichiometric Coefficients</td>
</tr>
<tr>
<td>1 1 3.5 0 0 0 0</td>
<td>Reaction number, Forward Stoichiometric Coefficients (There should be nspe of them)</td>
</tr>
<tr>
<td>2 0 0.5 1.0 0 0 0</td>
<td></td>
</tr>
<tr>
<td>c nr xkstpp(nr,ns) ns-&gt;</td>
<td>Heading for Backward Stoichiometric Coefficients</td>
</tr>
<tr>
<td>1 0 0 3 0 4 0</td>
<td>Reaction number, Backward Stoichiometric Coefficients (There should be nspe of them)</td>
</tr>
<tr>
<td>2 0 0 0 1.0 0 0</td>
<td></td>
</tr>
<tr>
<td>c nr third0(nr) third(nr,ns) ns-&gt;</td>
<td>Heading Title for Correction Mechanism (This input is used to change the power of the reaction constant)</td>
</tr>
</tbody>
</table>
### TABLE 7. Typical Two-step Chemistry Input File

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.0 0.1 1.65 0.0 0.0 0.0 0.0</td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in forward and backward direction (Note that this mechanism required modified power for both reactions)</td>
</tr>
<tr>
<td>1 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>2 0.0 0.0 0.25 1.0 0.0 0.5 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>2 0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:**
- ThirdO(1) = -1, Global type kinetics
- ThirdO(1) = 0, Modified kinetics with fractional power
- ThirdO(1) >0 detailed kinetics with 3-body corrections

\[
\text{c3h8} + 7/2 \text{O2} \rightarrow 3\text{co} + 4\text{H2O}
\]

\[
K_f = A \exp(E_f/(RT)) \tau^0 [\text{c3h8}]^{2.1} [\text{O2}]^{1.65}
\]

\[
K_b = 0.0
\]

\[
\text{co} + .5\text{O2} \leftrightarrow \text{co2}
\]

\[
K_f = A \exp(E_f/(RT)) \tau^0 [\text{CO}]^1 [\text{H2O}]^{0.5} [\text{O2}]^{0.25}
\]

\[
K_b = A \exp(E_b/(RT)) \tau^0 [\text{CO2}]^1
\]

Mechanism of Westbrook and Dryer\(^{27}\)

---

69
### 4.4.3 Typical Detailed Chemistry Input file

**TABLE 8. Detailed Chemistry Input File**

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>c Large complex chemistry</td>
<td>Title</td>
</tr>
<tr>
<td>18</td>
<td>Number of reactions</td>
</tr>
<tr>
<td>c1 ( \text{H}_2 + \text{OH} \leftrightarrow \text{H}_2\text{O} + \text{H} )</td>
<td>Title, helpful to write the reaction mechanism (blank lines are acceptable)</td>
</tr>
<tr>
<td>c2 ( \text{O}_2 + \text{H} \leftrightarrow \text{OH} + \text{O} )</td>
<td></td>
</tr>
<tr>
<td>c3 ( \text{H}_2 + \text{O} \leftrightarrow \text{OH} + \text{H} )</td>
<td></td>
</tr>
<tr>
<td>c4 ( \text{O}_2 + \text{H} \leftrightarrow \text{HO}_2 )</td>
<td></td>
</tr>
<tr>
<td>c5 ( \text{H}_2 + \text{M} \leftrightarrow \text{H} + \text{H} )</td>
<td></td>
</tr>
<tr>
<td>c6 ( \text{O}_2 + \text{M} \leftrightarrow \text{O} + \text{O} )</td>
<td></td>
</tr>
<tr>
<td>c7 ( \text{H} + \text{HO}_2 \leftrightarrow \text{H}_2 + \text{O} )</td>
<td></td>
</tr>
<tr>
<td>c8 ( \text{H}_2\text{O} + \text{M} \leftrightarrow \text{OH} + \text{H} )</td>
<td></td>
</tr>
<tr>
<td>c9 ( \text{H}_2\text{O} + \text{O} \leftrightarrow 2\text{OH} )</td>
<td></td>
</tr>
<tr>
<td>c10 ( \text{O} + \text{HO}_2 \leftrightarrow \text{O}_2 + \text{OH} )</td>
<td></td>
</tr>
<tr>
<td>c11 ( \text{H} + \text{HO}_2 \leftrightarrow 2\text{OH} )</td>
<td></td>
</tr>
<tr>
<td>c12 ( \text{OH} + \text{HO}_2 \leftrightarrow \text{O} + \text{H}_2\text{O} )</td>
<td></td>
</tr>
<tr>
<td>c13 ( \text{H}_2 + \text{H}_2\text{O}_2 \leftrightarrow \text{H}_2\text{O} + \text{HO}_2 )</td>
<td></td>
</tr>
<tr>
<td>c14 ( \text{OH} + \text{H}_2\text{O}_2 \leftrightarrow \text{H}_2\text{O} + \text{H}_2\text{O}_2 )</td>
<td></td>
</tr>
<tr>
<td>c15 ( 2\text{HO}_2 + \text{M} \leftrightarrow \text{O}_2 + \text{H}_2\text{O}_2 )</td>
<td></td>
</tr>
<tr>
<td>c16 ( \text{H} + \text{H}_2\text{O}_2 \leftrightarrow \text{OH} + \text{H}_2\text{O} )</td>
<td></td>
</tr>
<tr>
<td>c17 ( \text{H}_2\text{O}_2 + \text{M} \leftrightarrow 2\text{OH} )</td>
<td></td>
</tr>
<tr>
<td>c18 ( \text{H} + \text{O} \leftrightarrow \text{OH} + \text{M} )</td>
<td></td>
</tr>
<tr>
<td>( \text{c nr ar(nr) ear(nr) tfrr} )</td>
<td>Heading Title for Forward reaction rates</td>
</tr>
<tr>
<td>Input file content</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------</td>
</tr>
<tr>
<td>1 4.74E+10 -3068.95 0.0</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction</td>
</tr>
<tr>
<td>2 1.85E+11 -8233.65 0.0</td>
<td></td>
</tr>
<tr>
<td>3 4.2E+11 -6919.98 0.0</td>
<td></td>
</tr>
<tr>
<td>4 1.35E+9 503.27 0.0</td>
<td></td>
</tr>
<tr>
<td>5 2.2E+11 -4834.04 0.0</td>
<td></td>
</tr>
<tr>
<td>6 1.8E+15 -59396.074 1.0</td>
<td></td>
</tr>
<tr>
<td>7 1.3E+11 -1509.814 0.0</td>
<td></td>
</tr>
<tr>
<td>8 1.3E+12 -52913.94 0.0</td>
<td></td>
</tr>
<tr>
<td>9 6.8E+10 -9242.577 0.0</td>
<td></td>
</tr>
<tr>
<td>10 5.0E+10 -503.27 0.0</td>
<td></td>
</tr>
<tr>
<td>11 2.04E+11 -538.5 0.0</td>
<td></td>
</tr>
<tr>
<td>12 8.0E+9 -1499.75 0.0</td>
<td></td>
</tr>
<tr>
<td>13 7.91E+10 -12581.78 0.0</td>
<td></td>
</tr>
<tr>
<td>14 6.1E+9 -719.678 0.0</td>
<td></td>
</tr>
<tr>
<td>15 1.8E+9 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>16 7.8E+8 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>17 1.44E+14 -22903.875 0.0</td>
<td></td>
</tr>
<tr>
<td>18 7.1E+12 0.0 0.0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c nr br(nr) ebr(nr) tbr</th>
<th>Heading Backward reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2.03138E+12 -10985.23 0.274934</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction</td>
</tr>
<tr>
<td>2 4.52315E+08 412.94 0.410439</td>
<td></td>
</tr>
<tr>
<td>3 2.2825E+11 -5998.66 0.0246801</td>
<td></td>
</tr>
<tr>
<td>4 5.1015E+13 -25765.13 0.423663</td>
<td></td>
</tr>
<tr>
<td>5 4.3848E+7 416.66 0.0320618</td>
<td></td>
</tr>
<tr>
<td>6 1.51713E+9 840.326 0.525515</td>
<td></td>
</tr>
<tr>
<td>7 1.275E+10 -27672.1 0.4273321</td>
<td></td>
</tr>
<tr>
<td>8 5.72655E+6 7491.96 0.313192</td>
<td></td>
</tr>
<tr>
<td>9 8.613E+8 -404.787 0.250384</td>
<td></td>
</tr>
<tr>
<td>10 2.6504E+9 -25743.37 0.403293</td>
<td></td>
</tr>
<tr>
<td>11 2.7688E+7 -17119.5 0.8083150</td>
<td></td>
</tr>
<tr>
<td>12 4.03565E+10 -35609.15 0.130335</td>
<td></td>
</tr>
<tr>
<td>13 5.07334E+12 -3160.89 0.411882</td>
<td></td>
</tr>
<tr>
<td>14 3.761E+9 -18043.58 0.14644</td>
<td></td>
</tr>
<tr>
<td>15 1.0845E+10 -16734.1 0.0205195</td>
<td></td>
</tr>
<tr>
<td>16 1.04212E+5 -33982.3 0.89952</td>
<td></td>
</tr>
<tr>
<td>17 8.487E+4 3519.125 0.0</td>
<td></td>
</tr>
<tr>
<td>18 2.1057E+16 -51573.5 1.0666587</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c nr xkstp(nr,ns) ns-&gt;</th>
<th>Heading</th>
</tr>
</thead>
</table>

71
# TABLE 8. Detailed Chemistry Input File

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reaction number, Forward Stoichiometric Coefficients (There should be no of them)</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>c nr xstpp(m nr,n) ns-&gt;</th>
<th>Heading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reaction number, Backward Stoichiometric Coefficients (There should be no of them)</td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 8. Detailed Chemistry Input File

<table>
<thead>
<tr>
<th>Input file content</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>c nr third1(nr) third2(nr,nx) ns-&gt;</td>
<td>Heading</td>
</tr>
<tr>
<td>1 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in forward direction (for the nspe species)</td>
</tr>
<tr>
<td>2 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>3 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>4 2.0 2.5 0.3 0.0 20.3 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>5 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>6 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>7 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>8 2.0 3.2 3.2 15.7 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>9 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>10 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>11 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>12 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>13 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>14 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>15 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>16 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>17 1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
<tr>
<td>18 2.0 4.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0</td>
<td></td>
</tr>
</tbody>
</table>

Taken from Lee\textsuperscript{35}
V: Graphical User Interface (GUI)

This is a preliminary manual for alpha version 0.1 of the ALLSPD-3D Graphical User Interface (GUI). This GUI corresponds to version 1.0 of the ALLSPD-3D Code.

5.1 Installation

The ALLSPD-3D GUI was written in C and requires X/Motif Release 1.1. It has been built successfully on the following platforms: IBM RS6000, SGI, and also beta ported to Solaris and HPUX.

To install the GUI, first create a separate directory and untar the GUI tar file. This will create a directory tree with the following structure:

```
|--- CHANGES
|--- Makefile
|--- Manual
    |--- bin/
|--- chemistry/
    |--- common/
    |--- files/
    |--- grid/
|--- help_files/
    |--- include/
    |--- model/
    |--- solve/
|--- species/
    |--- view/
|--- xsmallspd.c
```

Set the XMALLSPD_ROOT environment variable to point to the directory where the GUI will be built. For instance:

```
setenv XMALLSPD_ROOT /u/quealy/gui
```

Also, modify your path to include $XMALLSPD_ROOT and $XMALLSPD_ROOT/bin. For example:

```
set path=( $path $XMALLSPD_ROOT $XMALLSPD_ROOT/bin )
```

These can be included in your .cshrc file so they are issued every time a new shell is created.
A top level Makefile will compile the entire directory structure and create a GUI executable named ‘xsmallsp’. At this point, you can type ‘make’, and you should see the code compile. Sample output from this stage is included below.

```
>make
  dir_list=\file grid model solve view common ;
  for dir in $dir_list ; do
    echo making GUI in directory $dir : \
    ( cd $dir ; make all ) ;
  done
making GUI in directory file
  cc -g -I/usr/arc/insp95gui/include -c file.c
  echo "FILE module built..."
FILE module built...
making GUI in directory grid
  cc -g -I/usr/arc/insp95gui/include -c grid.c
  echo "GRID module built..."
GRID module built...
making GUI in directory model
  cc -g -I/usr/arc/insp95gui/include -c model.c
  cc -g -I/usr/arc/insp95gui/include -c geom_info.c
  cc -g -I/usr/arc/insp95gui/include -c geom_scale.c
  cc -g -I/usr/arc/insp95gui/include -c geom_index.c
  cc -g -I/usr/arc/insp95gui/include -c geom_file.c
  cc -g -I/usr/arc/insp95gui/include -c geom_fill.c
  cc -g -I/usr/arc/insp95gui/include -c geom_util.c
  cc -g -I/usr/arc/insp95gui/include -c prop_flow.c
  cc -g -I/usr/arc/insp95gui/include -c prop_spec.c
  cc -g -I/usr/arc/insp95gui/include -c prop_spec_species.c
  cc -g -I/usr/arc/insp95gui/include -c prop_util.c
  cc -g -I/usr/arc/insp95gui/include -c prop_lp.c
  cc -g -I/usr/arc/insp95gui/include -c bc.c
  echo "MODEL module built..."
MODEL module built...
making GUI in directory solve
  cc -g -I/usr/arc/insp95gui/include -c solve.c
  cc -g -I/usr/arc/insp95gui/include -c control.c
  cc -g -I/usr/arc/insp95gui/include -c output.c
  cc -g -I/usr/arc/insp95gui/include -c solution.c
  echo "SOLVE module built..."
SOLVE module built...
making GUI in directory view
  cc -g -I/usr/arc/insp95gui/include -c view.c
  echo "VIEW module built..."
VIEW module built...
making GUI in directory common
  cc -g -I/usr/arc/insp95gui/include -c util.c
  echo "COMMON module built..."
COMMON module built...
make smallsp
  cc -g -I/usr/arc/insp95gui/include -c smallsp.c
  cc -o smallsp $smallsp smallsp.o file.c grid.c model.c solve.c view.c common.c -L/usr/arc/insp95gui/lib -L/usr/arc/insp95/gui/lib -L/usr/arc/insp95/gui/lib -L/usr/arc/insp95gui/lib -L/usr/arc/insp95gui/lib -L/usr/arc/insp95gui/lib -L/usr/arc/insp95gui/lib -L/usr/arc/insp95gui/lib -L/usr/arc/insp95gui/lib
```

At the time of writing this manual, several assumptions have been made, especially pertaining to running the parallel version of the code.
Assumptions:

| If PVM is used, it is installed and the proper PVM environment variables have been set. |
| Both the GUI server and all compute nodes share a common file system. |
| User ids are identical across all platforms. |
| If LSF is used, it is available on the GUI host as well as the compute nodes. |
| Pre and post processors exist on the host where the GUI is running, and all executables are in the user’s path. |
| If no reactions are specified in a prop.dat file, it is assumed that there are no liquid properties. |

The GUI sets default values for most variables used by the ALLSPD-3D code. These default values are set in two locations in the GUI code:

`model/model.c: MODEL_InvokeModelSimDefaults()`
`solve/solve.c: SOLVE_InvokeSolveSimDefaults()`

The user may edit these routines and adjust the default values if desired, and then recompile the GUI to generate a new `xsmallsp` executable.

### 5.2 General Overview

The ALLSPD-3D GUI allows the user to create the following ALLSPD-3D input files:

<table>
<thead>
<tr>
<th>allspd.dat</th>
<th>(Input file)</th>
</tr>
</thead>
<tbody>
<tr>
<td>prop.dat</td>
<td>(Property file)</td>
</tr>
<tr>
<td>gridlink.dat</td>
<td>(Grid linkage file)</td>
</tr>
</tbody>
</table>
The GUI allows a user to start either the serial or parallel code on local or remote hosts, using several startup options. This is subject to the assumptions listed in the installation section.

The ALLSPD-3D GUI also gives a user access to a number of pre- and post-processing tools, again subject to the assumptions in the installation section.

The following 'Scorecard' is included for people who wish to work with the GUI source code itself. This is a cross reference between variable names used by the ALLSPD-3D code, and their corresponding GUI variable names.
The main menu of the ALLSPD-3D GUI is illustrated in Figure 7. The remaining sections of this chapter highlight each of the options from this main menu briefly describes each function.

![ALLSPD-3D GUI Main Menu Window](image)

**FIGURE 7. ALLSPD-3D GUI Main Menu Window**

### 5.3 File

The File section of the main menu allows you to read and write the three main ALLSPD-3D files (input, property and grid linkage files), and also to exit the GUI.

For clarification in this manual, the files will be referred to by their default names: `allspd.dat` (input file), `gridlink.dat` (grid linkage file), and `prop.dat` (property file). Note that the GUI accepts and will generate filenames which differ from the default. The default names are used in this manual for simplicity.

The order of reading the three input files can be important. Since species information may be included in each of these three files, it is wise to read an existing `prop.dat` file first. If `prop.dat` is not read first, all species information at the end of the `allspd.dat` file and the `gridlink.dat` file is ignored, and a warning message is displayed.

When reading the `gridlink.dat` file, the `grid.dat` filename is also requested. The GUI does not read the entire `grid.dat` file. It simply reads the initial information in the file to verify that the `gridlink.dat` file matches the `grid.dat` file.
5.4 Grid

The Grid section of the main menu gives the user access to grid generation packages. Currently the GUI allows starting the CFD-GEOM package developed by CFD Research. Gridgen access has not yet been implemented.

The user may set the XMALLSPD_CFDGEOM_EXE environment variable to point to the location of the CFD-GEOM executable. If this environment variable is not set it is assumed the default name ‘CFD-GEOM’ is used, and that the executable is in the user’s path. It is assumed that licenses to access CFD-GEOM are in place, and that CFD-GEOM will run on the same host as the GUI.

5.5 Model

The model menu is illustrated in Figure 8. Each of the options from this model menu are briefly described below.

![Figure 8. ALLSPD-3D GUI Model Menu Window](image)

5.5.1 Model/File

The File option from the Model Module simply allows you to close the Model Module window.
5.5.2 Model/Manipulate Grid

The 'Manipulate Grid' option from the Model Module window allows the user to import a grid developed externally, display grid information, perform grid scaling if necessary, and define internal blockages.

The GUI currently supports PLOT3D and CFD-GEOM grid files. The files can be 2D or 3D files. All 2D files are converted by the GUI to a 3D format usable by ALLSPD-3D. The option to import a Gridgen grid file has not been implemented yet.

The 'Grid Information' option displays grid parameter information for each grid block. If a grid file has not been specified, an error message is displayed.

The 'Grid Scaling' option allows the user to scale the current grid from either inches, feet or centimeters to meters (which is what ALLSPD-3D expects). The user is allowed the option to select a new file name. If a new file name is not selected, the original grid is overwritten.

**WARNING:** Since this is an alpha version of the GUI, it is STRONGLY ADVISED that you do not overwrite the original grid. Any errors occurring during the scaling process could destroy the original file. To prevent this from happening, keep a copy of the original files in a protected place with read only access.

The 'Define Internal Blockages' option allows the user to define any number of internal blockages for each block.

5.5.3 Model/Problem Description

The 'Problem Description' section of the Model Module allows the user to select several problem options. The options to select are:

<table>
<thead>
<tr>
<th>Turbulent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reacting</td>
</tr>
<tr>
<td>Spray</td>
</tr>
<tr>
<td>Eddy Breakup Model</td>
</tr>
</tbody>
</table>

81
If the application is Turbulent, the user has the additional option of using a variable \( C_\mu \) in the turbulent eddy viscosity formulation or the CMOTT \( \varepsilon \) boundary condition in place of the default \( \varepsilon \) boundary condition.

### 5.5.4 Model/Properties

The ‘Properties’ section of the Model Module is the most complex section of the ALLSPD-3D GUI. Under this option, the user may specify Fluid properties, Spray properties, Chemistry properties (Thermal & Kinetics), and Ignition properties. Each of these areas is described in detail below.

#### 5.5.4.1 Model/Properties/Fluid

Under the Fluid Properties window, the user is able to specify various initial fluid properties, including:

<table>
<thead>
<tr>
<th>Initial Velocity ((x,y,z))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Temperature</td>
</tr>
<tr>
<td>Initial Pressure</td>
</tr>
<tr>
<td>Reference length</td>
</tr>
</tbody>
</table>

Also, the user is given four options for setting initial flow conditions:

<table>
<thead>
<tr>
<th>Initial velocity plane is defined ONLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform Initial conditions</td>
</tr>
<tr>
<td>User Defined Initial conditions (\text{expects uwt file})</td>
</tr>
<tr>
<td>Initial plane from Swirl Cups (\text{expects uwt file})</td>
</tr>
</tbody>
</table>

At the time of writing this manual the \(\text{uwt.dat}\) filename must be entered via the Model/Properties/Fluid window if the \(\text{allspd.dat}\) file indicates that such a file is to be read. If the ALLSPD-3D code expects the filename but it has not been entered through the GUI, an error message is displayed when the user attempts to start the code.
5.5.4.2 General Information on Species Database

The GUI relies on a species database to obtain property information for the prop.dat file. The species database that is distributed with the GUI source code is located in $XMALSPD_ROOT/species/species_list. There are currently 31 species in this database.

The user is able to customize this species database by entering information via the GUI and storing the newly created species information into the database. This allows the user to add to this database as desired.

**WARNING:** Since this is an alpha version of the GUI, it is STRONGLY ADVISED that you keep a backup copy of any species database that is built by the user. Any errors occurring during the writing of new information to the existing species database could destroy the existing database. To prevent this from happening, keep a copy of a customized species database in a protected place with read only access. Also, if you are in the process of creating a species database, save it frequently and check for errors. And do not forget to make backups!

5.5.4.3 Locating the Master Species Database

The GUI follows a pre-defined search path when trying to locate the Master Species Database. First the environment variable XMALSPD_SPECIES_LIST is examined. If this environment variable has been defined, the species database is read from this location. If not defined, the search path continues in the following order:

- .xmalspd/species_list
- $HOME/.xmalspd/species_list
- $XMALSPD_ROOT/species/species_list

The first species database file which is located is then used to read the master species list into the database.

When updating the master list, the file search order is as follows:
$XMALSPD_SPECIES_LIST
.xmallspd/species_list
$HOME/.xmallspd/species_list

If the file is not located in any of these locations, this means that the original distribution species list has been altered. To prevent overwriting the original, the .xmallspd directory is created in the current working directory, and the species list is written there (.xmallspd/species_list).

The GUI uses the chemical name for searching the database, so this must always be unique. Also, at the time of writing this manual, the GUI is ‘case sensitive.’ Most species in the database have been entered with uppercase letters. Some however require lowercase letters (such as iC4H10 for iso-Butane).

5.5.4.4 Locating the Master ‘Default Chemistry Package’ Database

The GUI follows a pre-defined search path when trying to locate the Master ‘Default Chemistry Package’ Database. First the environment variable XMALLSPD_CHEMISTRY_LIST is examined. If this environment variable has been defined, the DefaultChemList (and any selected packages) are read from this location. If not defined, the search path continues in the following order:

.xmallspd/DefaultChemList
$HOME/.xmallspd/DefaultChemList
$XMALSSPD_ROOT/chemistry/DefaultChemList

The first DefaultChemList database file which is located is then used to read the master ‘Default Chemistry Package’ list into the database.

When updating the master ‘Default Chemistry Package’ list, the file search order is as follows:

$XMALSSPD_CHEMISTRY_LIST .xmallspd/DefaultChemList
$HOME/.xmallspd/DefaultChemList
If the file is not located in any of these locations, this means that the original distribution DefaultChemList has been altered. To prevent overwriting the original, the .xmallspd directory is created in the current working directory, and the DefaultChemList is written there (.xmallspd/DefaultChemList).

5.5.4.5 Model/Properties/Spray Initialization

The user cannot enter the Spray Initialization window unless the Spray problem option has been set. Once this window is activated, the user may either enter spray initialization information (default), or read this information from a file (spray.dat).

At the time of writing this manual the spray.dat filename must be entered via the Model/Properties/Spray window if the allspd.dat file indicates that such a file is to be read. If the ALLSPD-3D code expects the filename but it has not been entered through the GUI, an error message will be displayed when the user attempts to start the code.

If the user opts to enter the spray initialization information, another window opens allowing the user to enter information such as:

<table>
<thead>
<tr>
<th>Block with nozzle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spray location</td>
</tr>
<tr>
<td>Number of angles and range</td>
</tr>
<tr>
<td>Particle size information</td>
</tr>
<tr>
<td>Spray source term treatment option</td>
</tr>
<tr>
<td>Iteration to start spray</td>
</tr>
<tr>
<td>Spray Frequency</td>
</tr>
<tr>
<td>Spray temperature</td>
</tr>
<tr>
<td>Mass flow rate</td>
</tr>
<tr>
<td>Velocity magnitude</td>
</tr>
</tbody>
</table>

5.5.4.6 Model/Properties/Customize Liquid Species

The user is able to either select the Spray Species from a list, or to customize
the spray property information via the GUI. If the user chooses to select the spray species from a list, the list is built from the current species database, as determined above.

If the user opts to customize the spray species information, this information can then be added to the user’s species database for future use. Properties to enter for customizing include:

<table>
<thead>
<tr>
<th>Species Name</th>
<th>(for example, ‘Oxygen’)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical Name</td>
<td>(for example, ‘O2’)</td>
</tr>
<tr>
<td>Critical Pressure</td>
<td></td>
</tr>
<tr>
<td>Critical Temperature</td>
<td></td>
</tr>
<tr>
<td>Melting Temperature</td>
<td></td>
</tr>
<tr>
<td>Boiling Temperature</td>
<td></td>
</tr>
<tr>
<td>Phase change enthalpy</td>
<td></td>
</tr>
<tr>
<td>Specific heat for liquid phase</td>
<td></td>
</tr>
<tr>
<td>Thermal conductivity for liquid phase</td>
<td></td>
</tr>
<tr>
<td>Density at injection temperature</td>
<td></td>
</tr>
<tr>
<td>Vapor pressure coefficients</td>
<td></td>
</tr>
</tbody>
</table>

5.5.4.7 Model/Properties/Read Default Chemistry Package

This section of the GUI allows the user to read a pre-defined default chemistry package. Several default chemistry packages have been distributed with the ALLSPD-3D GUI. The following packages currently exist:

<table>
<thead>
<tr>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heptane</td>
</tr>
<tr>
<td>Hydrogen Air</td>
</tr>
<tr>
<td>Methanol</td>
</tr>
<tr>
<td>Propane(1step)</td>
</tr>
<tr>
<td>Air</td>
</tr>
</tbody>
</table>
By building the species database and customizing the thermal properties and kinetics, the user can continue to build and add to the default chemistry packages.

5.5.4.8 Model/Properties/Write Default Chemistry Package

This section of the GUI allows the user to write a customized chemistry package to the chemistry database. Thermal and Kinetics information must be specified before the user is allowed to generate a chemistry package. Once the new package is written to the database, the user is able to read the default chemistry package just like the pre-defined packages which were supplied with the GUI.

NOTE: Currently if a name is entered for a new Default Chemistry Package which duplicates a pre-existing package, the GUI posts an error message and expects the user to select a new name. More flexibility may be added at a later date.

5.5.4.9 Model/Properties/Customize Thermal

The ‘Customize Thermal’ option allows the user to either select Thermal Properties from the species database, or to enter new species thermal properties by hand.

If the user chooses to enter the Thermal property information, the following information is requested for the species database:

<table>
<thead>
<tr>
<th>Species name</th>
<th>(for example, Oxygen)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chemical name</td>
<td>(for example, ‘O2’)</td>
</tr>
<tr>
<td>Specific Heat</td>
<td></td>
</tr>
<tr>
<td>Enthalpy constant</td>
<td></td>
</tr>
<tr>
<td>Entropy constant</td>
<td></td>
</tr>
<tr>
<td>Viscosity</td>
<td></td>
</tr>
<tr>
<td>Thermal Conductivity</td>
<td></td>
</tr>
<tr>
<td>Diffusion parameters</td>
<td></td>
</tr>
</tbody>
</table>
There are some restrictions upon the order in which species should be selected. The eddy breakup model assumes that the major fuel constituent is first and that oxygen (O2) is second. This order is also assumed for a global reaction mechanism. In general, if nitrogen is chosen as a component, it should be picked as the last species in the list so that round-off error will be minimized.

5.5.4.10 Model/Properties/Customize Kinetics

The user cannot enter the ‘Customize Kinetics’ window unless the ‘Reacting’ problem option has been set. Once this window is activated, the user is able to enter reaction information.

The user may enter a reaction set name. Also, the user must specify the known reaction rates (forward reaction rate constants only, backward reaction rate constants only, or both) for all reactions.

The user may then add any number of reactions. For each reaction, the user is asked for:

<table>
<thead>
<tr>
<th>Reaction type (Global, Westbrook &amp; Dryer, Finite Rate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Modified Arrhenius rates</td>
</tr>
<tr>
<td>pre-exponential</td>
</tr>
<tr>
<td>temperature exponent</td>
</tr>
<tr>
<td>activation energy</td>
</tr>
<tr>
<td>Backward Modified Arrhenius rates</td>
</tr>
<tr>
<td>pre-exponential</td>
</tr>
<tr>
<td>temperature exponent</td>
</tr>
<tr>
<td>activation energy</td>
</tr>
</tbody>
</table>

And for each species,

<table>
<thead>
<tr>
<th>Forward Stoichiometric Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forward Mass Action Coefficient (if appropriate)</td>
</tr>
<tr>
<td>Backward Stoichiometric Coefficient</td>
</tr>
</tbody>
</table>
5.5.4.11 Ignition

The ‘Ignition’ window allows the user to enter various ignition properties, including:

<table>
<thead>
<tr>
<th>Iteration to apply ignition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration to stop ignition</td>
</tr>
<tr>
<td>Temperature limit</td>
</tr>
<tr>
<td>Spark strength</td>
</tr>
<tr>
<td>Ignition Locations</td>
</tr>
</tbody>
</table>

5.5.4.12 Initial Mass Fractions

Once the user selects species information (via ‘Default Chemistry Package’ or ‘Customize Thermal’), a window requesting mass fraction information is displayed. The user is asked to enter an initial mass fraction value for each species that has been selected.

5.5.5 Model/Boundary Conditions

The ‘Boundary Condition’ window allows the user to specify boundary conditions for each block on each of six faces. It also allows specifying Dilution Holes and Transpiration Cooling areas.

The boundary condition options include:

<table>
<thead>
<tr>
<th>Upstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downstream</td>
</tr>
<tr>
<td>Adiabatic Wall</td>
</tr>
<tr>
<td>Isothermal Wall</td>
</tr>
<tr>
<td>Block Interface</td>
</tr>
<tr>
<td>Symmetry</td>
</tr>
<tr>
<td>Periodic</td>
</tr>
</tbody>
</table>
If a downstream or far field boundary condition is specified, the user is asked for an exit pressure value. If an isothermal wall is specified, the user is asked to enter a wall temperature value. If a block interface is specified, the user is asked to enter the block number of the neighboring block.

If a particular block has one or more dilution holes, the user can enter dilution hole information by pressing the ‘Dilution Hole’ pushbutton. The user may enter any number of dilution holes. The user is asked to enter the following information for each dilution hole:

| Dilution hole location | Mass flow rate | Temperature | Turbulent intensity | Direction cosine (x,y,z) |

If species have been selected, the user is asked to enter dilution mass fractions for each species.

If a particular block has one or more Transpiration Cooling areas, the user can enter transpiration cooling information by pressing the ‘Transpiration Cooling’ pushbutton. The user may enter any number of transpiration cooling areas. The user is asked to enter the following information for each transpiration cooling area:

| Transpiration cooling area location | Mass flow rate | Direction cosine (x,y,z) |

If species have been selected, the user is asked to enter transpiration mass fractions for each species.
5.5.6 Model/Help

Help files are included in the GUI. If a help file does not currently exist, pressing that ‘Help’ button will have no affect.

5.6 Solve

The solve menu is illustrated in Figure 9. Each of the options from this solve menu are briefly described below.

![Figure 9. ALLSPD-3D GUI Solve menu window](image)

5.6.1 Solve/File

The File option from the Solve Module simply allows you to close the Solve Module window.

5.6.2 Solve/Control

The ‘Control’ menu option allows the user to specify whether a restart file is to be used, and to set iteration and other miscellaneous parameters.

The ‘Restart’ option allows the user to specify whether a restart file is to be used. If a restart file is to be used, a file selection box is displayed so the user may enter the name of the restart file. Note that if you are running the parallel code, a restart file exists for each process. These are named:

- `save.dat-001`
- `save.dat-002`
where \( n \) is the number of processes used in this particular calculation. When entering the filename for the restart file in this situation, do not enter a specific file extension. In other words, enter `save.dat` rather than `save.dat-001`.

Also note that a restart file created by the serial code is NOT compatible with the parallel code, and vice versa.

At the time of writing this manual the restart filename must be entered via the Solve/Control window if the `allspd.dat` file indicates that a restart file is to be used. If the ALLSPD-3D code expects the filename but it has not been entered through the GUI, an error message will be displayed when the user attempts to start the code.

The ‘Iterations’ window allows the user to specify the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total number of Iterations</td>
<td></td>
</tr>
<tr>
<td>Iteration frequency for writing a PLOT3D file</td>
<td></td>
</tr>
<tr>
<td>Iteration frequency for writing a restart file</td>
<td></td>
</tr>
<tr>
<td>Number of iterations between turbulent equation iterations</td>
<td></td>
</tr>
<tr>
<td>Number of iterations between solutions to the species continuity equations</td>
<td></td>
</tr>
<tr>
<td>Number of iterations to under relax the time step</td>
<td></td>
</tr>
<tr>
<td>Relaxation factor for time step</td>
<td></td>
</tr>
</tbody>
</table>

The ‘Miscellaneous’ window allows the user to specify the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Viscosity</td>
<td></td>
</tr>
<tr>
<td>CFL Numbers</td>
<td></td>
</tr>
<tr>
<td>Pre-conditioning numbers</td>
<td></td>
</tr>
</tbody>
</table>

If Artificial Viscosity is ‘ON’, the user is asked for 2nd and 4th order coefficients.
5.6.3 Solve/Output

The ‘Output’ window allows the user to specify PLOT3D parameters. The user is given a list of variables which can be written to PLOT3D files. The list includes:

<table>
<thead>
<tr>
<th>Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>U Momentum</td>
</tr>
<tr>
<td>V Momentum</td>
</tr>
<tr>
<td>W Momentum</td>
</tr>
<tr>
<td>Static Temperature</td>
</tr>
<tr>
<td>Gage Pressure</td>
</tr>
<tr>
<td>Static Pressure</td>
</tr>
<tr>
<td>Turbulent CMU</td>
</tr>
<tr>
<td>Static Enthalpy</td>
</tr>
<tr>
<td>Total Enthalpy</td>
</tr>
<tr>
<td>Turbulent Kinetic Energy</td>
</tr>
<tr>
<td>Dissipation</td>
</tr>
<tr>
<td>Turbulent Eddy Viscosity/Laminar Viscosity</td>
</tr>
<tr>
<td>Normal Distance</td>
</tr>
</tbody>
</table>

plus all mass fractions for the selected species.

A maximum of 15 choices can be selected. These are written in the order selected to three PLOT3D files. The first five selections are written to the first PLOT3D file (default = allspd.PFV). For compatibility with CFD-VIEW, this file should include any vector quantities, e.g. momentum, as the 2nd, 3rd, and 4th choices. The second five selections are written to the second PLOT3D file (default = allspd.PFS1). And the third set of five are written to the final PLOT3D file (default= allspd.PFS2). These files are written in ASCII format for compatibility across platforms.

5.6.4 Solve/Solution

The ‘Solution’ option from the Solve menu allows the user to setup an ap-
plication, and start and stop the application.

5.6.4.1 Solve/Solution/Setup Environment

The 'Setup Environment' option allows the user to specify whether the ALLSPD-3D executable has been compiled for 'Serial' or 'Parallel' mode. It also allows the user to specify the job submission mode (Interactive or Batch), and the host selection mechanism.

In interactive mode, the user can specify the hosts to be used, or allow LSF to select the hosts to be used, subject to user specified resource requirements. Refer to the LSF manual for details on specifying resource requirements. An example resource string might be

cs && (type == RS6K)

which might mean that the user wants an RS6000 compute server. In batch mode, the user can submit a job to a specified LSF queue. Again, refer to the LSF manual for details on specifying batch queues.

If the user manually selects hosts to run the application, the GUI searches for a host list specifying existing valid hosts. The XMALLSPD_HOST_LIST environment variable defines the location of this host list. If this environment variable has not been set, the GUI follows a search path through:

.xmallspd/host_list

$HOME/.xmallspd/host_list

If these files cannot be found, the GUI creates a new file in .xmallspd/host_list using the current host as the initial entry.

5.6.4.2 Solve/Solution/Solve

The 'Solve' option allows the user to specify whether the grid pre-processor should be run, and actually submits the job to the selected environment. Note that at this time, the ALLSPD-3D executable must be compiled to match the
grid size and number of species selected by the GUI. Eventually the GUI may query the code to verify that the ALLSPD-3D code matches the GUI inputs, however this has not been implemented at this time.

The Solve option starts a `run-allspd` script which is located in the $XMALLSPD_ROOT/bin directory. The run-allspd script uses `allspd3d` as the default name for the ALLSPD-3D executable. This can be modified by setting the XMALLSPD_EXEC environment variable. `pregrid` is the default name for the grid pre-processor. Again, this can be modified by setting the XMALLSPD_PREGRID environment variable.

The GUI creates three ‘temporary’ input files when the ‘SOLVE’ button is pushed. These files reflect the current values entered via the GUI. These files are named JOB.allspd3d_input.<pid>, JOB.allspd3d_prop.<pid>, and JOB.allspd3d_link.<pid>, where <pid> is the UNIX process id of the run-allspd script. These files are written to the current working directory and are used as input to the ALLSPD-3D code.

### 5.6.4.3 Solve/Solution/Stop

At the time of writing this manual, the Stop option has not been fully implemented. Currently the user may stop a serial or parallel job which is started on the local host (i.e., the host which is running the GUI), and a parallel job which is started on a remote host. Jobs submitted to a batch scheduler should be stopped by using the batch scheduler stop mechanism. Serial jobs which are started on remote hosts must be killed manually using the UNIX ‘kill’ command.

### 5.6.5 Solve/Help

The help files are included in the GUI. If a help file does not currently exist, pressing that ‘Help’ button will have no affect.
5.7 View

The View section of the main menu gives the user access to post-processing packages. Currently the GUI allows starting FAST, and the CFD-VIEW package developed by CFD Research.

The user may set the XMALLSPD_CFDVIEW_EXEC environment variable to point to the location of the CFD-VIEW executable. If this environment variable is not set it is assumed that the default name ‘CFD-VIEW’ is used, and that the executable is in the user’s path. It is assumed that licenses to access CFD-VIEW are in place, and that CFD-VIEW will run on the same host as the GUI.

The user may set the XMALLSPD_FAST_EXEC environment variable to point to the location of the FAST executable. If this environment variable is not set it is assumed that the default name ‘fast’ is used, and that the executable is in the user’s path. It is assumed that licenses to access FAST are in place, and that FAST will run on the same host as the GUI.

5.8 Help

Most of the help files are included in the GUI, however some have not. If a help file does not currently exist, pressing that ‘Help’ button will have no affect.

5.9 Known Bugs

Several locations in the GUI require a user to hit an “Add” button before entering data. If the user enters data without hitting Add, the data appears to be entered, but it is actually lost. This must be modified to prevent a user from entering data if a specific item has not been added.
5.10 Future Plans

The following items may be modified in the future:

The assumptions regarding the GUI host and all compute nodes sharing a common file system should be relaxed to allow an arbitrary file system.

Currently an entry cannot be deleted from the species database or the default chemistry packages. More flexibility to manipulate these databases should be added.
VI : Code Structure and
Hardware
Implementations

6.1 General Structure

The ALLSPD-3D code is not currently considered a "black box" production
code, however, it is not a pure "research" code either. During the course of this
development, we have tried very hard to run test cases by modifying only the
user interface files, not the source code. The current version of the code can run
in either serial or parallel mode. In other words, it can run on a Cray-YMP, a
single workstation (SGI, IBM, HP,...), or a cluster of workstations. Also, the
Graphical User Interface (GUI) allows users to build input data files (grid,
chemistry, flow conditions) and initiate interactive post-processing. This sec-
tion will discuss these features. The ALLSPD-3D code consists of a main pro-
gram and a set of subroutines. The general structure illustrated in Figure 10
shows a flowchart encompassing the entire calculation. Beside each box in the
flow diagram appears the name(s) of the primary subroutine(s) responsible for
the associated task. Short comments in each subroutine source listing also de-
scribe its function.
FIGURE 10. ALLSPD-3D Flow Chart
FIGURE 11. ALLSPD-3D Flow Chart (cont'd)
FIGURE 12. ALLSPD-3D Flow Chart (cont’d)
6.2 Working Units

The M-K-S system of units is used for all variables throughout the code. The units for several variables are listed as follows.

Length: meter (m)
Time: second (s)
Mass: kilogram (kg)
Temperature: Degrees Kelvin (K)
Energy: Joules (J)
Force: Newton (N)
Velocity: meter/second (m/s)
Pressure: N/m$^2$
Density: kg/m$^3$
Specific heat: J/(kg K)
Dynamic viscosity: kg/(m s)
Kinematic viscosity: m$^2$/s

The dimensional units required for chemical kinetic rate pre-exponents $A_f$ and $A_b$ are in the M-K-S system. Typically, the activation energy is normalized by the ideal gas constant and represented as an activation temperature in the code. The required dimensional units for these quantities are,

$$[A_f(k)] or [A_b(k)] = \left[ \frac{\text{kmols}^1 \text{sec}^{-1}}{\text{m}^3} \right]$$ (80)

where,
\[ m_k = \sum_{i=1}^{N} v'_{ik} \quad \text{or} \quad \sum_{i=1}^{N} v''_{ik} \] (81)

respectively, and

\[
\begin{bmatrix} E_f \\ R_u \end{bmatrix} \quad \begin{bmatrix} E_b \\ R_u \end{bmatrix} = [K] \] (82)

The ALLSPD-3D finite chemistry modules have been developed with a generalized chemistry formulation. Therefore, any finite rate and global chemistry models that can be fitted into the Arrhenius form shown in Equation (25) can be used with the code. The method of inserting various chemical reaction mechanisms is shown in the Input/Output Files section along with several examples.

### 6.3 Coordinate System

The governing equations in the code are transformed from the original Cartesian components to generalized coordinates. Either right-hand or left-hand rule based coordinates are acceptable for an ALLSPD-3D calculation, although the right-hand rule is the standard convention and most commercial grid generation tools will not allow the use of left-handed grid system. Figure 13 shows two typical examples of acceptable grid systems. Here, physical (x-y-z) coordinate system, the block numbers and transformed coordinates (ξ-η-ζ) are illustrated as a reference. Although there is no particular restriction on the primary flow direction in the ALLSPD-3D code, the ξ direction is preferred.
FIGURE 13. Coordinates

FLOW INTO A CIRCULAR PIPE

FLOW OVER A WEDGE FLAME HOLDER
6.4 ALLSPD-3D For 2-D Problems

The current ALLSPD-3D code can be used to compute two-dimensional or axisymmetric flows. The optimal method for calculating 2-D or axisymmetric...
flows is to use the periodic boundary conditions, which require a minimum of 2 grid points in the periodic direction. In the current version of the code, it is suggested that the periodic boundary condition be used in the \( j \) or \( \eta \) direction, if possible. (See the I/O section for details on defining periodic boundaries.) Users must generate either two parallel grid points for a 2-D planar case or 2 grid points with an angle for an axisymmetric case (see Figure 14).

### 6.5 ALLSPD-3D Code Compilation

The code is designed to run on different computer platforms in either serial or parallel modes. The compilation for different machines is controlled by the `Makefile` using the proper `make` commands, which are described in this section. But before doing that, users must define some parameters required to define the problem size. The parameters in the `allspd.h`, which resides in the subdirectory `CFD/src/Include`, are used to define the problem size. These parameters are described as follows.

- **I_MAX**: The maximum number of grid points in the \( i \) direction among all blocks.
- **J_MAX**: The maximum number of grid points in the \( j \) direction among all blocks.
- **K_MAX**: The maximum number of grid points in the \( k \) direction among all blocks.
- **IJK_TOTAL**: \( \sum_{n=1}^{nbk} (i_{max} + 2) \times (j_{max} + 2) \times (k_{max} + 2) \) where \( i_{max}, j_{max}, \) and \( k_{max} \) are the grid points of the \( nth \) block in \( i, j, k \) respectively.
- **NUM_ZONES**: Number of blocks defined in the domain.
- **NUM_SPECIES**: Number of species.
- **NUM_REACT**: Number of reaction equations.
- **MAX_OBSTACLES**: Maximum number of internal blockages.
- **MAX_DILUTION**: Maximum number of dilution holes.
MAX_TRANSPIR: Maximum number of transpiration holes.

MAX_SPARKS: Maximum number of ignition sparks.

MAX_SPRAY_PARTICLES: Maximum number of spray particles.

MAX_SPRAY_INTEGRATION: Maximum number of integration time steps for the spray particles.

The make command is utilized for easy compilation on a variety of computer platforms. But before compiling for a new test case on any machine, run make veryclean to get rid of any old files leftover from previous compilations. The make command targets for both serial and parallel versions follow.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Serial Code</th>
<th>Parallel Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cray</td>
<td>make cray</td>
<td>N/A</td>
</tr>
<tr>
<td>HP-UX</td>
<td>make hppa</td>
<td>make p_hppa</td>
</tr>
<tr>
<td>IBM RS/6000</td>
<td>make rs6k</td>
<td>make p_rs6k</td>
</tr>
<tr>
<td>SGI/IRIX5</td>
<td>make sgi5</td>
<td>make p_sgi5</td>
</tr>
<tr>
<td>SUN/SunOS4.1</td>
<td>make sun4</td>
<td>make p_sun4</td>
</tr>
<tr>
<td>SUN/Solaris2</td>
<td>make sun4sol2</td>
<td>make p_sun4sol2</td>
</tr>
</tbody>
</table>

### 6.6 Program Execution (sans GUI)

After compiling the code, you can use the make command to install the executables allspd3d and pregrid wherever you intend to run your test case. This is done quite simply by utilizing the DEST definition in the Makefile with the install target. For example, if you would like to install the ALLSPD-3D executables in the "./Tests/Swirl-combustion" directory (relative to where you built the code), you would run make DEST=./Tests/Swirl-combustion install after a successful compilation.

Once the executables are installed, you need to create the various input files...
needed for your test case. At a minimum, the allspd.dat, grid.dat, gridlink.dat, and prop.dat files (or their equivalents named according to your preferences) must exist in the test directory. Other input files may be needed based upon the details of your particular test case. (See the File I/O section for details on file descriptions, formats, etc.) After you have all the needed files in place, you can run the code.

If your test case utilizes the turbulence model, you must first run pregrid which will pre-process the grid and generate a file called ynbl.dat. The ynbl.dat file contains normal distances (yn) to the nearest wall for each point in the grid. It also contains a crude estimate of the largest dimension of the grid (xlmax) for initialization of the turbulence quantities. The options for pregrid are listed below with the default file names which will be used in their absence.

<table>
<thead>
<tr>
<th>option</th>
<th>default file name</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-O</td>
<td>out-grid.dat</td>
<td>-Oout-grid.dat</td>
</tr>
<tr>
<td>-f</td>
<td>gridlink.dat</td>
<td>-gridlink.dat</td>
</tr>
<tr>
<td>-g</td>
<td>grid.dat</td>
<td>-ggrid.dat</td>
</tr>
<tr>
<td>-y</td>
<td>ynbl.dat</td>
<td>-ynbl.dat</td>
</tr>
</tbody>
</table>

During the execution of pregrid, various bits of information will appear on the standard output to inform you of the pre-processor's progress as it reads and processes grid.dat. When pregrid is finished, the out-grid.dat file will contain a synopsis of the data found in gridlink.dat for your verification. This pre-processing step is needed only once unless you change the grid. If you decide to restart a test case, there is no need to re-run pregrid.

Now you are ready to run the flow solver. In the simplest case, all you have to do is run allspd3d. The various options for using alternate file names are listed in the following table. (Again, see the File I/O section for details on file formats
and descriptions.)

**TABLE 11. The allspd3d Command Line Options**

<table>
<thead>
<tr>
<th>option</th>
<th>default file name</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>-i</td>
<td>allspd.dat</td>
<td>-allspd.dat</td>
</tr>
<tr>
<td>-o</td>
<td>out.dat</td>
<td>-oout.dat</td>
</tr>
<tr>
<td>-r</td>
<td>res.dat</td>
<td>-res.dat</td>
</tr>
<tr>
<td>-k</td>
<td>reske.dat</td>
<td>-kreske.dat</td>
</tr>
<tr>
<td>-p</td>
<td>allspd</td>
<td>-pallspd</td>
</tr>
<tr>
<td>-s</td>
<td>save.dat</td>
<td>-ssave.dat</td>
</tr>
<tr>
<td>-F</td>
<td>face.dat</td>
<td>-Fface.dat</td>
</tr>
<tr>
<td>-g</td>
<td>grid.dat</td>
<td>-ggrid.dat</td>
</tr>
<tr>
<td>-f</td>
<td>gridlink.dat</td>
<td>-gridlink.dat</td>
</tr>
<tr>
<td>-u</td>
<td>uvw.dat</td>
<td>-uvw.dat</td>
</tr>
<tr>
<td>-l</td>
<td>spray.dat</td>
<td>-spray.dat</td>
</tr>
<tr>
<td>-c</td>
<td>prop.dat</td>
<td>-prop.dat</td>
</tr>
<tr>
<td>-y</td>
<td>yml.dat</td>
<td>-yym.dat</td>
</tr>
</tbody>
</table>

As allspd3d starts up, it reads the various input files and reports a synopsis to out.dat and standard output. As the calculation progresses, numerous messages will appear on the standard output and in various output files. When the code finishes, your directory will contain output files and (if selected) restart files.

To continue the calculation, you need to edit allspd.dat accordingly and execute allspd3d again. The code will then read in the files save.dat and face.dat, in addition to all the other inputs. This will include the yml.dat file previously generated by pregrid if a turbulent flow is being simulated. You should see the code report a successful restart message after reading all the necessary files. After this message, the code executes as normal.
6.7 Parallel Implementation

6.7.1 Theory

The theory behind the parallel processing version of ALLSPD-3D is quite simple: the code is inherently divided in the data domain, therefore domain decomposition is used. (How's that for simplicity?) The multiple grid zone feature provides natural dividing lines in the data for decomposing the problem onto multiple processes, i.e., each grid zone is a natural candidate for parallel processing. Choosing which data to send between processes is also quite simple: essentially whatever variables are copied out of the 1-D arrays and into the 3-D arrays during CALL CONV13 in the serial code. This data is transferred between processes using a message-passing library instead of shared memory to maximize portability.

The transferred data supplies exactly the information a process needs from its neighboring zones without requiring memory space for all of the data in other zones. Because each process needs data for only its own grid zone (including those ghost cells which actually belong to neighboring zones), each process only needs enough memory for the largest zone. (ALLSPD-3D sizes all of its 3-D data arrays based on the largest zone in the problem.) This reduced memory feature of parallel processing can be very beneficial with large problem sizes. Also, since each process only calculates data on its zone, the time needed to calculate a single iteration is reduced to (more or less) the time needed for the most numerically intensive zone. The only cost for these great benefits of parallel processing is the time it takes to transfer data between neighbors.

6.7.2 Implementation

The PVM (Parallel Virtual Machine) message-passing library was chosen because of its wide acceptance, installed user base, and portability. PVM is used in a wide variety of applications on numerous architectures and has become a de-facto standard for message-passing libraries. The PVM library has many fea-
tures including spawning of processes on a virtual machine and the communication of various message types between architectures which may have inherently different data structures. These features are used in the parallel version of ALLSPD-3D. (For more information on PVM, visit http://www.epm.ornl.gov/pvm/.)

When the parallel version of ALLSPD-3D is executed, it automatically spawns a child process for each grid zone except the one zone that the parent runs. The parent process then becomes the controlling node in the parallel calculation and collects standard output from all the children for display in the controller's output. (This can produce some ugly output, but it is still the best way to gather all the information.) Each process then runs as a single zone calculation with needed data passed between neighbors across the zone interfaces.

The data that is transferred between grid zones (and therefore processes) includes both geometric data and flow data. The geometric data is passed between neighbors twice at startup: first to transfer X, Y, Z coordinate information about neighboring grids, then to update the various derived quantities. This is necessary since each process reads in the entire grid, but only stores the data for its own zone. Then the flow information (pressure, velocities, enthalpy, species concentrations, viscosity, etc.) is transferred between neighbors after initialization and after each iteration. To minimize the size of each message, the data is transferred in individual messages, e.g., one message for X values, another for U velocity values, etc. (All these messages are sent using PVM's default data format which allows a heterogeneous mix of machines to transparently share data without concern for data representation details on each architecture.) All this message passing fills in the data at the ghost cells for each zonal interface.

Sending data to neighbors after each iteration impacts the behavior of the code both superficially and significantly. The transfer of data obviates any need to move data in and out of the 3-D arrays. It also forces the ghost cells to contain data which is lagged a full iteration, i.e., the data one zone uses from a neighbor is old information. In the serial version of ALLSPD-3D, the zone interfaces can
use data updated during the current iteration depending upon the zone ordering. (This fully-lagged boundary condition at the zone interfaces is not necessarily bad. Some test cases have actually had improved convergence in parallel processing, presumably from the dampening of errors across lagged boundaries.) This is a fundamental change in the propagation of information through the flow field and results in differing convergence histories for the parallel and serial versions of ALLSPD-3D.

By sending and receiving data between neighbors and then synchronizing with only neighboring zones, a loose synchronization of the total calculation is achieved. Basically, the process that takes the longest to calculate its zone will force all the others to wait for it because of the interdependence of the grid zones. This also means that load balancing will be best achieved by dividing the grid into more or less equally sized zones. (The only caveat here is that the spray model requires additional resources and therefore fewer grid points should be used in the zone containing spray calculation to achieve load balance.)

6.7.3 Notes About Coding Details

The ALLSPD-3D code is primarily written in Fortran, but much of the parallel coding is in C. This results in the sharing of some data between C and Fortran which is not pretty. Fortunately, the basic data transfer of geometric and flow data is handled by Fortran code (to avoid the perverse practice of accessing Fortran COMMON BLOCK data from C) and the parallel management is performed by C code.

If you decide to modify the basic sequencing of sends, receives, and synchronizations, be very careful. The checking for abort signals is critical to avoid deadlocking the calculation because one node has decided to exit while another is waiting for some message that will never arrive.

The amount of data transferred between neighbors is based on the largest possible size of a face because of limitations in Fortran. This means that messages could be greatly oversized for particular problems, especially 2-D calcu-
lations. If you wish to reduce the data passed between zones to a more reasonable size for a specific test case, you can change the definition of MAXFACE in the CFD/src/Include/allspd.h file. (But be very careful.) Also, the communication time for data transfer can be improved for particular test cases. For generality, the parallel version of ALLSPD-3D uses the default data transfer mode of PVM. If your virtual machine is homogeneous, you could switch to raw data mode or data in place mode. Just look for PVM_DATA_XFER_C and PVM_DATA_XFER_F in the CFD/src/Include/allspd.h file to change the data transfer mode.

Currently, ALLSPD-3D has no provision for spawning children to specific machines based upon CPU speed, grid zone size, etc. Therefore, the gridding will have a direct influence on the load balance of the overall calculation. Also, the PVM daemons will decide which host will calculate which zone, usually without regard to existing CPU loads. This means that you can easily wind up with processes spawned in the worst (or best) possible allocation, especially in highly heterogeneous virtual machines. Another impact from PVM is the connection used between machines. A non-shared communication medium (e.g., ATM, high-speed switches, etc.) is best. Standard ethernet is very common, but it is also the slowest, especially when all the processes need to communicate at the same time.

6.7.4 Compilation and Run-time Details

To compile and run the parallel processing version of ALLSPD-3D, you must have PVM 3.3 installed on all hosts that you intend to use. The Makefile uses the environment variable PVM_ROOT and PVM_ARCH to know where PVM include files and libraries are located, so make sure PVM is set up properly. Then compilation is just a matter of issuing the proper command. (I recommend that you always run make veryclean whenever starting a new test case or building for a different architecture.) Assuming you want to run ALLSPD-3D in parallel on a cluster of IBM RS/6000 workstations, you would run make
p_rs6k. The make command will then compile the various modules of ALLSPD-3D including those needed for parallel processing. The pre-processing of the source code will automatically include those sections of code needed for parallel processing and leave out sections not needed. If your cluster of workstations is heterogeneous, just build with the correct parallel target (e.g., make p_sgi for SGI machines) and install the resulting executables in separate directories according to the needs of PVM. (And don’t forget to run make veryclean between builds.)

After the executable(s) are built and installed in the proper locations, run pregrid as normal, then start PVM with a hostfile which is configured for your particular cluster and your problem. (This is basically a PVM issue and will not be covered here except to say that the ep [executable path] and wd [working directory] options in the hostfile are critical. Note that the GUI will handle these details for you.) After the pvm daemons are running on all the hosts, you start up ALLSPD-3D pretty much as normal. Be careful not to use pathnames on heterogeneous virtual machines since ALLSPD-3D will spawn processes on the virtual machine using the same command line you issue. After this, each process will read all the input files but only use the parts it needs, e.g., each process will read the entire grid but store only its zone. Whenever a process writes out an output file, it uses the name supplied on the command line (or the default name) with a grid zone extension, i.e., the process calculating zone 1 will output its turbulence residual data for its zone (and only its zone) to reske.dat-001 instead of reske.dat. A slight exception to this rule is the global residual file res.dat. Each process gets its own residual file for the flow equations on its zone. The controlling node, however, also outputs a residual for the entire flow field to res.dat after gathering information from all the processes and calculating a residual spanning the entire grid. After all the processes finish the calculation, you should probably halt the PVM daemons.

When the code finishes, you will have numerous files because of each process dumping out its own results. The solution files are single zone solutions in multi-zone Plot3D format, but can be easily assembled into a single file. The re-
start files (save.dat-nnn and face.dat-nnn) are inherently incompatible with the serial code. These files contain only the data needed for a single zone to restart the calculation while the serial code needs all the zones in a single file. Assembling the numerous restart files of a parallel calculation into a single file for use by the serial code is probably possible but likely to be a painful experience and is not recommended.
VII : Examples

This section describes two test cases computed using the ALLSPD-3D code. These cases are designed to help the users obtain the experience needed in setting up the ALLSPD-3D input data files. The ALLSPD-3D solutions are also included so that user-obtained solutions can be compared with the other computational results and also with the experimental data. Re-running these cases will help users gain experience needed to create the ALLSPD-3D inputs and is a highly recommended step for new users. The two test cases included in this section are:

(1) 3-D non-reacting laminar and turbulent transition duct flows
(2) Axisymmetric turbulent swirl-can combustor flow with spray

7.1 3-D Non-Reacting Transition Duct Flow

The fully turbulent and three-dimensional circular to rectangular transition duct problem of Davis\textsuperscript{41} was chosen as the first test case because of the extensive and detailed experimental data available. The grid has 52,521 cells [41x21x61] and three typical streamwise stations are shown in Figure 15. The
tests conditions are shown in Table 12.

<table>
<thead>
<tr>
<th>$Re_D$</th>
<th>195,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_{ref}$</td>
<td>298K</td>
</tr>
<tr>
<td>$U_{ref}$</td>
<td>29 m/s</td>
</tr>
</tbody>
</table>

TABLE 12. Reference Conditions for the Transition Duct

FIGURE 15. The Transition Duct Grid at Three Streamwise Stations

Before running the fully turbulent case, we investigated a laminar flow in the transition duct by reducing the inlet flow speed. The Reynolds number is about 195 for the laminar flow. The input data files, gridlink.dat, allspd.dat, and prop.dat, needed to set up this laminar flow are listed in Figure 16 to Figure 18. The velocity vectors obtained from the ALLSPD-3D solution are shown in Figure 19, and the convergence history is shown in Figure 20.

FIGURE 16. The gridlink.dat Input File for the Transition Duct
FIGURE 17. The allspd.dat Input File for the Laminar Transition Duct

FIGURE 18. The prop.dat Input file for the Transition Duct
FIGURE 19. Velocity Vectors for the Laminar Transition Duct

FIGURE 20. Convergence History for the Laminar and Turbulent Transition Duct

The fully turbulent transition duct flow case requires the input data file allspd.dat listed in Figure 17. The gridlink.dat and prop.dat are identical to those for the previous laminar case. For the turbulent flow, it is noted that iuvwt is equal to 1 and therefore the uuvwt.dat, an initial condition file, has to be provided as a separate input file. Figure 22 shows the velocity vectors on the two symmetry planes of the transition duct. Comparisons of the u velocities at various streamwise stations along the two major axis with experimental data are shown in Figure 23. This figure shows that reasonable agreement with experi-
mental data can be obtained with the ALLSPD-3D code. The streamwise locations of the velocity profile comparisons shown in Figure 23 are illustrated in Figure 22. Here, additional velocity vectors on the symmetry planes are shown. Figure 20 also shows the convergence history for this turbulent flow ALLSPD-3D calculation.

```
0 2000
100
200
1 0 1
0
1
5.000000e-01 1.000000e+00
1.000000e+10 1.000000e+10 1.000000e+10
1.000000e+00 1.000000e+00 1.000000e+00
1
1.000000e-01
2e-13
1
1
2.995000e+01 0.000000e+00 0.000000e+00
2.970000e+02
1.013250e+05
1.021400e+01
0
0
30000
50000
0.
1.600000e+03
0.000000e+00
0
0
1 2 3 4 5 11 12 13 7 9 0 0 0 0
2
'02_T<1000' 2.335700e-01
'N2_T<1000' 7.664300e-01
```

**FIGURE 21. The allspd.dat Input File for the Turbulent Transition Duct**

![Figure 21](image)

**FIGURE 22. Velocity Vectors for the Turbulent Transition Duct**

![Figure 22](image)
7.2 Axisymmetric Swirl-Can Combustor Flow with Spray

The second sample test case was used to validate the current turbulence model and the spray model for swirling flow. The experiment was conducted by Roback and Johnson \(^{42}\) for a non-reacting turbulent flow. However, we use this case to further demonstrate the ALLSPD-3D spray model by initiating combustion in the flow field. Since this is an axisymmetric flow (see Section 6.4 of the ALLSPD-3D manual), only two grid points in the circumferential direction and...
the periodic boundary condition are used. Here, the combustion is initiated by injecting the liquid methanol fuel into the flow. The input data files needed are listed in Figure 24 to Figure 26. It should be noted that iuvw is equal to 1 and therefore the uvwt.dat input file is also needed. The reference conditions are provided in Table 13.

<table>
<thead>
<tr>
<th>ReD</th>
<th>61,180</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{ref}</td>
<td>300 K</td>
</tr>
<tr>
<td>U_{ref}</td>
<td>16 m/s</td>
</tr>
</tbody>
</table>

**TABLE 13. Reference Conditions for the Swirl Can Combustor**

```
0
6
49 2 38
15 2 38
19 2 38
49 2 24
15 2 24
19 2 24
4 1 6 6 4 7 0 0 0 0 0 0
4 4 6 4 4 7 0 0 0 0 0 0
2 4 6 6 4 7 1 0 1 3 2 5 0 0 0 0 0 0
4 1 6 6 3 4 0 0 0 0 0 0 0
4 4 6 6 3 4 0 0 0 0 0 0 0
2 4 6 6 3 4 1 0 1 3 2 5 0 0 0 0 0 0
2 0 0 0 4 0
3 1 0 0 5 0
0 2 0 0 6 0
5 0 0 0 1
6 4 0 0 0 2
0 5 0 0 0 3
```

**FIGURE 24. The gridlink.dat Input File for the Swirl Can Combustor**
0
2000
100
100
1 0 1
1
1
1.000000e-01 0.000000e+00
1.000000e+10 1.000000e+10 1.000000e+10
1.050000e-00 1.050000e+00 1.050000e+00
1
1.000000e-01
1e-13
1
1
1
1.600000e+01 0.000000e+00 0.000000e+00
3.000000e+02
1.013250e+05
6.100000e-02
0
0
300
50000
1
1 25 40 1 2 2 30
1.600000e+03
5.000000e+07
0
1
1
30 30 1 2 5 24
1 0.000000e+00 0.000000e+00
1.000000e+01 3.700000e+01
300
20
0
3.000000e+02
1.907808e-04
3.000000e+01
1 2 3 4 5 15 16 17 18 19 11 12 13 14 16
5
‘CH40’ 0.000000e+00
‘O2’ 2.335700e-01
‘CO2’ 0.000000e+00
‘H2O’ 0.000000e+00
‘N2’ 7.664300e-01

FIGURE 25. The allspd.dat Input File for the Swirl Can Combustor
Figure 26. The prop.dat Input File for the Swirl Can Combustor

The grid used by the ALLSPD-3D code on the swirl can combustor is shown in Figure 28. In this figure, every other point is plotted with each zone in separate colors for clarity. The grid is comprised of six blocks as detailed in Table 14. The velocity vectors obtained from the ALLSPD-3D solution at 2000 iterations is shown in Figure 29. This solution was obtained using the ALLSPD-3D code in the parallel mode where the calculation was performed on 6 workstation nodes. The velocity vectors are colored according to flow field temperature to show that the fuel has ignited. This ignition produces a flowfield where the temperature ranges from 300 K to 2100 K. In Figure 29, the color blue is used to indicate low temperature and the color red is used to indicate high temperature. The convergence history for this calculation is shown in Figure 27.
<table>
<thead>
<tr>
<th>Block</th>
<th>Block Dimensions</th>
<th>Number of Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>49 x 2 x 38</td>
<td>3724</td>
</tr>
<tr>
<td>2</td>
<td>15 x 2 x 38</td>
<td>1140</td>
</tr>
<tr>
<td>3</td>
<td>19 x 2 x 38</td>
<td>1444</td>
</tr>
<tr>
<td>4</td>
<td>49 x 2 x 24</td>
<td>2352</td>
</tr>
<tr>
<td>5</td>
<td>15 x 2 x 24</td>
<td>720</td>
</tr>
<tr>
<td>6</td>
<td>19 x 2 x 24</td>
<td>912</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>10292</td>
</tr>
</tbody>
</table>

TABLE 14. Swirl Can Combustor Grid Block Sizes

![Swirl Can - 2D](chart.png)

FIGURE 27. Convergence History for the Swirl Can Combustor
FIGURE 28. Grid for the Swirl Can Combustor

FIGURE 29. Temperature Colored Velocity Vectors
Acknowledgments

The authors would like to express deep gratitude to Dr. Robert M. Stubbs, Chief of the Computational Methods for Space Branch, and Dr. Edward J. Mularz, former Chief of the Aerothermochemistry Branch, at NASA Lewis Research Center for their coordination and guidance on the ALLSPD-3D code development. Many thanks are also due to Drs. Peter Sockol, John Deur, and Jeffrey Moder for numerous technical discussions to improve the efficiency of the code. Also, we would like to thank Bonnie McBride for helping set up the thermophysical and transport properties of the species database. Finally, special thanks to many of our colleagues at NASA Lewis, including Drs. Shaye Yungster, Suri Raju, Sheng-Tao Yu, Jian-Shun Shuen, and Nan-Suey Liu amongst others, for assistance in the early development of the code.
References


10 Rabinowitz, M. Personal communications, NASA Lewis Research Center, Cleveland, Ohio, 1994.


Appendix A

Jacobian Matrices

The number of species equations is assumed to be five (N=5) in the Jacobian matrices given below

\[ A^{(1)} = \begin{bmatrix}
\frac{U}{RT} & \rho \alpha_1 & \rho \alpha_2 & \rho \alpha_3 & \frac{\rho U}{C_p T} \\
\alpha_1 + \frac{u U}{RT} & \rho (U + \alpha_1 u) & \rho u \alpha_2 & \rho u \alpha_3 & \frac{\rho u U}{C_p T} \\
\alpha_2 + \frac{v U}{RT} & \rho v \alpha_1 & \rho (U + \alpha_2 v) & \rho v \alpha_3 & \frac{\rho v U}{C_p T} \\
\alpha_3 + \frac{w U}{RT} & \rho w \alpha_1 & \rho w \alpha_2 & \rho (U + \alpha_3 w) & \frac{\rho w U}{C_p T} \\
\frac{U H}{RT} & \rho (u U + \alpha_1 H) & \rho (v U + \alpha_2 H) & \rho (w U + \alpha_3 H) & \rho U \phi
\end{bmatrix} \quad (A\ 1) \]

\[ A^{(2)} = \begin{bmatrix}
\rho U & 0 \\
0 & \rho U
\end{bmatrix} \quad (A\ 2) \]

\[ A^{(3)} = \begin{bmatrix}
\rho U \Theta_1 & \rho U Y_1 \Omega_2 & \rho U Y_1 \Omega_3 & \rho U Y_1 \Omega_4 \\
\rho U Y_2 \Omega_1 & \rho U \Theta_2 & \rho U Y_2 \Omega_3 & \rho U Y_2 \Omega_4 \\
\rho U Y_3 \Omega_1 & \rho U Y_3 \Omega_2 & \rho U \Theta_3 & \rho U Y_2 \Omega_4 \\
\rho U Y_4 \Omega_1 & \rho U Y_4 \Omega_2 & \rho U Y_4 \Omega_3 & \rho U \Theta_4
\end{bmatrix} \quad (A\ 3) \]

\[ U = \alpha_1 u + \alpha_2 v + \alpha_3 w \quad (A\ 4) \]
\[ \alpha_1 = \xi_x \quad \alpha_2 = \xi_y \quad \alpha_3 = \xi_z \quad \text{(A 5)} \]

\[ \Omega_i = -W \left( \frac{1}{W_i} - \frac{1}{W_{\text{ref}}} \right) + \frac{(h_i - h_{\text{ref}})}{C_p T} \quad \text{(A 6)} \]

\[ \Theta_i = 1 + Y_i \Omega_i \quad \phi = 1 - \frac{H}{C_p T} \quad \text{(A 7)} \]

Here \( W \) is the molecular weight of the gas mixture and \( C_p \) is the mixture specific heat \( C_{p_m} \). The Jacobian matrix \( B^{(m)} \) is obtained by letting \( \alpha_1 = \eta_x, \alpha_2 = \eta_y, \alpha_3 = \eta_z \) and Jacobian matrix \( C^{(m)} \) is obtained by letting \( \alpha_1 = \zeta_x, \alpha_2 = \zeta_y, \alpha_3 = \zeta_z \).

The viscous term Jacobian has the following form

\[
R_{\xi x}^{(1)} = \frac{1}{f} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & \mu_e \left( \frac{1}{3} \alpha_1^2 - \Lambda \right) & \mu_e \alpha_{12} & \mu_e \alpha_{13} & 0 \\
0 & \mu_e \alpha_{12} & \mu_e \left( \frac{1}{3} \alpha_2^2 - \Lambda \right) & \mu_e \alpha_{23} & 0 \\
0 & \mu_e \alpha_{13} & \mu_e \alpha_{23} & \mu_e \left( \frac{1}{3} \alpha_3^2 - \Lambda \right) & 0 \\
0 & \pi_u & \pi_v & \pi_w & \frac{k_e \Lambda}{C_p}
\end{bmatrix}
\]

\[
R_{\xi x}^{(2)} = \frac{1}{f} \begin{bmatrix}
\left( \mu_l + \frac{\mu_l}{\sigma_e} \right) \Lambda & 0 \\
0 & \left( \mu_l + \frac{\mu_l}{\sigma_e} \right) \Lambda
\end{bmatrix}
\]

\[ \text{(A 8)} \]

\[ \text{(A 9)} \]
\[ R_{\xi \xi}^{(3)} = \frac{1}{j} \begin{bmatrix} \rho D_{1m} \Lambda & 0 & 0 & 0 \\ 0 & \rho D_{2m} \Lambda & 0 & 0 \\ 0 & 0 & \rho D_{3m} \Lambda & 0 \\ 0 & 0 & 0 & \rho D_{4m} \Lambda \end{bmatrix} \] (A 10)

\[ \alpha_{12} = \frac{1}{3} \alpha_1 \alpha_2 \quad \alpha_{13} = \frac{1}{3} \alpha_1 \alpha_3 \] (A 11)

\[ \alpha_{23} = \frac{1}{3} \alpha_2 \alpha_3 \quad \Lambda = \alpha_1^2 + \alpha_2^2 + \alpha_3^2 \] (A 12)

\[ \pi_u = \mu_e \left( u \left( \frac{1}{3} \alpha_1^2 - \Lambda \right) + v \alpha_{12} + w \alpha_{13} \right) \]

\[ \pi_v = \mu_e \left( v \left( \frac{1}{3} \alpha_1^2 - \Lambda \right) + u \alpha_{12} + w \alpha_{23} \right) \]

\[ \pi_w = \mu_e \left( w \left( \frac{1}{3} \alpha_3^2 - \Lambda \right) + v \alpha_{13} + w \alpha_{23} \right) \]

\[ T^{(3)} = \begin{bmatrix} \frac{\partial \omega_1}{\partial Y_1} & \frac{\partial \omega_1}{\partial Y_2} & \frac{\partial \omega_1}{\partial Y_3} & \frac{\partial \omega_1}{\partial Y_4} \\ \frac{\partial \omega_2}{\partial Y_1} & \frac{\partial \omega_2}{\partial Y_2} & \frac{\partial \omega_2}{\partial Y_3} & \frac{\partial \omega_2}{\partial Y_4} \\ \frac{\partial \omega_3}{\partial Y_1} & \frac{\partial \omega_3}{\partial Y_2} & \frac{\partial \omega_3}{\partial Y_3} & \frac{\partial \omega_3}{\partial Y_4} \\ \frac{\partial \omega_4}{\partial Y_1} & \frac{\partial \omega_4}{\partial Y_2} & \frac{\partial \omega_4}{\partial Y_3} & \frac{\partial \omega_4}{\partial Y_4} \end{bmatrix} \quad T^{(2)} = \begin{bmatrix} \frac{\partial S_1}{\partial \xi} & \frac{\partial S_1}{\partial \epsilon} \\ \frac{\partial S_2}{\partial \xi} & \frac{\partial S_2}{\partial \epsilon} \\ \frac{\partial S_3}{\partial \xi} & \frac{\partial S_3}{\partial \epsilon} \\ \frac{\partial S_4}{\partial \xi} & \frac{\partial S_4}{\partial \epsilon} \end{bmatrix} \] (A 13)

In order to reduce the stiffness of the chemical species source term, the source terms are treated implicitly. The chemical species source, Equation(24), is a function of temperature and species concentration and therefore a chain rule is required to evaluate the source term in general. The linearization of the left hand side leads to the following expressions:
\[
\frac{\partial \omega_i}{\partial Y_j} = -\rho \left[ MW \left( \frac{1}{MW_j} - \frac{1}{MW_N} \right) - \frac{(h_j - h_N)}{C_p T} \right] \frac{\partial \omega_i}{\partial \rho} \\
+ MW \sum_{k=1}^{N} \left( CR_{ik} \left( K_j \frac{v^i_j}{Y_j} \right) \prod_{l=1}^{N} C_i^{\nu_{ia}} \right) - \frac{(h_j - h_N)}{C_p} \frac{\partial \omega_i}{\partial T}
\]

\[
\frac{\partial \omega_i}{\partial \rho} = MW \sum_{k=1}^{N} \left( CR_{ik} \left( \frac{K_i}{\rho} \left( \sum_{l=1}^{N} v^i_l \right) \prod_{l=1}^{N} C_i^{\nu_{ia}} - \frac{K_i}{\rho} \left( \sum_{l=1}^{N} v^i_l \right) \prod_{l=1}^{N} C_i^{\nu_{ia}} \right) \right)
\]

\[
\frac{\partial \omega_i}{\partial T} = MW \sum_{k=1}^{N} \left( CR_{ik} \left( \frac{K_i}{T} \left( \alpha_j (k) - \frac{E_{fb}}{R_a T} \right) \prod_{l=1}^{N} C_i^{\nu_{ia}} - \frac{K_i}{T} \left( \alpha_j (k) - \frac{E_{fb}}{R_a T} \right) \prod_{l=1}^{N} C_i^{\nu_{ia}} \right) \right)
\]

(A 14)

where, \(E_{fb}\) and \(K_{fb}\) are function of the \(k^{th}\) index.

The turbulence source term shown in Equation(35) can be very stiff numerically. Therefore, the ALLSPD-3D code solves the equation in a point implicit manner. The linearization of the turbulent source term required the use of the following source Jacobian term. This formulation has been optimized for convergence. This form has been tested for a number of flow situations and model formulations

\[
T_{ij} = \begin{bmatrix}
\left( \frac{p_k}{\rho_k} - \frac{e}{\rho_k} \right) & 0 \\
0 & C_{\varepsilon 1} \left( \frac{\varepsilon}{\rho \varepsilon} - 2f_{\varepsilon} \frac{C_{\varepsilon 2}}{C_{\varepsilon 1}} \right)
\end{bmatrix}
\]

(A 15)

\[
\dot{p}_k = -\frac{2}{3} \mu \left( \frac{\partial u_k}{\partial x_k} \right) \left( \frac{\partial u_k}{\partial x_k} \right) + \text{MIN} \left( 0, -\frac{2}{3} \mu \rho k^2 \delta_{ij} \frac{\partial u_i}{\partial x_j} \right)
\]

(A 16)
Appendix B

Thermodynamic Database Information

The database used by the GUI is an assembly of gaseous and liquid properties for several fuels and oxidizers. Most of the thermodynamic and transport properties were fourth order polynomial curve fits to those provided by McBride et al. The original data from this report are split into two temperature ranges. These curves have been combined into a single fit for a temperature range 300K to 3000K. The diffusion constants used in the Chapman-Enskog formulation for species diffusion were taken from Svehla. The liquid properties came from Reid. An example of a database entry is shown in Table 15 and described in Table 16 (Superscripts are used to relate Table 15 to the appropriate descriptions in Table 16).

<table>
<thead>
<tr>
<th>z^A</th>
<th>CH4^B</th>
<th>Methanol^C</th>
<th>32.042^D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1^E</td>
<td>1^F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.85381449E-01^G</td>
<td>1.68622442E-02</td>
<td>-9.24980079E-06</td>
<td>2.62973642E-09</td>
</tr>
<tr>
<td>-2.23317873E-06^H</td>
<td>4.24359048E-08</td>
<td>-1.09136602E-11</td>
<td>2.16723754E-15</td>
</tr>
<tr>
<td>-1.79329923E-02^I</td>
<td>9.74961128E-05</td>
<td>6.49563168E-08</td>
<td>-3.10853977E-11</td>
</tr>
<tr>
<td>3.626^j</td>
<td>481.8^k</td>
<td>-2.50864348E+04^L</td>
<td>1.97745991E+01^M</td>
</tr>
<tr>
<td>79.565^n</td>
<td>513.0^o</td>
<td>337.71^p</td>
<td>175.0^q</td>
</tr>
<tr>
<td>1178.623^r</td>
<td>2.49e3^s</td>
<td>0.2^t</td>
<td>791.5^u</td>
</tr>
<tr>
<td>-7.28936^v</td>
<td>1.53679</td>
<td>-3.08367</td>
<td>-1.02456</td>
</tr>
</tbody>
</table>

TABLE 15. Sample GUI Database Entry

141
<table>
<thead>
<tr>
<th>Entry Name</th>
<th>Entry Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a Entry Number</td>
<td>Database book keeping</td>
</tr>
<tr>
<td>b Chemical Name</td>
<td>The chemical name of the species in upper case letters. The GUI is case sensitive. However, a species such as iso-butane would be IC3H8</td>
</tr>
<tr>
<td>c Common Name</td>
<td>A more general specie name</td>
</tr>
<tr>
<td>d Molecular Weight</td>
<td>The species molecular weight</td>
</tr>
<tr>
<td>e Gas Phase Flag</td>
<td>Indicates to the GUI that a gas phase exists in the database</td>
</tr>
<tr>
<td>f Liquid Phase Flag</td>
<td>Indicates to the GUI that a liquid phase exists in the database</td>
</tr>
<tr>
<td>g Specific Heat Coefficients</td>
<td>The coefficients used in the fourth order polynomial fit for specific heat</td>
</tr>
<tr>
<td>h Viscosity Coefficients</td>
<td>The coefficients used in the fourth order polynomial fit for viscosity</td>
</tr>
<tr>
<td>j Thermal Conductivity Coefficients</td>
<td>The coefficients used in the fourth order polynomial fit for thermal conductivity</td>
</tr>
<tr>
<td>k $\sigma$</td>
<td>Collision cross section for diffusion</td>
</tr>
<tr>
<td>l $T_c$</td>
<td>Temperature for collision integral correction</td>
</tr>
<tr>
<td>m Enthalpy Constant</td>
<td>Integration constant for enthalpy</td>
</tr>
<tr>
<td>n Entropy Constant</td>
<td>Integration constant for entropy</td>
</tr>
<tr>
<td>o Critical Pressure</td>
<td>Critical pressure for species</td>
</tr>
<tr>
<td>p Critical Temperature</td>
<td>Critical temperature for species</td>
</tr>
<tr>
<td>q Boiling Temperature</td>
<td>Boiling temperature for species</td>
</tr>
<tr>
<td>r Melting Temperature</td>
<td>Melting temperature for species</td>
</tr>
<tr>
<td>s Latent Heat</td>
<td>Latent heat of vaporization at the spray temperature (293K)</td>
</tr>
<tr>
<td>t Liquid Specific Heat</td>
<td>Specific heat of the liquid at the spray temperature (293K)</td>
</tr>
<tr>
<td>u Liquid Thermal Conductivity</td>
<td>Thermal conductivity of the liquid at the spray temperature (293K)</td>
</tr>
<tr>
<td>v Liquid Density</td>
<td>Density of the liquid at the spray temperature (293K)</td>
</tr>
<tr>
<td>w Vapor Pressure Coefficients</td>
<td>Coefficients for the polynomial fit to fuel vapor pressure.</td>
</tr>
</tbody>
</table>
### ALLSPD3D V1.0 - I/O GUIDE

**BC**: Boundary Condition  
**IC**: Initial Condition  
**EQ**: Equations  
**O**: Dimensional Units or Index Count

#### allspd.h - Parameter File

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L_MAX, L_MAX, K_MAX</td>
<td>Maximum Grid Cells in x-y-z direction</td>
</tr>
<tr>
<td>UJK_TOTAL</td>
<td>Total Number of Cells</td>
</tr>
<tr>
<td>NUM_ZONES</td>
<td>Total Number of Zones</td>
</tr>
<tr>
<td>NUM_SPECIES</td>
<td>Total Number of Species</td>
</tr>
<tr>
<td>NUM_REACT</td>
<td>Total Number of Reactions</td>
</tr>
<tr>
<td>MAX_OBSTACLES</td>
<td>Total Number of Obstacles</td>
</tr>
<tr>
<td>MAX_DILUTION</td>
<td>Total Number of Dilation Locations(&gt;1)</td>
</tr>
<tr>
<td>MAX_TRANSPIR</td>
<td>Total Number of Transpiration Locations(&gt;1)</td>
</tr>
<tr>
<td>MAX_SPARKS</td>
<td>Total Number of Spark Ignition Location(&gt;1)</td>
</tr>
<tr>
<td>MAX_SPRAY_PARTICLES</td>
<td>Maximum Number of Spray Particles</td>
</tr>
<tr>
<td>MAX_SPRAY_INTEGRATION</td>
<td>Maximum Number of Integration steps(&gt;1)</td>
</tr>
<tr>
<td>NUM_FLOW_EQNS</td>
<td>=5</td>
</tr>
<tr>
<td>NUM_TURB_EQNS</td>
<td>=2 (fixed)</td>
</tr>
<tr>
<td>SCHMITD_NUM</td>
<td>= 0.9</td>
</tr>
<tr>
<td>PRANDTL_NUM</td>
<td>= 0.7</td>
</tr>
<tr>
<td>TURB_CONST_K</td>
<td>= 1.0</td>
</tr>
<tr>
<td>TURB_CONST_EPS</td>
<td>= 1.3</td>
</tr>
<tr>
<td>MAXARGS</td>
<td>= 31 (fixed)</td>
</tr>
</tbody>
</table>

#### allspd.dat - Input File

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>FREAD</td>
<td>= 0 = a fresh run, = 1 = a restart run</td>
</tr>
<tr>
<td>NITMAX</td>
<td>Total number of iterations</td>
</tr>
<tr>
<td>NUPLOT</td>
<td>Increment at which Plot3d files are generated</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASAVE</td>
<td>Increment at which the restart file is generated</td>
</tr>
<tr>
<td>ITURB, ICMS,</td>
<td>ITURB, = 0 laminar flow, = 1 turbulent flow, ICMS, = 0 constant C_p(=0.09), = 1 variable C_p</td>
</tr>
<tr>
<td>IDISBC</td>
<td>IDISBC, = 0 Dirichlet e-wall BC, = 1 Neumann e wall BC</td>
</tr>
<tr>
<td>ICOMB</td>
<td>ICOMB = 0 non-reacting flow, = 1 reacting flow</td>
</tr>
<tr>
<td>IARTV</td>
<td>IARTV = 0 no artificial damping, = 1 with artificial damping</td>
</tr>
<tr>
<td>VLS4, VLS2</td>
<td>VLS4 = 4th order damping coefficient, VLS2 = 2nd order damping coefficient</td>
</tr>
<tr>
<td>CFL()</td>
<td>CFL number, CFL(1) =flow EQ, CFL(2) = k-e EQ, CFL(3) = species EQ</td>
</tr>
<tr>
<td>PREFAC()</td>
<td>LU scheme pre-conditioning factor, PREFAC(1) = flow EQ, PREFAC(2) = k-e EQ, PREFAC(3) = species EQ</td>
</tr>
<tr>
<td>NSWITCH</td>
<td>CFL ramping function switch. CFL number is incremented by CFL = CFL-factau from faclst to 1 over nswitch iterations.</td>
</tr>
<tr>
<td>FACNL</td>
<td>Initial CFL factor value(-0.1)</td>
</tr>
<tr>
<td>BOND</td>
<td>Convergence criterion[default 1e-13]</td>
</tr>
<tr>
<td>NKECALL</td>
<td>Turbulence EQ update increment</td>
</tr>
<tr>
<td>NSCALE</td>
<td>Species EQ update increment</td>
</tr>
<tr>
<td>LUVWT</td>
<td>= -1 only inlet BC specified by uin,vin, and win and zero IC = 0 uniform flow BC and IC = 1 user provides BC and IC using UUVWT, = 2 specified swirl BC in UUVWT file.</td>
</tr>
<tr>
<td>UIN, VIN, WIN</td>
<td>Initial velocities, uin, vin, win = x-y-z components of velocity (m/s)</td>
</tr>
<tr>
<td>TREF</td>
<td>Reference temperature (K)</td>
</tr>
<tr>
<td>TREF</td>
<td>Reference pressure (Pa)</td>
</tr>
<tr>
<td>RREF</td>
<td>Reference length scale (m)</td>
</tr>
<tr>
<td>IEHM</td>
<td>Eddy break-up model switch, = 0 off(default), = -1 on</td>
</tr>
<tr>
<td>IEQATE</td>
<td>Equilibrium Model switch, = 0 off, = 1 reverse reaction constant calculation, = -1 forward reaction constant calculation</td>
</tr>
<tr>
<td>IT11G</td>
<td>Starting point to add Ignition Energy</td>
</tr>
<tr>
<td>IT21G</td>
<td>End adding Ignition Energy</td>
</tr>
<tr>
<td>NOLGT</td>
<td>Number of ignition sources, (If = 0, skips to next input of TOFF)</td>
</tr>
<tr>
<td>THBG</td>
<td>Ignition spark position(s)</td>
</tr>
<tr>
<td>T2G, I...</td>
<td>nOth ignition is located in 1G1G(2) block.</td>
</tr>
<tr>
<td>N1G, N2G, K1G,</td>
<td>Starting I,j,k-index of the ignition spark location.</td>
</tr>
<tr>
<td>J2G, K2G</td>
<td>Ending I,j,k-index of the ignition spark location.</td>
</tr>
<tr>
<td>TOFF</td>
<td>Cut-off temperature at which the ignition is turned off</td>
</tr>
<tr>
<td>SPARKUSER</td>
<td>Strength of ignition spark (J/m^3-s)) = 0, (default strength used)</td>
</tr>
<tr>
<td>NSDATA</td>
<td>= 0, default spray data used, = 1, the spray data are to be provided by user-generated file, SPRAY.DAT.</td>
</tr>
<tr>
<td>ISPRAY</td>
<td>= 0 Single-phase flow calculation(skip to input of INLET(I), = 1 Two-phase flow calculation.</td>
</tr>
<tr>
<td>ISOSSPR</td>
<td>Grid block location of spray zone</td>
</tr>
<tr>
<td>ISPRAY1, ISPRAY2,</td>
<td>Spray nozzle location (i,j,k) between ISPRAY1, ISPRAY2 points in i direction, between ISPRAY1, ISPRAY2 points in j direction, and between ISPRAY1, ISPRAY2 points in k direction.</td>
</tr>
<tr>
<td>KSPRAY1, KSPRAY2</td>
<td>nangle = number of angles increments</td>
</tr>
<tr>
<td>ANGLE1, ANGLE2</td>
<td>between (angle1,angle2), angle1 = Lower spray angle (say 10 degree) angle2 = Upper angle (say 45 degree). Angle measured in x-z plane only.</td>
</tr>
<tr>
<td>SX1, SX2</td>
<td>= spray particle size range in terms of [um in radius).</td>
</tr>
<tr>
<td>NSPR</td>
<td>Time step to initiate the spray</td>
</tr>
<tr>
<td>NCEALSPR</td>
<td>Spray update increment</td>
</tr>
<tr>
<td>JMPR</td>
<td>= 0 (reserved for later version use)</td>
</tr>
<tr>
<td>TSIPR</td>
<td>Liquid spray temperature(K)</td>
</tr>
<tr>
<td>FLOWSPRY</td>
<td>Liquid spray mass flow rate (kg/s)</td>
</tr>
<tr>
<td>VTSPR</td>
<td>Liquid spray velocity (m/s)</td>
</tr>
<tr>
<td>INPLOT()</td>
<td>Plot file flag(up to 15), = 1 density, = 2, 3, 4, u-v-w momentums, = 5 static temperature, = 6 gage pressure, = 7 static pressure, = 8 turbulent Ck, = 9 static enthalpy, = 10 total enthalpy, = 11 turbulent kinetic energy, = 12 turbulent dissipation, = 13 turbulent eddy viscosity, = 14 distance to walls, = 14 species mass fraction</td>
</tr>
<tr>
<td>NSP</td>
<td>Number of species in IC</td>
</tr>
<tr>
<td>NAME, YOL(I)</td>
<td>Species name and Initial species mass fraction</td>
</tr>
</tbody>
</table>
### gridlink.dat - Grid Parameter Input File

(i): block location counter

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>:</td>
<td></td>
</tr>
<tr>
<td>name, Yst(ns)</td>
<td>Species name and Initial species mass fraction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>iblank</td>
<td>Flag for grid.dat, 0 = non-blanked grid.dat, 1 = blanked grid.dat</td>
</tr>
<tr>
<td>nblank</td>
<td>total number of grid blocks (&lt;NUM_ZONES&gt;)</td>
</tr>
<tr>
<td>lmaxb(), jmaxb(), kmaxb()</td>
<td>Number of grid points in i, j, and k direction, respectively, in nblank block (up to nblank blocks required)</td>
</tr>
<tr>
<td>least(i), west(i), north(i), south(i), top(i), bottom(i), bclm(i,1), ... bclm(i,6)</td>
<td>Block BC indices nblank block (up to nblank blocks required)</td>
</tr>
</tbody>
</table>

### prop.dat - Property Input File

specnm= name of the species Input(< 80 char enclosed with in "")

heading: Datafile Comments (Optional, but blank lines required)

<table>
<thead>
<tr>
<th>Input Variable</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>heading</td>
<td>title</td>
</tr>
<tr>
<td>nspec</td>
<td>number of species</td>
</tr>
<tr>
<td>specnm(nx),wm(nx)</td>
<td>wn(nx) = molecular weight for each species</td>
</tr>
<tr>
<td>nxv=1,ns</td>
<td></td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>specnm(nx), cp0(nx), cp1(nx), cp2(nx), cp3(nx), cp4(nx), cp5(nx), cp6(nx)</td>
<td>curve fit coefficients for specific heat</td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>specnm(nx), euv(nx), eu1(nx), eu2(nx), eu3(nx), eu4(nx)</td>
<td>curve fit coefficients for dynamic viscosity</td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>specnm(nx), cd0(nx), cd1(nx), cd2(nx), cd3(nx), cd4(nx)</td>
<td>curve fit coefficients for thermal conductivity</td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>specnm(nx), sigm(nx)</td>
<td>species name, effective collision integral factor, Equation(45)</td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>specnm(nx), edk(nx)</td>
<td>species name, effective temperature, Equation(45)</td>
</tr>
<tr>
<td>heading</td>
<td>title of the property</td>
</tr>
<tr>
<td>heading</td>
<td>Title</td>
</tr>
<tr>
<td>are</td>
<td>Number of reactions</td>
</tr>
<tr>
<td>Input Variable</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>heading</td>
<td>Title, helpful to write the reaction mechanism (blank lines are acceptable)</td>
</tr>
<tr>
<td>heading</td>
<td>Heading for Forward reaction rates</td>
</tr>
<tr>
<td>nr, ar(nr), ea(nr), (frr(nr),</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction Factor</td>
</tr>
<tr>
<td>heading</td>
<td>Heading for Backward reactions</td>
</tr>
<tr>
<td>nr, br(nr), ebr(nr), (brr(nr)</td>
<td>Reaction number, Pre-exponent, Activation Temperature[K], Temperature Correction Factor (note this is forward only reaction)</td>
</tr>
<tr>
<td>heading</td>
<td>Heading, Forward Stoichiometric Coefficients</td>
</tr>
<tr>
<td>nr, skst(n,1,ns)</td>
<td>Reaction number, Forward Stoichiometric Coefficients (There should be ns of them)</td>
</tr>
<tr>
<td>heading</td>
<td>Heading, Backward Stoichiometric Coefficients</td>
</tr>
<tr>
<td>nr, skstpp(n,ns)</td>
<td>Reaction number, Backward Stoichiometric Coefficients (There should be ns of them)</td>
</tr>
<tr>
<td>heading</td>
<td>Heading Title for Correction Mechanism</td>
</tr>
<tr>
<td>nr, thrid0(n)</td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in forward direction (for the ns species)</td>
</tr>
<tr>
<td>thrid0(n)</td>
<td>Reaction number, Reaction type, Power of the reaction mechanism in reverse direction (for the ns species)</td>
</tr>
<tr>
<td>perl,terl,tloli</td>
<td>Liquid phase information</td>
</tr>
<tr>
<td>tmelt</td>
<td>perl - critical pressure (N/m²)</td>
</tr>
<tr>
<td></td>
<td>tcrl - critical temperature (K)</td>
</tr>
<tr>
<td></td>
<td>tboill - boiling point temperature (K)</td>
</tr>
<tr>
<td></td>
<td>tmelit - melting point temperature (K)</td>
</tr>
<tr>
<td>hfg,cppar,rkpar,</td>
<td>Liquid phase information</td>
</tr>
<tr>
<td>rhopar</td>
<td>hfg- latent heat of vaporization (J/kg)</td>
</tr>
<tr>
<td></td>
<td>cppar- specific heat (J/kg-K)</td>
</tr>
<tr>
<td></td>
<td>rkpar- conductivity (J/m-s-K)</td>
</tr>
<tr>
<td></td>
<td>rhopar density (kg/m³)</td>
</tr>
<tr>
<td>vp1,vp2,vp3,vp4</td>
<td>Critical pressure curve fit coefficients</td>
</tr>
</tbody>
</table>

**Input Variable**

<table>
<thead>
<tr>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
</tr>
<tr>
<td>Thrid0(1) = -1, Global type kinetics</td>
</tr>
<tr>
<td>Thrid0(1) = 0, Modified kinetics with fractional power</td>
</tr>
<tr>
<td>Thrid0(1) &gt;0 detailed kinetics with 3-body corrections</td>
</tr>
<tr>
<td>$K_f = A \exp(E_f/RT) T^\gamma \exp(30h)^1$ [02]</td>
</tr>
<tr>
<td>$K_b = 0.0$</td>
</tr>
<tr>
<td>Mechanism of Westbrook and Dryer [37]</td>
</tr>
</tbody>
</table>

**grid.dat Input file format**

**For non-blanked format:**

```plaintext
read(units,*),nbblk
read(units,*),imax(n),imax(n),kmax(n),n=1,nbblk
do n=1,nbblk
read(units,*)
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
end do
```

**For blanked format:**

```plaintext
read(units,*),nbblk
read(units,*),imax(n),imax(n),kmax(n),n=1,nbblk
do n=1,nbblk
read(units,*)
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
    ((x(j,k),j=1,imax(n)),j=1,imax(n)),k=1,kmax(n)),
end do
```

**uwrt.dat Input file format**

**For Inlet BC input:**

```plaintext
do n=1,nbblk
read(units,*),imax(n),jmax(n),kmax(n),n=1,nbblk
```

**For swirl Input**

```plaintext
do n=1,nbblk
read(units,*),irmax
```

**For Mass fractions of the species, recup(i)=rmax, where r is the radial distance from the center of the swirl cup ucp(i), vcp(i), wcp(i)=u+v+w-velocity (m/s), tcup(i)= temperature (K), tcup(i)= turbulence intensity**
The dimensional units required for chemical kinetic rate pre-exponents \( A_f \) and \( A_b \) are in the M-K-S system.

\[
[A_f(k)] \text{ or } [A_b(k)] = \left[ \frac{\text{kmols}^{-1} \cdot \text{m}^3 \cdot \text{s}^{-3}}{\text{sec}} \right]
\]

\[
r_m = \sum_{i=1}^{N} v_{ik} \quad \text{or} \quad \sum_{i=1}^{N} v''_{ik}
\]

where the reaction is given by,

\[
\sum_{i=1}^{N} v_{ik} C_{i} = \sum_{i=1}^{N} v''_{ik} C_{i} \quad , \quad k = 1, 2, 3 \ldots Nr
\]

### ALLSPD Code Working Units

- **Length:** meter (m)
- **Time:** second (s)
- **Mass:** kilogram (kg)
- **Temperature:** Degrees Kelvin (K)
- **Energy:** Joules (J)
- **Force:** Newton (N)
- **Velocity:** meter/second (m/s)
- **Pressure:** N/m²
- **Density:** kg/m³
- **Specific heat:** J/(kg K)
- **Dynamic viscosity:** kg/(m s)
- **Kinematic viscosity:** m²/s

### ALLSPD Grid Orientation

- Y-Dir = North (jmax)
- X-Dir = West (jmin)
- Z-Dir = Bottom (kmin)
- Z-Dir = Top (kmax)
- Y-Dir = South (jmin)
This report describes the ALLSPD-3D computer program for the calculation of three-dimensional, chemically reacting flows with sprays. ALLSPD-3D employs a coupled, strongly implicit solution procedure for turbulent spray combustion flows. ALLSPD-3D utilizes generalized curvilinear coordinates with both multi-block and flexible internal blockage capabilities to handle complex geometries. The ALLSPD-3D algorithm, which employs preconditioning and eigenvalue rescaling techniques, is capable of providing efficient solution of flows over a wide range of subsonic Mach numbers. The chemistry treatment in the code is general enough that an arbitrary number of reactions and species can be defined by the user. In addition, for general industrial combustion applications, the code provides both dilution and transpiration cooling capabilities. A stochastic spray model and an efficient method for treating the spray source terms in the gas-phase equations are used to analyze evaporating spray droplets. Although written for three-dimensional flows, the code can be used for two-dimensional and axisymmetric flow computations as well. The code is written in such a way that it can be run on various computer platforms (supercomputers, workstations, and parallel processors) while the GUI (Graphical User Interface) should provide a user-friendly tool for setting up and running the code.