Appendix C

Description of Key Variables

compute v_{cl} and T_{cl}. Typically a few attempts are needed to establish the desired inflow profiles.

Notes on input data

Mesh definition

Considering only the x direction, NX is the number of real computational cells (i.e., without ghost cells), X(1) and XL are the upper and lower limits of the computational domain (in cm), and IGRIDX is a switch indicating whether a uniform (IGRIDX=0) or nonuniform (IGRIDX=1) mesh is desired. If the mesh is to be nonuniform, the USER.f subroutine will be called where mesh coordinates are specified or generated. Similar data are required for the y and z directions.

Torch definition

Input parameters are used to define the torch currency (CURCY), voltage (VOLTG), efficiency (EFFCY), and flow rates (FLRT1 and FLRT2). This information is used to provide an estimate of the inlet velocity and temperature profiles. Since individual species concentrations are strongly dependent upon the inlet temperature profile, an iteration is used within START.f to approximate the inflow conditions. As coded, LAVA assumes the velocity and temperature inlet profiles can be approximated as:

$$T = (T_{cl} - T_w)\left[1 - \left(\frac{r}{R}\right)^{\eta_T}\right] + T_w$$

$$v = v_{cl}\left(1 - \left(\frac{r}{R}\right)^{\eta_v}\right)$$

where r is the radial coordinate, R the torch radius, and the *cl* and *w* subscripts define the centerline and wall quantities, respectively. Unknown are the centerline temperature and velocity, plus the n_v and n_T exponents. The approach, then, is to select n_v and n_T (PWVEL and PWTEP in input file) and use the prescribed mass flow and torch power to

Appendix B

LAVA Sample Input File

Chemistry

A-6) J. D. Ramshaw and C. H. Chang, Iteration Scheme for Implicit Calculations of Kinetic and Equilibrium Chemical Reactions in Fluid Dynamics, *Journal of Computational Physics*, **116**, 359 (1995).

A-7) C. H. Chang and J. D. Ramshaw, Modeling of Nonequilibrium Effects in a High-Velocity Nitrogen-Hydrogen Plasma Jet, *Plasma Chemistry and Plasma Processing,* **16 (Supplement)**, 5S (1996).

Particles

A-8) C. H. Chang, Numerical Simulation of Alumina Spraying in an Argon-Helium Plasma Jet, T*hermal Spray:International Advances in Coatings Technology,* (ASM International, Materials Park, OH, 1992) p. 793.

A-9) Y. P. Wan, V. Prasad, G.-X. Wang, S. Sampath, and J. R. Fincke, Modeling of Powder Particle Heating, Melting and Vaporization in Plasma Spraying Processes, HTD-Vol. 361-4, Proceedings of the ASME Heat Transfer Division, Volume 4, pp. 66-77, 1998 (also to be published in the Journal of Heat Transfer).

Applications

A-10) C. H. Chang and J. D. Ramshaw, Numerical Simulation of Argon Plasma Jets Flowing into Cold Air, *Plasma Chemistry and Plasma Processing,* **13**, 189 (1993).

A-11) J. R. Fincke, C. H. Chang, W. D. Swank, and D. C. Haggard, Entrainment and Demixing in Subsonic Thermal Plasma Jets: Comparison of Measurements and Predictions, *Int. J. Heat Mass Transfer*, **37**, 1673 (1994).

A-12) C. H. Chang and J. D. Ramshaw, Computational Study of High-Speed Plasma Flow Impinging on an Enthalpy Probe, *Plasma Chemistry and Plasma Processing,* **16**, 17 (1996).

Appendix A

LAVA Key Publications

This Appendix contains a list of some key publications pertaining to LAVA. The articles are roughly categorized and, within categories, ordered from the most basic to the more specific and complex. For a description of code capabilities, the basic equations solved, and the numerical techniques employed, the reader is referred to these publications.

General Relations and Numerical Methods

A-1) J. D. Ramshaw and C. H. Chang, Computational Fluid Dynamics Modeling of Multicomponent Thermal Plasmas, *Plasma Chemistry and Plasma Processing*, **12**, 299 (1992).

A-2) A. A. Amsden, J. D. Ramshaw, P. J. O'Rourke, and J. K. Dukowicz, KIVA: A Computer Program for two- and three-dimensional fluid flows with chemical reactions and fuel sprays, *LANL Technical Report LA-10245-MS*, 1985.

A-3) J. D. Ramshaw and C. H. Chang, Ambipolar Diffusion in Multicomponent Plasmas, *Plasma Chemistry and Plasma Processing*, **11**, 395 (1991).

A-4) J. D. Ramshaw and C. H. Chang, Ambipolar Diffusion in Two-Temperature Multicomponent Plasmas, *Plasma Chemistry and Plasma Processing*, **13**, 489 (1993).

A-5) C. H. Chang and J. D. Ramshaw, Numerical Simulation of Nonequilibrium Effects in an Argon Plasma Jet, *Phys. Plasmas*, **1** 3698 (1994).

19) Move lava.x (the executable) and lava.inp (the input file) to the directory where they will be used. Run the code simply by typing “lava.x”.

References

1. M. W. Chase, C. A. Davies, J. R. Downey, D. J. Frurip, R. A. McDonald, and A. N. Syverud, JANEF Thermochemical Tables-Third Edition, *J. Phys. Chem. Ref. Data*, **14-Suppl. 1**, 1, 1985.

tivity, density and specific heat of the particle materials in both solid and liquid states.

- Modify FUNCTION routines defining the latent heat of vaporization and molecular weight of the particle materials.

When modified, move PTCHEATING.f back to the source directory (typically apgm).

15) Modify PRINT.f routine. This subroutine provides output in a variety of forms and is very much user-defined for each problem. It is probably easiest to use the existing routine as a template and design your own output.

When modified, move PRINT.f back to the source directory (typically apgm).

16) Modify TECPLOT.f and HDFOUT.f routines. These subroutines provide output in either TECPLOT or HDF formats for plotting. Modifications can be made in either routine to control which data is available for plotting.

When modified, move TECPLOT.f and HGFOUT.f back to the source directory (typically apgm).

17) Develop input file (lava.inp). All input information not hard-coded as described above is provided in this file. Rather than a line-by-line description of the input structure, a typical input file for the FGM problem is included in Appendix B. This appendix does include some description of the input required for mesh development and specification of torch inlet conditions. Most input variables can be identified simply by comparing the variable name with its description in the FORTRAN routine EPIL.f, a copy of which is included in Appendix C.

18) Compile/load all source code using “make” file in source directory.

bulence wall function; the approach and basic logic is explained in references A-2 and A-8. It is necessary to define the torch face indexing (DATA statement, see Step 9 and Figure 3) as well as how the various chemical species in the problem contribute to the plasma (again, analogous to he approach in Step 5).

When modified, move WALLF.f back to the source directory (typically apgm).

13) Modify USERPP.f routine. This subroutine contains plasma properties needed to compute particle-plasma interactions. As currently coded, viscosities and thermal conductivities are available for argon, hydrogen, nitrogen and air plasmas as well as for an argon-helium mixture. If the problem being set up involves one or more of these plasmas, their DATA statements are activated and all others are “commented-out”. If other plasmas are present, new property data must be included in the routine. It is also necessary to define how the various chemical species in the problem contribute to these plasmas, much the same as was done in Step 5 above. Remember that appropriate declaration statements (e.g., DOUBLE PRECISION) are needed for any newly defined variables.

USERPP.f contains empirical correlations for particle drag and heat transfer coefficients; comment statements provide references where these relations were obtained. If different correlations are desired, they can be implemented here.

USERPP.f also contains the models and logic to predict particle evaporation. Two vaporization models are currently provided. The particular model used is governed by a flag in the input file.

When modified, move USERPP.f back to the source directory (typically apgm).

14) Modify PTCHEATING.f as needed. PTCHEATING.f is actually a collection of subroutines added to LAVA by Y. P. Wan (1998) to model particle heat conduction. Typical user modifications are as follows:

- Modify subroutine SETUPPTC to define the finite difference grid within the particles.
- Modify a variety of FUNCTION routines which are used to define the thermal conduc-

often and is probably best disabled.

When modified, move CHMIMP.f back to the source directory (typically apgm).

11) Modify the START.f routine which mainly provides initial and boundary condition information required to start a calculation. Much of the routine is an iteration used to approximate the inlet velocity and temperature profiles. Required modifications typically include the following:

- Define LNREV in the PARAMETER statement. LNREV is equal to the number of independent chemical reactions for the inflowing plasma. For the simple N_2 test problem, there are four chemical reactions solved, however, only three are independent, thus LNREV = 3.
- Define indexing on the torch wall (DATA statement). See Step 9 and Figure 3.
- Define boundary condition flags.
- Define inflow species on all boundaries.
- Define inflow species from torch (torch power, efficiency, and flow rates, velocity and temperature profiles and swirl number are all set in input).
- If necessary, modify logic setup for call to subroutine CHEMEV, which computes the equilibrium chemistry for the inflowing plasma. Since CHEMEV.f does not share COMMON blocks with LAVA, a number of local arrays are filled in START.f for passage during the subroutine call. Users are advised to take a careful look at this logic to insure the arrays are loaded correctly.
- Define the initial state in the computational domain. By default, LAVA assumes a cold stagnant gas. No changes are required to the pressure, temperature or velocity if this matches the initial state in the problem being defined. Typically it is necessary to modify the initial species in the domain.
- Insure that any newly introduced variables are in “DOUBLE PRECISION” statements.

When modified, move START.f back to the source directory (typically apgm).

12) Modify WALLF.f routine. This routine updates velocities and other scaler variables by the tur-

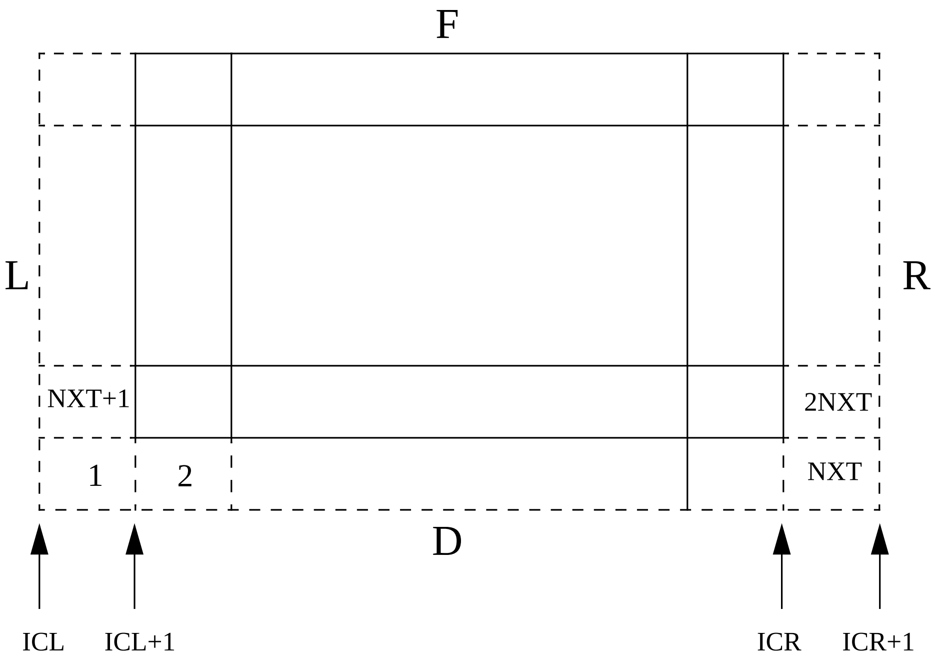
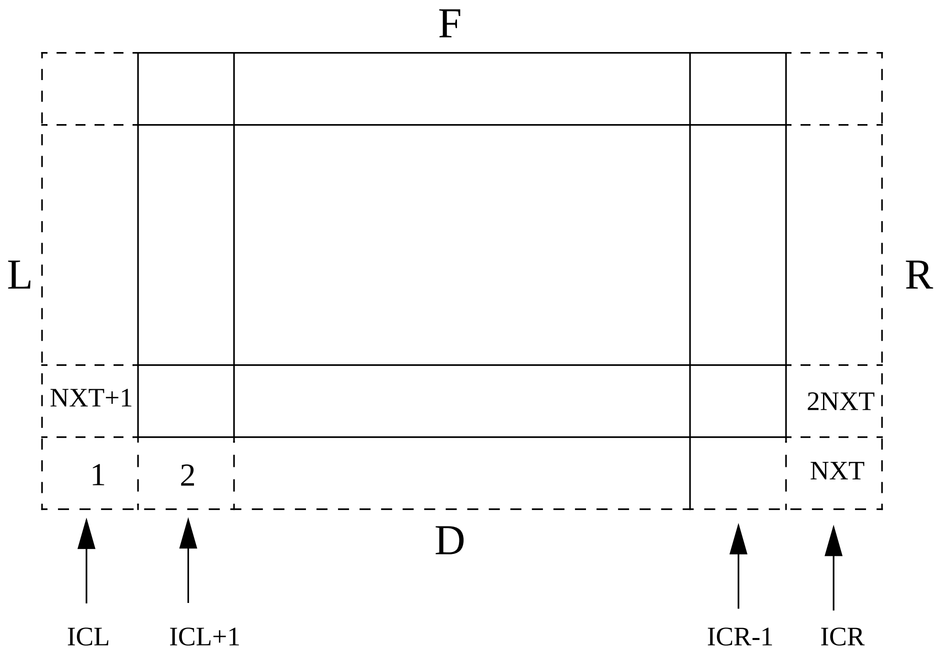


Figure 4. Indexing scheme for the left and right velocity boundary conditions.

where C_f (cm^3/mole-s), Z_f (dimensionless) and E_f (K) are provided in the input data file (lava.inp) and T is the temperature. As with the equilibrium coefficients, if a particular reaction does not fit this standard form, alternative functional forms must be hard-coded in subroutine CHMIMP.f. Relations for both the rate coefficient (k_f) and d(ln k_f)/dT must be provided. Comment statements show where this coding can be inserted in the subroutine. Note that the indexing within the subroutine must correspond with the chemical reaction indexing specified in the input file.

It is also recommended that users check the write statement which provides a singular matrix warning. If the chemical reactions are not independent, this warning will be printed very



NX - number of cells in x direction, excluding ghost cells

NXT - number of cells in x direction, including ghost cells

ICL - left boundary index

ICR - right boundary index

Figure 2. Indexing scheme for the left and right cell-centered boundaries.

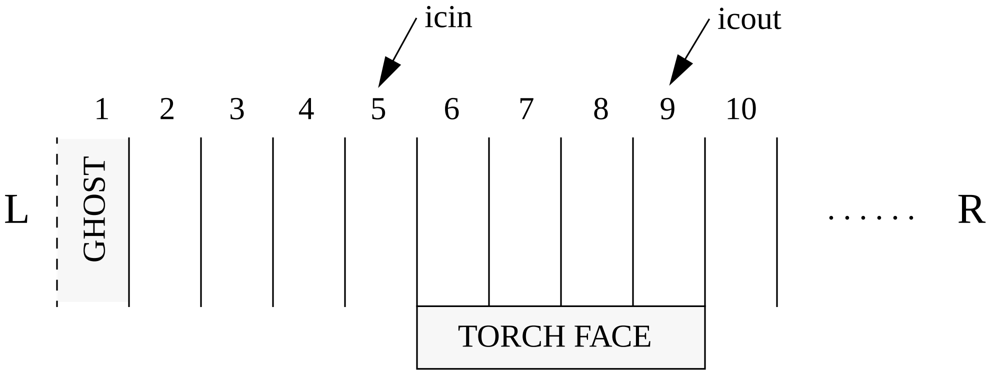


Figure 3. Typical indexing scheme for a torch face.

9) Define boundary conditions by modifying BCCC.f and BCMOM.f

BCCC.f assists with the definition of Cell-Centered boundary conditions. As coded, BCCC.f contains logic statements for “Dirchlet” type boundaries, with the actual boundary values set in subroutine START.f. Cell-centered “Neumann” type boundaries are completely set in BCCC.f. The subroutine is reasonably straight-forward and will not be described in detail here. Figure 2 shows the left-right indexing scheme (i.e., in the x-direction) assumed in the code. Similar indexes are defined for the derriere (ICD) and front (ICF) boundaries and top (ICT) and bottom (ICB) boundaries. Note that if a torch face (wall) is present, the “icin” and “icout” indexes must be set in a DATA statement. An example in Figure 3 shows how these are defined.

BCMOM.f defines velocity boundary conditions at all cell edges. Figure 4 shows the indexing scheme at the left and right boundaries, or in the x direction; similar indexes are used for the y and z directions. Note that the cell-centered and velocity boundary condition indexes are similar on the left hand side, but differ on the right.

When modified, move BCCC.f and BCMOM.f back to the source directory (typically apgm).

10) Modify chemistry routine CHMIMP.f if needed.

In LAVA, the standard functional form for equilibrium coefficients is given by:

$$K_{eq} = exp\left(A_s ln T_A + \frac{B_s}{T_A} + C_s + D_s T_A + E_s T^2\right)$$

where A_s, B_s, C_s, D_s and E_s are constants provided in the input file (lava.inp) and T_A is the temperature divided by 1000. If a particular reaction does not fit this standard form, alternative functional forms must be hard-coded in subroutine CHMIMP.f. Relations for both the equilibrium coefficient (K_{eq}) and d(ln K_{eq})/dT must be provided.

The standard equation for forward rate coefficients is the Arrhenius form given by:

$$k_f = C_f T^{Z_f} e^{\left(\frac{E_f}{T}\right)}$$

unnecessary statements. For example, for the simple one-component nitrogen plasma example problem considered above, one would "comment-out" all XPTY INCLUDE statements except N2.V, N2.K and N2.R.

- Modify the INCLUDE statements such that only the appropriate species are included for enthalpy properties. For the simple nitrogen plasma example problem with five species, all ENTHALPY INCLUDE statements would be "commented-out" except N2.E, N2+.E, N.E, N+.E, and e-.E.

When modified, move HOT.f back to the source directory (typically apgm).

8) If required, set up any nonuniform meshing in subroutine USER.f. Recall from the previous step that if either IGRIDX, IGRIDY or IGRIDZ are set to one in HOT.f, subroutine USER.f is called wherein the mesh coordinates (e.g., x(1), x(2), . . . , x(NX)) are defined. Note that spatial dimensions are assumed to be in centimeters, and the mesh definition must include the ghost cells (x(0) and x(NXT)) as shown in Figure 1 for the x direction.

When modified, move USER.f back to the source directory (typically apgm).

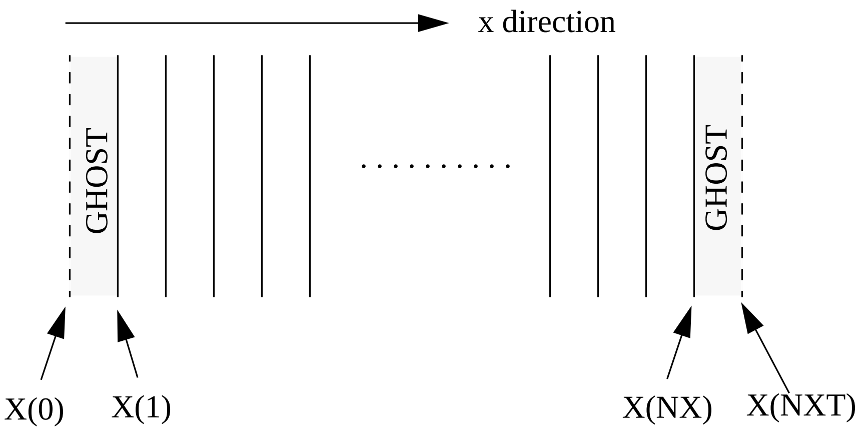


Figure 1 Grid nomenclature for modifications to subroutine USER.f.

```
cNb = spd(ic,1)*rmw(1)+spd(ic,2)*rmw(2)+half*(spd(ic,3)*rmw(3)+spd(ic,4)*rmw(4))
```

Component volume fractions are then computed from the molar densities.

When modified, move XPORT.f back to the source directory (typically apgm).

6) Modify the COML.h routine. Typically, only three of the four parameter statements (which control array dimensioning) near the front of the routine must be modified. For a description of the FORTRAN variables used in LAVA, see the EPIL.f (EPILOG) file.

As an example, for the simple N_2 test problem considered above, and assuming one kinetic and three equilibrium chemical reactions and a 10x10 two-dimensional mesh, the following variables would be set in COML.h:

NXT = 12	x cells + 2 ghost cells
NYT = 12	y cells + 2 ghost cells
NSP = 5	number of chemical species
LNRK = 1	number of kinetic reactions
LNRE = 3	number of equilibrium reactions
NCMP = 1	number of “components” for transport properties

When modified, move COML.h back to the source directory (typically apgm).

7) Modify the HOT.f routine. The first 113 lines are standard coding and are typically not changed; below this point, modifications are required as follows:

- Modify the INCLUDE statements such that only the appropriate components are included for transport properties. Typically this is done by “commenting-out” any

Species Number	Species	File in ENTHALPY subdirectory	Index in DATA statement
1	N_2	N2.E	1
2	N_2^+	N2+.E	2
3	N	N.E	3
4	N^+	N+.E	4
5	e^-	e-.E	5

When modifications are complete, move the ENTHALPY subdirectory back to the source directory (typically apgm).

5) Modify the XPORT.f routine. As described in Step 3, transport properties are approximated via mixture rules based on a simplified set of plasma “components” rather than trying to account for properties of individual species. In XPORT.f, the FORTRAN statements in lower case must be modified to define component volume fractions based on the density and molecular weight of each species; typical statements are included in the coding. Remember that appropriate declaration statements (e.g., DOUBLE PRECISION) are needed for any newly defined variables.

As an example, consider the N_2 test problem described above. The molar density of the nitrogen “component,” would be defined as:

$$C = \rho_{N_2}\left(\frac{1}{M_{N_2}}\right) + \rho_{N_2^\dagger}\left(\frac{1}{M_{N_2^\dagger}}\right) + \frac{1}{2}\left[\rho_N\left(\frac{1}{M_N}\right) + \rho_{N^\dagger}\left(\frac{1}{M_{N^\dagger}}\right)\right]$$

where ρ is the species density, M the molecular weight, and the individual species are identified in Step 4. Note that the N and N^+ contributions are multiplied by 1/2 since properties are specified for cold species. The FORTRAN statement for the above equation would be:

N2.K	thermal conductivity
N2.V	viscosity
N2.R	radiation loss

are modified to set the component number (shown underlined in the DATA statement below) to one. Note that this change must be made in four separate places in the routine.

DATA (CND(N,1),N=1,50)

When modifications are complete, move the XPTY subdirectory back to the source directory (typically apgm).

4) Define enthalpy property data. The data are arranged in individual files for each species, in the same manner as is done for transport properties in step 3. Again, the temperature increment is 100 K and ranges from 0-20,000 K. The data are taken directly from the JANEF[1] tables, however these tables end at 6000 K. For molecular gases (e.g., H_2, N_2, O_2, Air) a linear extrapolation is used beyond 6000 K; for monotonic gases, the linear extrapolation is not sufficiently accurate, and the approach is to use “partition” functions (see reference 27 in [A-1]).

The ENTHALPY subdirectory contains files of temperature dependent enthalpy data for several species typically encountered in plasma sprays. The user must decide a species numbering scheme and appropriately modify the indexes in these routines; the indexes must be consistent with the input data (lava.inp). Obviously, if the data are not available in ENTHALPY, new coding must be developed.

As an example, again consider a simple problem involving only a nitrogen plasma. Assuming five different species are identified on input, five routines/files must be modified to provide the proper indexing (see table below). The indexes are located in FORTRAN DATA statements just as was shown in step 3; again, indexes must be modified in four places.

Additionally, two subdirectories, /ENTHALPY and /XPTY should be copied “one level up”, modified, and copied back into the source directory. These directories contain property data which will be described in more detail in steps 3 and 4 below.

3) Define transport property data. Since in a typical simulation many different species are involved and transport data is often not available for all species, transport properties in LAVA are currently approximated using a mixture rule, assuming only a few species (e.g., those with known properties) are present. For example, in the FGM simulation, 15 different species are included in the simulation, however the transport properties are approximated using a combined flow of Ar, H_2, and Air. Thus the “effective” properties are given by:

$$P = P_{Ar}V_{Ar} + P_{H_2}V_{H_2} + P_{Air}V_{Air}$$

where P is a typical property (either viscosity, thermal conductivity, or radiation loss) and V is the volume fraction. Note that “radiation loss” is not a transport property, but is included with the transport properties because it is temperature dependent.

The XPTY subdirectory contains files of temperature dependent transport property data for several components typically encountered in plasma sprays. The data are arranged in FORTRAN data statements with temperature increments of 100 K (thus the 201 entries provide a temperature range of 0-20,000 K). The user must decide a component numbering scheme and appropriately modify the indexes in these routines; further, these indexes must be consistent with the XPORT.f routine. Note that the indexes are for *components* (e.g., Ar, H_2, Air) rather than the *individual species* in the simulation. Obviously, if the data are not available in XPTY, new coding must be developed.

As an example, consider a simple problem involving only a nitrogen plasma. For transport properties, the only component will be N_2, and will be given a component index of 1. Thus the following existing data files:

Input Set-up Procedure

1) Begin with a fresh copy of the source code. Typically this “source” exists as a tar file (e.g. apgm.tar) which is moved to an empty directory and unpacked via a tar command (e.g., tar xvf apgm.tar), which builds a subdirectory entitled “apgm” containing all FORTRAN source routines, a UNIX make file and a typical input data file (lava.inp).

2) Copy all FORTRAN routines which might require input modifications “one level up” (e.g., cp LAVA.f ..). Then, when appropriately modified, the routine can be copied back into the source subdirectory (e.g., cp LAVA.f apgm) where it will be available for later compilation. This is simply a bookkeeping technique which keeps the routines needing modification separate and provides an unmodified backup copy of each routine during modification. The FORTRAN routines which contain input type coding and potentially require modification are listed below, along with a brief description of their general purpose:

BCCC.f sets cell centered boundary conditions

BCMOM.f sets velocity boundary conditions

CHMIMP.f calculates the change in species densities and internal energy due to fast kinetic and equilibrium chemical reactions

COMLh contains parameter statements and general common blocks

HOT.f initializes data and constants

PRINT.f controls printing of information during time marching

START.f sets boundary and initial conditions (including inlet velocity and temperature profile)

WALLF.f updates velocities and other scaler variables by turbulent wall function

XPORT.f provides viscosity, conductivity, and diffusion coefficients

USER.f provides user specified grid information

USERPP.f provides drag and heat transfer coefficients for particles

Introduction

The LAVA code was developed over the past ten years as a research tool for the simulation of thermal plasma jets. At the present time, the only existing formal documentation for the code are a variety of technical articles which were published by the code developers in various scientific journals. Appendix A contains of list of the key articles, roughly categorized and ordered from the most basic to the more specific and complex. For a description of code capabilities, the basic equations solved, and the numerical techniques employed, the reader is referred to these articles.

Because LAVA was developed as a research tool and has been used principally by the code authors, no effort has been made to make the input structure “user-friendly”. Much of the input must be done via “hard-coding” in FORTRAN; input error checking is very limited. Obviously, use of the code by other than its creators can be difficult. This brief guide is an effort to provide a somewhat systematic procedure one can follow in order to set up a thermal plasma jet simulation. The guide grew out of a set of notes which were developed as the author spent a few days with one of the code’s creators (C.H. Chang; spring 1997) learning how to use the software.

Note that since LAVA has been an evolving code, many versions of the software exist. This guide is specific to the version current during the fall of 1998 (including particle conduction, melting, and vaporization). Note further that there is nothing sacred about the order of the steps in the procedure. Obviously some must come before others, but mostly the order was that preferred by Chang, who had used the software extensively.

As a general note, the authors of LAVA have made an effort to identify which portions of coding should be considered for modification during input and which should not. Although this convention is not adhered to universally, as a general rule, any FORTRAN coding in lower case is problem dependent and should be considered for modification during problem setup; any coding in upper case is typically not modified.

Input Guide for the
LAVA Plasma Jet Simulation Program

R. L. Williamson

*Idaho National Engineering
and Environmental Laboratory*

January 1999