AEDC-TR-05-2

APTU Nozzle Code Manual
Version 3.0

Graham V. Candler
110 Union Street SE
Minneapolis, MN 55455
candler@aem.umn.edu

January 2005

Final Report for Period 1 October 2002 — 30 September 2004

STATEMENT A: Approved for public release; distribution unlimited.

ARNOLD ENGINEERING DEVELOPMENT CENTER
ARNOLD AIR FORCE BASE, TENNESSEE
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Technical Information Center.

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

DESTRUCTION NOTICE

For unclassified documents, destroy by any method that will prevent disclosure or reconstruction of the document.

APPROVAL STATEMENT

This report has been reviewed and approved.

ROBERT A. WILSON, CAPT
Applied Technology Division
Test Operations Directorate

Approved for publication:

FOR THE COMMANDER

ROBERT T. CROOK
Deputy Chief, Applied Technology Division
Test Operations Directorate

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
00-01-2005	Final	1 October 2002 – 30 September 2004

4. TITLE AND SUBTITLE

APTU Nozzle Code Manual, Version 3.0

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Graham V. Candler

5d. PROJECT NUMBER
9829

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

Graham V. Candler
110 Union Street Se
Minneapolis, MN 55455

8. PERFORMING ORGANIZATION REPORT NO.

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

Arnold Engineering Development Center
Directorate of Technology
1099 Avenue C
Arnold AFB TN 37389-9013

10. SPONSOR/MONITOR'S ACRONYM(S)
AEDC/DOT

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AEDC-TR-05-2

12. DISTRIBUTION/AVAILABILITY STATEMENT

STATEMENT A: Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Available in the Defense Technical Information Center (DTIC).

14. ABSTRACT

A nozzle simulation code developed for AEDC is designed for the analysis of hypersonic converging-diverging nozzles including the effects of high pressure, finite-rate chemical reactions and vibrational relaxation, and turbulent wall boundary layers. In addition, a module has been written to predict the onset of water vapor condensation for hydrocarbon combustion product test gases. The flow solver is based on the data-parallel line relaxation (DPLR) numerical technique. The software package includes chemical species set-up codes, a batch grid generator, the flow solver, and a post processor. Gas models include air, nitrogen, and the products of combustion of isobutene with air and make-up oxygen. Software access is limited to qualified users.

15. SUBJECT TERMS
chemically reacting Navier-Stokes computer program

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Capt. Robert A. Wilson
Unclassified	Unclassified	Unclassified	SAR	32	19b. TELEPHONE NUMBER *(Include area code)* 931-454-5261

Table of Contents

1. Physical Models

The nozzle simulation code developed for AEDC is designed for the analysis of hypersonic converging-diverging nozzles including the effects of high pressure, finite-rate chemical reactions and vibrational relaxation, and turbulent wall boundary layers. In addition, a module has been written to predict the onset of water vapor condensation for hydrocarbon combustion product test gases. In the following section, the physical models are discussed; additional details will be available in a paper under preparation for publication.

1.1 High-Pressure Model

The nozzle simulation code uses the excluded volume thermal equation of state (also known as the Abel or Clausius equation of state). This is the first correction to the perfect gas equation of state, and it accounts for the volume occupied by the gas particles. The thermal equation of state is written as[1]

$$p = \frac{\rho RT}{1 - b_o \rho / \bar{M}}$$

where R is the mixture gas constant, b_o is the co-volume of the mixture, and $\bar{M}$ is the mixture average molecular weight. For a reacting gas, $b_o/\bar{M}$ is a constant, making the model depend on only a single parameter. b_o is obtained from the effective molecular diameter, σ, with

$$b_o = \frac{2}{3}\pi\sigma^3 N_o$$

where N_o is the Avogadro number. The model works well for high pressure flows that are not near the critical state, as is the case for hypersonic nozzle flows.

Note that the excluded-volume equation of state must be used when computing the nozzle supply conditions. In addition, if the specific enthalpy of the gas mixture is required, the correct expression for enthalpy must be used:

$$h = e + p/\rho = e + \frac{RT}{1 - b_o \rho / \bar{M}}$$

The code includes the high-pressure correction in the inflow boundary condition.

The value of the parameter $b_o/\bar{M}$ is obtained from fitting to high-accuracy equation of state data over the range of interest.[2,3] For pure N_2 the appropriate value is

$b_o/\bar{M} = 0.001120\,\mathrm{m}^3/\mathrm{kg}$ and for Air $b_o/\bar{M} = 0.001024\,\mathrm{m}^3/\mathrm{kg}$. For other gases, the effective molecular diameter may be obtained from Table 1 or other sources. The mixture value is then computed by number-weighting (or mole-weighting) the pure gas values:

$$\frac{b_o}{\bar{M}}\Big|_{\mathrm{mix}} = \frac{2}{3}\pi N_o \sum_s \gamma_s \frac{\sigma_s^3}{M_s}$$

where the sum is over all species in the gas mixture, and γ_s is the mole fraction of species s.

Gas	Symbol	Molecular Diameter (Å)
Argon	Ar	3.8
Carbon Dioxide	CO_2	2.8
Carbon Monoxide	CO	2.8
Ethane	C_2H_6	4.4
Helium	He	2.0
Hydrogen	H_2	2.4
Methane	CH_4	4.4
n-Butane	C_4H_{10}	4.9
Nitrogen	N_2	3.0
Oxygen	O_2	2.8
Propane	C_3H_8	4.9
Water	H_2O	2.8

Table 1. Effective molecular diameters for relevant gases; from `www.domnickhunter.com`.

1.2 Internal Energy

In hypersonic nozzle flows, the vibrational energy modes of the gas are often excited. The nozzle code models the vibrational excitation using a simple harmonic oscillator, which is valid for typical reservoir temperatures. The gas mixture vibrational energy per unit volume, E_v, is given by

$$E_v = \sum_s \rho_s e_{vs}$$

where the vibrational energy for each species is

$$e_{vs} = \frac{R}{M_s} \sum_r g_{rs} \frac{\theta_{v\,rs}}{\exp(\theta_{v\,rs}/T_v) - 1}$$

where r is summed over each vibrational mode of species s, $\theta_{v\,rs}$ is the characteristic temperature of vibration for that mode, and g_{rs} is the degeneracy of that mode. These constants are well known and are available from sources such as the NIST-JANAF Thermochemical Tables[4] (previously the JANAF Thermochemical Tables). In the above expression, T_v is the vibrational temperature that characterizes the gas mixture. Thus, we have assumed that the vibrational modes of the gas relax with one another relatively quickly, resulting in a single vibrational temperature.

The vibrational energy modes are assumed to relax toward the translational modes using the Landau-Teller model. We compute a number-weighted relaxation time using the individual species relaxation times from Millikan and White.[5] This model is accurate for most species pairs, however it is known to be inadequate for carbon dioxide relaxation because of resonant energy transfer between modes. Therefore, we use the results of Camac[6] to improve the vibrational energy relaxation model when carbon dioxide is present.

To construct the total energy per unit volume of the mixture, E, we sum the translational-rotational energy, the vibrational energy, and the kinetic energy

$$E = \sum_s \rho_s c_{vs} T + E_v + \frac{1}{2}\rho(u^2 + v^2) + \sum_s \rho_s h_s^o$$

where c_{vs} is the translational-rotational specific heat at constant volume for species s. For example, diatomic molecules have $c_{vs} = \frac{5}{2}\frac{R}{M_s}$, and atoms $c_{vs} = \frac{3}{2}\frac{R}{M_s}$. h_s^o is the heat of formation of species s.

The code is set up so that the molecular parameters are input variables, allowing new gas models to be constructed relatively easily.

1.3 Finite-Rate Chemical Reactions

The chemical composition of the gas changes as it flows through the converging-diverging nozzle. In many cases, the expansion is so sudden that an chemical equilibrium assumption is not valid, and we must consider finite-rate recombination of the test gas. To do so, we solve a mass conservation equation for each chemical species, including a mass production or destruction term due to chemical reactions. The equation is

$$\frac{\partial \rho_s}{\partial t} + \frac{\partial}{\partial x_j}\left(\rho_s u_j + \rho_s v_{sj}\right) = w_s$$

where we have used index notation. Here v_{sj} is the diffusion velocity of species s relative to the mass-averaged velocity, u_j. w_s is the chemical source term due to reactions; this is constructed from the law of mass action and an appropriate chemical kinetics model.

For typical hydrocarbon reaction products, we consider an 11-species CHON model consisting of CO_2, H_2O, CO, N_2, O_2, H_2, OH, O, H, and N. The kinetics model used is based on rates taken from the Gas Research Institute Mechanism 3.0 (GRI-Mech3.0). The reactions considered are:

$$H_2 + M \rightleftharpoons H + H + M \tag{R1}$$

$$H_2O + M \rightleftharpoons OH + H + M \tag{R2}$$

$$OH + M \rightleftharpoons O + H + M \tag{R3}$$

$$O_2 + M \rightleftharpoons O + O + M \tag{R4}$$

$$CO_2 + M \rightleftharpoons CO + O + M \tag{R5}$$

$$NO + M \rightleftharpoons N + O + M \tag{R6}$$

$$N_2 + M \rightleftharpoons N + N + M \tag{R7}$$

$$O_2 + H \rightleftharpoons OH + O \tag{R8}$$

$$H_2 + O \rightleftharpoons OH + H \tag{R9}$$

$$OH + OH \rightleftharpoons H_2O + H \tag{R10}$$

$$OH + H_2 \rightleftharpoons H_2O + H \tag{R11}$$

$$OH + CO \rightleftharpoons CO_2 + H \tag{R12}$$

$$NO + N \rightleftharpoons N_2 + O \tag{R13}$$

$$O_2 + N \rightleftharpoons NO + O \tag{R14}$$

$$O_2 + CO \rightleftharpoons O + CO_2 \tag{R15}$$

$$N + OH \rightleftharpoons NO + H \tag{R16}$$

For air flows, we use a 5-species model with 5 reactions (R4, R6, R7, R13 and R14); and for a pure nitrogen flow we just use the nitrogen dissociation/recombination reaction (R7).

The forward reaction rates for each of these reactions is from GRI-Mech3.0, and we compute the backward rates using the forward rate and the equilibrium constant. We curve fit the equilibrium constant using a five-parameter fit over the temperature range of interest by minimizing the Gibbs free energy based on the Gordon and McBride thermochemical data.[7] The curve fits provided with the code are valid in the range $200 \leq T \leq 3200\,\mathrm{K}$; a different temperature range can be fit with minimal effort as needed. All equilibrium constants have been verified by comparison with the GRIMech3.0 tabulations.

1.4 Transport Properties and Turbulence Modeling

The molecular viscosity, thermal conductivity, and mass diffusivity are computed using the standard Wilke mixing rule[8], with pure species data obtained from the Blottner curve fits.[9] All relevant data are provided in the input files for the nozzle simulation code. In the case of nitrogen gas, the viscosity fits from AEDC are used

$$\mu = \begin{cases} 2.8676 \times 10^{-8}\, T^{0.979} & T \leq 227\,\mathrm{R}; \\ 6.8971 \times 10^{-7}\, T^{1.5}/(179.1 + T) & 227 < T \leq 795\,\mathrm{R}; \\ 1.9466 \times 10^{-7}\, T^{0.659} & T > 795\,\mathrm{R}. \end{cases}$$

where T is in Rankine and μ is in lbm/ft s (multiply by 1.4882 to obtain SI units of kg/m s).

The translational-rotational thermal conductivity is obtained through the use of an Eucken relation, which is essentially a variation of the Prandtl number to account for the mixture of different components. The thermal conductivity of pure species s is

$$k_s = \mu_s \left(\frac{9}{4} + c_{vs}\right) \frac{R}{M_s}.$$

The mixture value is then obtained using the Wilke mixing approach. The vibrational energy conduction is written as:

$$q_{vj} = \sum_s \eta_{vs} \mu_s \frac{\partial e_{vs}}{\partial x_j}$$

which is more accurate and stable than basing the vibrational heat flux on the vibrational temperature. The value of η_{vs} is taken to be 1.2 from Ref. 10. The mass diffusivity is based on simple Fickian diffusion with a constant Lewis or Schmidt number.

It is critical to predict the correct boundary layer thickness in hypersonic nozzles because it determines the effective area ratio of the nozzle, and therefore the facility operating

conditions. In hypersonic facilities, the nozzle wall boundary layers are almost always turbulent because of the extremely high unit Reynolds number in the throat region. However, conventional algebraic turbulence models (*e.g.* Baldwin-Lomax) do not work well because of the highly favorable pressure gradient and strong temperature gradients in the boundary layer. We have performed extensive comparisons between pitot pressure rake measurements and nozzle wall heat transfer rate measurements in the CUBRC LENS facilities.[11] We find that the one-equation Spalart-Allmaras turbulence model[12] with the Catris and Aupoix compressibility correction[13] gives good agreement with experiments across a wide range of nozzle operating conditions. This is the turbulence model that is implemented in the APTU nozzle code.

The code initializes the inflow turbulent viscosity according to the criteria given in Ref. 13. There are no other adjustable constants in the present implementation; the turbulent boundary layer is allowed to develop along the converging section of the nozzle according to the local flow conditions. Again, because hypersonic nozzles operate at high pressure stagnation conditions, the unit Reynolds numbers are so high that the boundary layer is fully developed before the flow reaches the nozzle throat.

1.5 Water Vapor Condensation

When hydrocarbon combustion products are expanded through a nozzle, the water vapor may condense into droplets which cause a sudden entropy rise and adversely affect tunnel operation. We have implemented a condensation onset prediction model in the code based on the work of Erickson *et al.*[14]

The main criterion for condensation onset is based on the local saturation temperature of the gas mixture. Once the gas becomes saturated, nucleation will begin. Under most hypersonic nozzle flow conditions, the nucleation rate is very large and the gas condenses suddenly when it becomes saturated.[14] Therefore, if we monitor the saturation temperature relative to the gas temperature, we can determine where condensation if likely to occur. In addition, Ref. 14 gives a model for the nucleation rate, J, which can be used to estimate the rate at which condensation will occur. However, in practice this rate is typically so large that condensation is very rapid once the gas is saturated. (Particularly because

the gas continues to expand in the nozzle, making the flow more and more susceptible to nucleation and condensation.)

We have implemented the Erickson *et al.* model for a combustion gas mixture. The saturation temperature, $T_{\rm sat}$ is found by solving the equation

$$p_{\rm H_2O} = \exp\bigl(55.897 - \frac{6641.7}{T_{\rm sat}} - 4.4864\,\ln T_{\rm sat}\bigr)$$

where $p_{\rm H_2O}$ is the local partial pressure of water vapor in the flow. This expression comes from a curve-fit to the vapor pressure of water at a given temperature. Therefore, when $T_{\rm sat} = T$, the gas is saturated and can begin to nucleate and condense. For $T_{\rm sat} > T$, the gas is supersaturated and nucleation will be very rapid.

The nucleation rate can be found by first computing the critical droplet radius, r_*, for water vapor

$$r_* = \frac{2\sigma_w}{\rho_\ell R_{\rm H_2O} T \ln(p_{\rm H_2O}/p_v)}$$

where ρ_ℓ is the density of liquid water $p_{\rm H_2O}$ is the partial pressure of water vapor, and p_v is the vapor pressure of water over a flat surface. An expression for p_v was fit in Ref. 14 as

$$p_v = \exp\bigl(55.897 - \frac{6641.7}{T} - 4.4864\,\ln T\bigr).$$

In the above expression, σ_w is the surface tension of water given by

$$\sigma_w = (82.27 + 75.612\,T_R - 256.889\,T_R^2 + 95.928\,T_R^3) \times 10^{-3}$$

where $T_R = T/T_c$ with $T_c = 647.3\,{\rm K}$ the critical temperature of water. Droplets larger than r_* tend to grow, while droplets less than r_* evaporate.

Erickson *et al.* derive an expression for the nucleation rate of water in a carrier gas. Their expression is

$$J = \frac{q_c}{1+Q}\frac{\rho_{\rm H_2O}^2}{\rho_\ell}\sqrt{\frac{2\sigma_w}{\pi m_{\rm H_2O}^3}}\,\exp\bigl(\frac{-4\pi r_*^2\sigma_w}{3kT}\bigr)$$

where q_c is the condensation coefficient (taken to be unity in Ref. 14) and

$$Q = \frac{2(\gamma_{\rm H_2O}-1)}{\gamma_{\rm H_2O}+1}\frac{L}{R_{\rm H_2O}T}\bigl(\frac{L}{R_{\rm H_2O}T} - \frac{1}{2}\bigr)$$

is the nonisothermal correction factor. L is the latent heat of evaporation at T

$$L = h_{\mathrm{H_2O}} - 4.2 \times 10^3\, T + 17.1175 \times 10^6$$

where $h_{\mathrm{H_2O}}$ is the enthalpy of gaseous water and the units of L are J/kg. Therefore, we can evaluate the rate of formation of water droplets as a function of the local state of the gas in the nozzle expansion. A sudden increase in J indicates onset of condensation and rapid formation of water droplets.

Note that we do not need to compute any of these quantities unless $T_{\mathrm{sat}} > T$. Also, we do not couple the nucleation and droplet growth processes to the flow, and therefore this calculation can be carried out as a post-processing step.

1.6 Boundary Conditions

The nozzle wall boundary conditions are straight-forward: no-slip for velocity and either adiabatic or isothermal. In the isothermal case, either the wall temperature can be set to a single value, or a list of wall temperatures at each grid point may be read in.

The inflow boundary condition is more difficult to set up because of vibrational excitation and the high pressure equation of state. The original version of the code used a characteristic-based inflow boundary condition, however it was difficult to obtain the correct stagnation conditions from the inflow plane into the solution domain.

The current version of the code uses a different approach. The total mass flux across a plane of the interior solution is computed, and the inflow velocity is then set so that the inflow has the correct mass flux; the temperature is then adjusted to maintain the specified stagnation enthalpy of the inflow based on assuming that the density varies only weakly as the gas accelerates. This approach then assures that the inflow conditions are correctly specified and that once the nozzle has choked, the mass flow through the inlet is correct. Extensive testing has validated this approach.

2. Nozzle Simulation Tools

The nozzle analysis tools comprise a computational fluid dynamics solver, codes to aid in determining reservoir conditions, a grid generator, and a code to generate plotting files. In this section, code input/output files are discussed, specific compilation instructions for various computers are given, and the steps for obtaining a solution are presented. In the next section several test cases are documented, with specific instructions to set up the codes, obtain reservoir conditions, run cases, and plot results. SI units are used throughout the codes except where noted. The nozzle analysis codes are:

- `aptu_nozzle.f` CFD code for hypersonic nozzle simulation and analysis. The code is set up to run either N_2, Air (5-species or non-reacting), and carbon-hydrogen-oxygen-nitrogen (CHON) combustion product gas mixtures.
- `n2eq.f, aireq.f, choneq.f` Codes to compute the equilibrium gas composition of nitrogen, air, or CHON given a pressure, temperature and elemental mixture. It uses the same curve-fits for the equilibrium constants as the nozzle code, so that the inflow conditions are compatible. The codes use the excluded volume model for high pressure flows used in the CFD code.
- `read.f` Code to read solution files and write out a Tecplot-compatible data file. For the hydrocarbon combustion product simulations, the saturation temperature and nucleation rate are written out.
- `nozgrid.f` This is a simple elliptical grid generator that takes as input a list of x and y nozzle coordinates and generates a nozzle grid. It adds a constant-area section upstream of the converging section, which works better with the inflow boundary condition.

2.1 Grid Generation

To run the CFD code, you first need to generate a grid using `nozgrid.f` or some other grid generator. `nozgrid.f` expects to read a datafile of x and y data (in inches) of the form given in the file `coords.dat`.* Note that the last line of `coords.dat` corresponds to `-999 -999 -999` to signify the end of the data. It is up to the user to provide a reasonable axial variation of the x-y pairs. In general, the nozzle coordinates should be clustered near the throat and in the regions of rapid slope change. A good nozzle grid will typically have about 500 x-y pairs; the number of grid points typically increases with the operating Mach number of the nozzle.

Within `nozgrid.f`, the wall normal spacing can be adjusted by the user by changing the value of `rk`; larger values increase the grid stretching (decrease the wall-normal spacing). The elliptical grid generator is very sensitive to the value of this parameter, so only small changes should be made to `rk`.

Running `nozgrid.f` results in the file `nozzle.grid` of the correct form for use with the CFD solver. Note that the CFD method uses dummy (or ghost) cells to enforce the boundary conditions. `nozgrid.f` generates a grid that includes the dummy cells and writes the grid according to the following format:

```
      open(21,file='nozzle.grid')
      write(21,*) il
      write(21,*) jl
      write(21,*) ((xg(j,i),i=1,il+1),j=1,jl+1),
     .            ((yg(j,i),i=1,il+1),j=1,jl+1)
      close(21)
```

Where `il` and `jl` are the number of finite volume cells (including the dummy cells) in the axial and normal directions, respectively. If a different grid generator is used, the grid must be written in this form including the dummy cells. To form the dummy cells, simply mirror the grid points about the inflow/outflow planes, nozzle centerline, and nozzle wall; the values of `il` and `jl` then to need to be increased appropriately.

* The format is a list of three real numbers per line: x, y, and another quantity that is not used. The units are inches for x and y; a straight-forward edit of `nozgrid.f` will allow conversion from other units.

Please note that the grid generated by `nozgrid.f` must be checked by the user to verify that the grid generation has been performed correctly. Given the wide range of hypersonic nozzle geometries, it is very difficult to develop a grid generation code that will work well for all cases.

2.2 Stagnation (Reservoir) Conditions

Codes are provided in the distribution to allow the user to compute the appropriate stagnation conditions for high-pressure mixtures of N_2, Air, and CHON mixtures. These codes (`n2eq.f`, `aireq.f`, and `choneq.f`) determine the equilibrium composition of the gas by solving the polynomial system of equations resulting from the definition of the equilibrium constants. The user must input the elemental composition of the gas for the CHON mixture; this is obtained from the known fuel composition and fuel/air mixture of the combustion products. The solver then determines the equilibrium composition at the input stagnation pressure and temperature. The output from these codes is then used in the input file for the nozzle analysis code.*

The CHON equilibrium solver uses the LAPACK linear algebra routines, and therefore it must be linked to them at compile. For Portland Group Fortran90 use:

```
pgf90 -r8 -fast -o choneq.x choneq.f -llapack -lblas
```

2.3 Code Compile

To compile the CFD code, you must use MPI and Fortran90, and you want to use whatever optimization is available. Double precision must be specified during the compile. The nozzle analysis code has been tested on a variety of machines and compilers:

For Portland Group (PGI) Fortran on a Linux (Pentium or Opteron) cluster:

```
mpif90 -r8 -fast -o aptu_nozzle.x aptu_nozzle.f
```

For Intel Fortran on a Linux cluster:

* Note that because the equilibrium codes solve for the roots of the polynomials in mixture mole fraction, sometimes spurious roots of these equations may be found. The codes have been tested at a range of conditions, but the user must be aware of this issue; improved initial estimates of the species mole fractions will overcome this problem.

```
mpif90 -r8 -axW -o aptu_nozzle.x aptu_nozzle.f
```

The `-axW` flag can be adjusted depending on the processor on the machine.

For Intel Fortran on an SGI Altix:

```
ifort -r8 -O3 -o aptu_nozzle.x aptu_nozzle.f -lmpi
```

More aggressive optimization should work too.

2.4 Input File

Next, you need to specify the run conditions; this is done in the file `nozzle.inp`. First, let us focus on setting the reservoir conditions in the input file. Use the example input files from the nitrogen, air, and CHON mixtures examples as a starting point. This will reduce the difficulty of providing the necessary species data for each mixture. The variables in `nozzle.inp` that need to be changed are:

`density:` stagnation density in kg/m^3

`Tin, Tvin:` stagnation temperature, vibrational temperature in K

`cs:` inflow **mass fraction** for each species in the order given in the relevant input file (N_2, air, or CHON).

`iwall:` for an adiabatic nozzle wall (seldom relevant), set `iwall = 1`. For a single uniform temperature isothermal nozzle wall set `iwall = 2`, and set `twall` to the appropriate wall temperature in Kelvins. And for a variable temperature isothermal wall, set `iwall = 3`. In the latter case, the wall temperature values (in Kelvins) must be specified in an external file (`twall.dat` in the example cases). These values correspond to the wall temperature at the center of each of the `il` axial-direction finite volumes. The name of this file is specified in the 6$^{\text{th}}$ line of the input file.

`iturb:` switch to turn on/off the Spalart-Allmaras turbulence model. Set `iturb = 1` for a turbulent boundary layer.

2.5 Code Execution

Now it is (finally) time to run the nozzle simulation code. The CFL number sequence and number of time steps are the main variables that must be adjusted during a run. These should be set so as to obtain a converged (steady-state) solution in the minimum number

of time steps. In general, the solution progresses by first reaching a steady-state in the converging section by establishing the choked condition at the throat. Then the solution propagates down the supersonic portion of the nozzle until the solution converges and the initial condition is swept out of the computational domain. This process mimics the physical nozzle starting process in a hypersonic facility. It is fundamentally impossible for the nozzle to be converged until a physically meaningful time has elapsed in the simulation. Therefore it is advantageous to run at the maximum stable time step to reduce the number of time steps to obtain a fully converged (started) nozzle flow.

However, because the problem is highly non-linear, it is not possible to simply start the simulation at an arbitrarily large time step. Rather, the time step must be ramped up slowly so that the solution remains physically meaningful (non-negative density and temperature) during the initial transient. Unfortunately, because each problem is different, the optimal ramping schedule varies with the reservoir conditions, the grid size, and most importantly, the degree of grid stretching. The examples give typical time step ramping functions for several cases.

You shouldn't need to change most of the switches in the `input.inp` file; you will need to change the following variables:

`istop:` number of time steps to be run before the code stops and writes the solution. The number of time steps required depends on the problem. As mentioned above, factors that affect the number of iterations required to fully start the nozzle flow include the grid size, the physical length of the nozzle, the area ratio of the nozzle, and the rate at which the user ramps up the CFL number (while maintaining a physically meaningful solution). For the cases I have run, the calculation converges in about 500 time steps for a short, low Mach number nozzle to 3000 for a long high Mach number nozzle.

`iconr:` switch to start from previous solution (`iconr = 1`) or from a guessed initial solution (`iconr` $\neq$ `1`). Generally set `iconr = -1` and run; if the solution needs to be run farther, set `iconr = 1` and run some more.

Additional information on the run-time parameters is given in `nozzle.inp`.

The ramping function for the CFL number sequence is given following the species information in `input.inp`. This list tells how the code ramps the CFL number from the initial value given as the variable `cfl`. Every 20 time steps the code takes the next value of CFL in the sequence up to the point where it detects a negative value of CFL; then it continues with the last non-negative value until the run is complete. Thus in a typical case, the calculation starts at `0.1` and after 20 iterations, goes to `1.0`, `5.0`, up to `1000.0` or usually more. The exact sequence depends on the problem and may differ from what I have given you. If you can't run at `cfl` $\geq$ `100` after several hundred iterations, there is probably something wrong.

From experience, the maximum CFL number is usually limited by a physical time step size – typically a CFL that corresponds to about a time step of several microseconds. Thus, for highly stretched grids with very small wall-normal spacing, the maximum CFL number can be large (as much as 10^6), while for lower Reynolds number flows, the maximum CFL may reach only several thousand.

As the calculation is running, several things should be monitored: The solution residual (the second column of numbers written to the screen and to `converge.m`) should generally trend downwards, though it may go through periods where it increases. In most cases, the residual will drop by 8 or more orders of magnitude as the solution converges. If the residual begins to increase significantly over a large number of time steps, the solution is probably diverging and the CFL is too large or has been increased too rapidly; restart the calculation with a less aggressive CFL ramping function. In addition to the residual, the time step size should be monitored as well (the third column of numbers written to the screen). Again, unless the calculation has progressed though a physically meaningful time, it cannot be converged to a steady-state. A simple estimate of the nozzle starting time will give guidance as to the number of time steps required to computationally start the nozzle.

In some cases, the residual will not decrease by a huge (8+ order of magnitude) factor due to oscillations induced by the turbulence model source terms. Experience shows that these oscillations do not affect the solution quality, and therefore the residual does not

give a full picture of whether the solution is converged. In practice, the solution at one or two points will oscillate between two states, resulting in a relatively large residual but no adverse effect to the overall solution. That is why it is important to also monitor the total flow time to determine whether the nozzle has started based on physical arguments. The new code has been significantly improved in this regard; in all test cases performed to date, this problem has not occurred.

Finally, it is now time to run the calculation! Depending on how your system does MPI, you should be able to do something like (for 8 processors):

```
mpirun -np 8 aptu_nozzle.x
```

This will result in an output file for each processor of the name `nozzle.flow.*`, which can be read by `read.f`.

2.6 Post-processing and Plotting

The code `read.f` reads the output files from the CFD code and writes out data files in Tecplot compatible format for plotting purposes. The code reads `nozzle.inp` to determine the gas model (`iset`) and writes data appropriate for the gas model. Again, the data are written in SI units.

To post-process the data, compile the code using a Fortran90 compiler. For Portland Group Fortran90:

```
pgf90 -r8 -fast -o read.x read.f
```

Then simply run the executable. The user will need to input the number of processors used in the simulation; all other information is obtained from the data files.

For the CHON mixture, `read.f` computes the saturation temperature and nucleation rate as described above. If additional variables need to be plotted, the code may be modified to compute them.

3. Examples

Included with the distribution are 4 example problems: N_2, Air, and two CHON hypersonic nozzle flows. This section describes how the input data are generated for each case, and details of each run are provided.

The files are located in separate directories: `N2example`, `AIRexample`, `HTTexample`, and `CHONexample`. In each directory, there is a copy of the grid files, input files, and solution files. The same version of `aptu_nozzle.f` and `read.f` was used to compute each flow field.

3.1 Nitrogen Example: AEDC Tunnel 9 Nozzle

In the code distribution, this example problem is provided in the directory `N2example`. In this case, a 301×121 (including dummy cells) grid was provided by Greg Molvik, along with the wall temperature at various locations along the nozzle. A small code (`twall.f`) was written to linearly interpolate the wall temperature for each of the wall grid cells.

For this case, the stagnation conditions are: $T_o = 3027\,\mathrm{R} = 1681.5\,\mathrm{K}$ and $p_o = 21,540\,\mathrm{psi} = 148.46\,\mathrm{MPa}$. Using $b_o/\bar{M} = 0.001120\,\mathrm{m}^3/\mathrm{kg}$, the density is computed to be $223.145\,\mathrm{kg/m}^3$. These are the values used in the input file (`nozzle.inp`).

Because the nozzle is very long ($> 12\,\mathrm{m}$) with a small throat, the physical nozzle start time is expected to be significant. The nominal nozzle exit Mach number is 14, which for the stagnation temperature, gives a test section axial velocity of $1850\,\mathrm{m/s}$. If we assume that the average speed of the gas in the nozzle is half of this value, the nozzle starting time is approximately $12/925 = 0.013\,\mathrm{sec}$. Therefore, our calculation must be run for at least this physical time before we can expect the nozzle to be started.*

In the input file, there is a fairly aggressive ramping function provided, which gives a final CFL number of 10^6, resulting in a time step of approximately $10\,\mu\mathrm{sec} = 10^{-5}\mathrm{sec}$. Therefore, once we have reached this large CFL, we expect the solution to require $0.013/10^{-5} = 1300$ time steps. With a factor of two safety factor, we run the solution

* The time integration is not perfectly time accurate, but the time step can be used to estimate the elapsed time of the simulation. In practice, it has been found that it does provide useful guidance to assess the progress of the solution.

for 2500 time steps.

For the example provided, we ran on 4 dual-processor Pentium Xeon 2.4 GHz machines connected with Myrinet. The compiler was Portland Group Version 4.1. The total run time was 550 seconds for 2500 time steps.

The output files are provided in the distribution, including the output from `read.f`. A plot of the L_2 norm of the solution residual is provided, which is from the file `converge.m`. Note the 14 order of magnitude decrease in the residual to machine zero at about 2000 iterations. In this case, the estimated number of time steps was reasonable though a little optimistic.

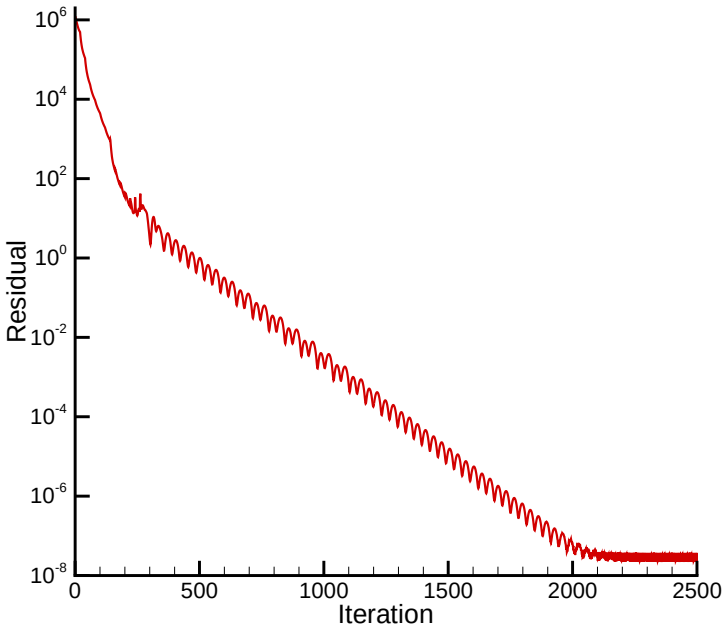

Figure 1. Solution residual for N_2 test case.

3.2 Air Example: Generic 5-Species Air Nozzle Flow

For this calculation, we use the same Tunnel 9 nozzle grid that was used for the previous example. We choose typical stagnation conditions of $p_o = 5496\,\text{psi}$ and total enthalpy $h_o = 2.9492 \times 10^7\,\text{ft}^2/\text{sec}^2$. These conditions are inserted into `aireq.f`, which is compiled and run. A guess for the stagnation temperature is required – in this case something like 3000 K works. The code then writes the computed stagnation conditions to the screen, which are then copied into `nozzle.inp`.

For this case, we ran a constant temperature cold wall by setting `iwall = 2`, and `twall = 300.0`. In addition, `iset` must be changed to `2` to use the 5-species air model. The CFD code is then run; in this case, we ran on 8 dual-processor Xeons with a total execution time of 547 sec. The increase in cost is due to the 5-species air model, which involves the solution of more equations and a more complex chemical kinetics model.

For this case due to the differences in the stagnation conditions and wall temperature distribution, the time step that corresponds to a CFL of 10^6 is $22\,\mu$sec. Therefore, for this run the maximum CFL was reduced by a factor of two. In this case, the solutions does not converge as well as the N_2 example, with only a total residual reduction of 8 orders of magnitude. This does not necessarily mean that the solution has not reached convergence; in this case the chemistry model or the turbulence model is causing small levels of switching between solutions. This can be verified by restarting the simulation and running for a meaningful number of time steps (say 200) and then comparing results.

3.3 CHON Example: NASA Langley 8′HTT Nozzle

To illustrate and validate the water condensation model, we simulated a case from the work of Erickson *et al.*[14] for the NASA Langley 8′ High Temperature Tunnel nozzle. Their approach is quasi-one-dimensional, so we cannot reproduce their results exactly. However, we generated a nozzle grid from the information provided in their report. Table 2 gives x-y coordinates of the diverging section of the nozzle at 8 locations; we fitted spline curves through these points, and then merged the converging section of the Tunnel 9 nozzle to that geometry. Then ran `nozgrid.f` to produce a 373×160 grid.

The conditions for the run are provided in Ref. 14; we chose to consider Case 1, which has $T_o = 1900\,\mathrm{K}$ and $p_o = 5.0\,\mathrm{MPa}$. This case has methane-air equivalence ratio of 0.798. To obtain the relevant mixture, we considered the stoichiometric reaction:

$$CH_4 + 2O_2 \rightarrow 2H_2O + CO_2$$

For an equivalence ratio of 0.798 in air, we then have:

$$0.798CH_4 + 2O_2 + \frac{0.79}{0.21}2N_2 \rightarrow 0.798 \times 2H_2O + 0.798CO_2 + 0.202 \times 2O_2 + \frac{0.79}{0.21}2N_2$$

which gives a mole fraction of the mixture of: $\gamma_{N_2} = 0.728923$, $\gamma_{H_2O} = 0.154624$, $\gamma_{CO_2} = 0.077312$, and $\gamma_{O_2} = 0.039140$. This mixture is put into `choneq.f`, which is then compiled and run. The equilibrium mixture at the 1900 K, 5 MPa stagnation conditions is then copied into `input.inp` and the variable `iset = 3`. For this case, we do not know the wall temperature distribution, so for simplicity, we use a constant temperature isothermal wall set to 800 K.

This case was run for 1000 iterations at a maximum CFL of 4000. The relatively low CFL number is a result of using a weakly stretched grid due to the lower operating Mach number of the facility. However, this CFL results in a 75.5 μsec time step, which rapidly starts the nozzle. In this case, we were only able to achieve 6 orders of magnitude convergence because of grid quality issues at the point where the constant area section joins the converging section. The run took 1048 seconds on 16 processors.

Figure 2 plots contours of the saturation temperature relative to the local temperature, $T_{sat} - T$; the gas starts to condense in regions where this quantity is positive. The figure

also plots the nucleation rate, J, in particles/m^3 sec. Clearly, J is very high in this region and the gas will condense very rapidly. These results show $T_{\text{sat}} > T$ at the same location as Ref. 14.

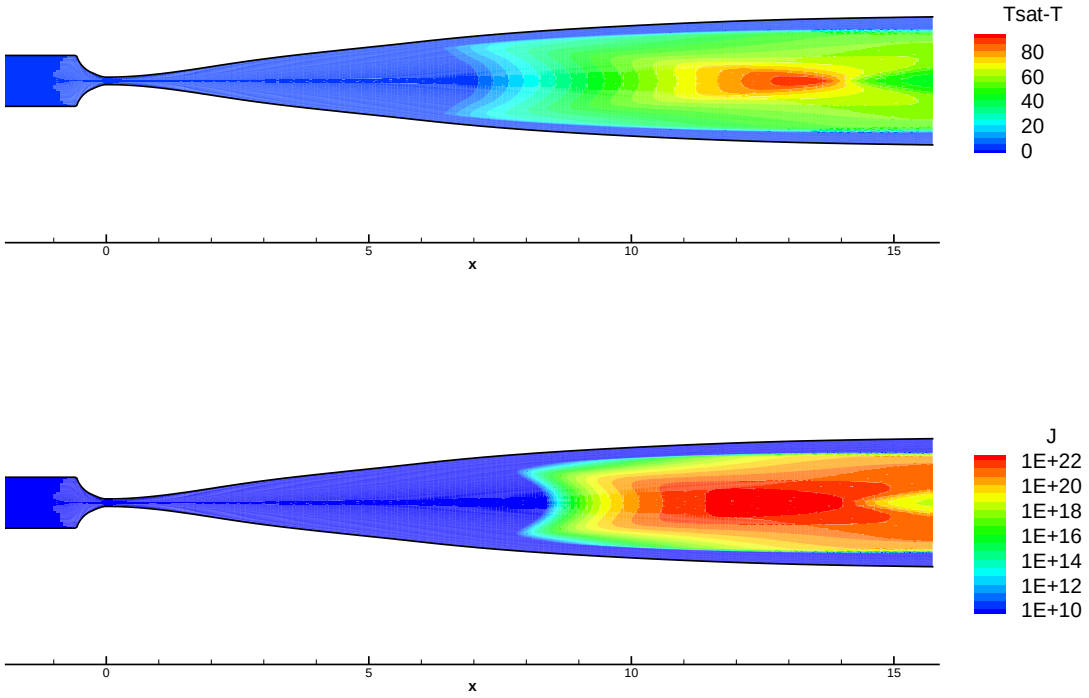

Figure 2. Saturation temperature relative to the local temperature (top) and nucleation rate (bottom).

3.4 CHON Example: Mach 6.5 APTU Nozzle

The final example is for the Mach 6.5 APTU nozzle at conditions provided by Greg Molvik. These correspond to the operating condition of $T_o = 4800\,\mathrm{R}$ and $p_o = 2800\,\mathrm{psi}$. The mole fractions are obtained by running `choneq.f` starting with the gas composition provided. Based on the effective molecular diameters (Table 1) and the gas composition, the value of $b_o/\bar{M}$ is computed to be $0.00076552\,\mathrm{m}^3/\mathrm{kg}$. This gives a density of $25.533\,\mathrm{kg}/\mathrm{m}^3$.

The nozzle contour data file, `coords.dat`, has a poor axial distribution of x-y coordinate pairs. Therefore, a short code was written to read in the data file, and linearly interpolate between the points to yield a better distribution of grid points. This code, `nozcont.f` is provided in the distribution. This code is first run to generate a new coordinates file (`nozcont.dat`), and then the grid generation file `nozgrid.f` is run to generate the grid. For the parameter settings in these codes, this results in a 410×160 grid.

The CFD code was run for 1000 time steps ramping to a maximum CFL of 2×10^5, corresponding to a time step of $18\,\mu$sec. The residual was reduced by 7 orders of magnitude, and the total elapsed physical time was approximately 13 msec, which is sufficient to fully converge the solution for this relatively short, low Mach number nozzle. The maximum residual is located at the corner where the constant area duct suddenly transitions to the converging part of the nozzle. The grid in this region is highly stretched, which likely leads to the high residual. The calculation required 1136 sec on 8 dual-processor Xeons. Figure 3 plots the computed Mach number contours for this case.

Figure 4 plots the solution residual for two grids. The poor result was obtained using the original `coords.dat` file, and the significantly improved convergence plot was obtained using the smoothly varying coordinates. Clearly, using a high quality grid is important in obtaining good convergence properties and a rapid solution. Note that the maximum allowable CFL on the poor grid was just 1000.

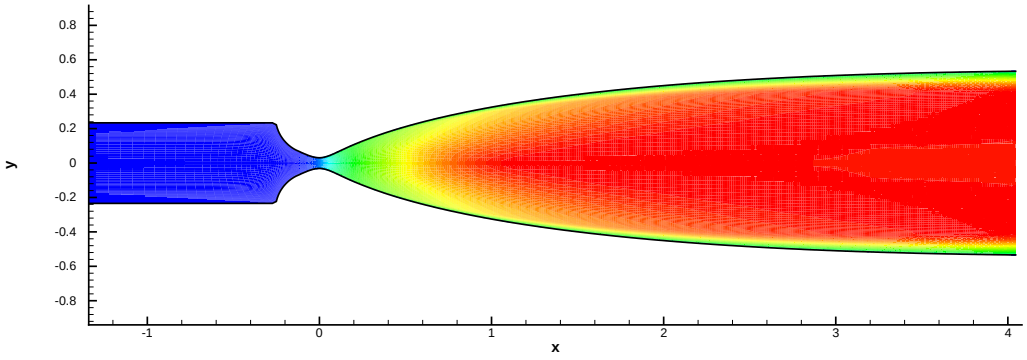

Figure 3. Mach number contours for the Mach 6.5 APTU nozzle example.

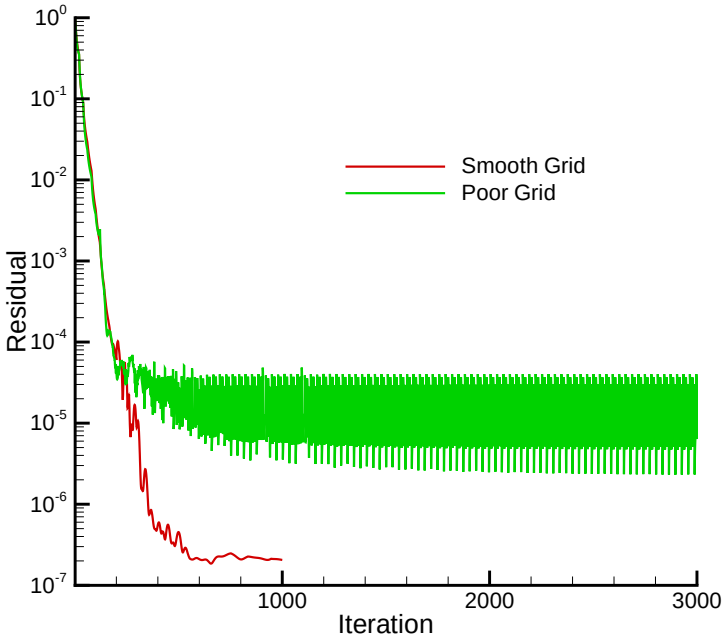

Figure 4. Convergence history for APTU problem on two different grids.

4. Code Structure

The schematic given in Fig. 5 illustrates the structure of the nozzle simulation code. First, several subroutines are called to initialize the MPI variables (`subroutine startup`), the input files are read in (`subroutine readin`), the geometry is read in from the grid file, and the mesh metrics are computed (`subroutine geometry`), and finally the flow field is initialized (`subroutine init`). When the code is starting from no initial solution, `subroutine nozzle_init` is called to initialize the nozzle flow using an approximate quasi-one-dimensional solution for the nozzle geometry; otherwise the solution is read in using `subroutine rdwrt`.

Then the solution begins by running a series of `istop` time steps. This involves computing the inviscid fluxes across each finite volume surface; this is done in `subroutine fluxes`. This subroutine then calls the viscous flux calculation, the chemical source term calculation, and then the Spalart-Allmaras source term calculation. The inflow boundary conditions are then computed in the subroutine `lbc_non`.

At this point, all required information for a time step has been computed. The time step is either done using an explicit time step (`iextst = 1`) or using the implicit data-parallel line-relaxation method[15] (`iextst` $\neq$ `1`). In the latter case, the subroutine `dplr` is called to assemble the Jacobian matrices, factorize the system of equations (`subroutine factor`), and then backward-substitute into the block tri-diagonal system (`subroutine solve`). Details of the linear algebra performed by these routines is given in Ref. 15.

Once the `istop` time steps are complete, the solution is written out using `subroutine rdwrt`, and the simulation ends.

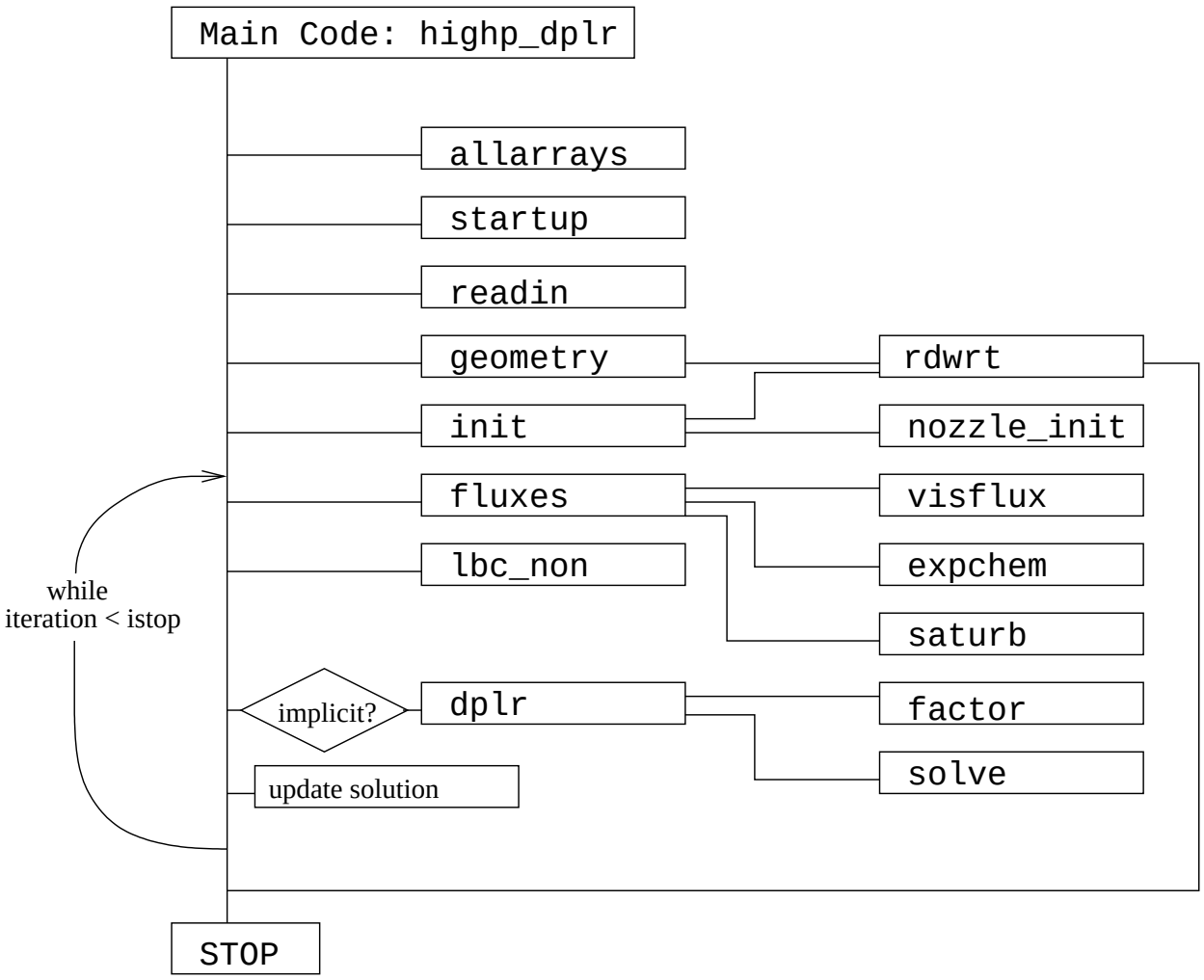

Figure 5. Schematic of `aptu_nozzle.f` code structure.

5. Variable Definitions

The following is a list of the major variables used in the nozzle simulation code. In all cases, the variables are dimensional and use SI units; these are: meters, seconds, Kelvins, Pascals, Joules, kilograms, and kilomoles.

Grid Variables

`x`	axial-direction distance	[m]
`y`	radial-direction distance	[m]
`si`	i-direction surface area of finite volume	[m]
`sj`	j-direction surface area of finite volume	[m]
`volin`	inverse volume of finite volume	[m^{-2}]

Flow Variables

`r`	density	[kg/m^3]
`rhos`	species density	[kg/m^3]
`u`	axial-direction velocity	[m/s]
`v`	radial-direction velocity	[m/s]
`c`	speed of sound	[m/s]
`t`	translational-rotational temperature	[K]
`tv`	vibrational temperature	[K]
`p`	pressure	[Pa]
`re`	total energy per unit volume	[J/m^3]
`rev`	vibrational energy per unit volume	[J/m^3]
`evs`	species vibrational energy per unit mass	[J/kg]
`rmut`	turbulent eddy viscosity	[kg/m s]
`cv`	translational-rotational specific heat	[J/kg K]
`gcon`	mixture gas constant	[J/kg K]
`res`	L2 solution norm of density	[ ]
`twall`	wall temperature	[K]

Thermodynamic/Rate Constants

`hform`	heat of formation	[J/kg]

`thetv`	characteristic temperature of vibration	[K]
`vdg`	vibrational degeneracy	[ ]
`smw`	species molecular weight	[kg/kmole]
`cvs`	species translational-rotational specific heat	[kg/kmole]
`rmu`	mixture viscosity	[kg/m s]
`rkap`	mixture thermal conductivity	[J/m s K]
`gsp`	species gas constant	[J/kg K]
`cf`	forward reaction rate coefficient	[m^3/kmole s]
`aeta`	temperature exponent for forward reaction rate	[ ]
`athet`	exponential constant for forward reaction rate	[K]

6. References

[1] Lordi, J.A., and R.E. Mates, "Nonequilibrium Expansions of High-Enthalpy Flows," Aerospace Research Laboratories, ARL 64-206, Buffalo, NY, Nov. 1964.

[2] Jacobsen, R.T., Stewart, R.B., McCarty, R.D., and Hanley, H.J.M., "Thermophysical Properties of Nitrogen from the Fusion Line to 3500 R (1944 K) for Pressures to 150,000 psia (10342×10^5 N/m^2)," National Bureau of Standards, NBS TN 648, Boulder, CO, Dec. 1973.

[3] Lemmon, E.W., R.T. Jacobsen, S.G. Penoncello, D.G. Friend, "Thermodynamic Properties of Air and Mixtures of Nitrogen, Argon, and Oxygen from 60 to 2000 K at Pressures to 2000 MPa," *J. Phys. Chem. Ref. Data,* Vol. 29, No. 3, 2000, pp. 331-385.

[4] Chase M.W. Jr., "NIST-JANAF Thermochemcial Tables, Fourth Edition," *Journal of Physical and Chemical Reference Data,* Monograph 9, 1998.

[5] Millikan, R.C. and D.R. White, "Systematics of Vibrational Relaxation," *J. Chem. Phys.,* Vol. 39, 1963, pp. 3209-3213.

[6] Camac, M., "CO_2 Relaxation Processes in Shock Waves," *Fundamentals of Phenomena in Hypersonic Flows,* edited by J.G. Hall, Cornell Univ. Press, Ithaca, NY, 1966, pp. 195-215.

[7] Gordon, S., and B.J. McBride, "Thermodynamic Data to 20,000 K for Monatomic Gases," NASA TP-1999-208523, June 1999.

[8] Wilke, C.R., "A Viscosity Equation for Gas Mixtures," *Journal of Chemical Physics,* Vol. 18, pp. 517-519, 1950.

[9] Blottner, F.G., M. Johnson, and M. Ellis, "Chemically Reacting Viscous Flow Program for Multi-Component Gas Mixtures," Report No. SC-RR-70-754, Sandia National Laboratories, Albuquerque, NM, 1971.

[10] Olejniczak, J., and G.V. Candler, "Vibrational Energy Conservation with Vibration-Dissociation Coupling: General Theory and Numerical Studies," *Physics of Fluids,* Vol. 7, No. 7, pp. 1764-1774, July 1995.

[11] Nompelis, I., G.V. Candler, and M.S. Holden, "Effect of Vibrational Nonequilibrium on Hypersonic Double-Cone Experiments," *AIAA Journal,* Vol. 41, No. 11, pp. 2162-

2169, Nov. 2003.

[12] Spalart, P.R., and Allmaras, S.R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1992.

[13] Catris, S., and B. Aupoix, "Density Corrections for Turbulence Models," *Aerospace Science and Technology,* Vol. 4, 2000, pp. 1-11.

[14] Erickson, W.D., G.H. Mall, and R.K. Prabhu, "Finite-Rate Water Condensation in Combustion Heated Wind-Tunnels," NASA TP 2833, 1988.

[15] Wright, M.J., D. Bose, and G.V. Candler, "A Data-Parallel Line Relaxation Method for the Navier-Stokes Equations," *AIAA Journal,* Vol. 36, No. 9, pp. 1603-1609, Sept. 1998.