The Gas Dynamics Functions for MATLAB

J.S. Jacob
Bently Rotor Dynamics Research Corporation

Introduction
Gas dynamics, or the study of compressible fluid flow, encompasses classical fluid mechanics, thermodynamics, and heat transfer. It is therefore more comprehensive and rigorous than simple incompressible fluid dynamics. It is purposed here to introduce the basic concepts of gas dynamics required to model simple, one-dimensional compressible flows and present algorithms for numerical computation.

Gas dynamics describes compressible fluid flow in thermodynamic and fluid dynamic quantities. The equations are normally tabulated for hand computation because they are quite complicated, and the most important one cannot be analytically inverted. These MATLAB functions replace the gas tables and additionally permit automated iterative solutions to more complex problems such as normal shocks and time-variable geometries.

All the difficult aspects of the relations governing compressible fluid flow can be distilled into just one function: the normalized gas dynamics function $\Gamma$. All essential quantities can be related through $\Gamma$. It will be shown that $\Gamma$ can be readily obtained for any point in the flow with the minimum required information about the flow or some very reasonable assumptions.

Other functions are needed for closely related values such as the Mach number and the pressure ratio. The state point flow number $\alpha$ is useful for relating the mass flow rate or linear velocity to the Mach number through parameters such as flow area, pressure and temperature. A precursor to the normalized gas dynamics function is the pressure ratio function $P$. It and its inverse are included because they are occasionally useful, even though $\Gamma$ is just $P$ divided by its maximum value $P^*$ such that $\Gamma$ has a range of 0 to 1.

Derivation of the Normalized Gas Dynamics Function

To develop the general gas dynamics equation, the following five relations are needed:

1. Continuity (conservation of mass) confined to steady-state flows, expressed as

$$m = \rho_1 V_1 A_1 = \rho_2 V_2 A_2 = \text{const}$$  \hspace{1cm} (1)
2. Total energy, consisting of enthalpy and kinetic energy, but ignoring gravitational potential energy. We will assume no energy additions to the flow.

\[
\Delta E = du + pdv + vdp + VdV = 0
\]  

(2)

3. The Gibbs equation, Tds equation, or the basic gas dynamics identity:

\[
Tds = du + pdv
\]  

(3)

4. The polytrope relating states to their isentropic stagnation reference states:

\[
\frac{p}{\rho^k} = \text{const}; \quad \rho = \rho_i \left( \frac{p}{p_i} \right)^{\frac{k}{k - 1}}
\]  

(4)

5. The ideal gas equation of state. Using this equation of state only limits us from considering semi-compressible fluids, mixed-state fluids, and cases of extreme pressure.

\[
p = \rho RT
\]  

(5)
To summarize the assumptions and restrictions, only one-dimensional, steady-state flow of an ideal gas is considered. That is much less restrictive than it sounds, for the following reasons. “Steady state” is relative to the speed of changes within the body of the fluid, which one realizes is the speed of sound. For short flows, time variations in flow must then be extremely rapid before the flow can no longer be considered static. The ideal gas equation of state models real gasses to a high degree of accuracy, up to the point at which one contemplates changes of state, e.g. condensation to the liquid state. Because of these assumptions, one judges Eqs. 1 – 5 to be perfectly applicable to gas bearings for rotating machinery.

The derivation proceeds by combining Eqs. 2 and 3 and integrating between the static state and the isentropic reference state to obtain $V$ in terms of the reference state. Eq. 4 is differentiated and used to relate $dp$ and $d\rho$. The resulting expression for $V$ is substituted into Eq. 1. Eq. 5 is used to eliminate density in favor of temperature. The result is

$$
\left( \frac{T_2}{T_1} \right)^{\frac{1}{2}} \left( \frac{p_{12}}{p_{1}} \right)^{\frac{1}{2}} \left( \frac{A_2}{A_1} \right) \left[ \left( \frac{p_1}{p_{12}} \right)^{\frac{k+1}{k}} - \left( \frac{p_1}{p_{1}} \right)^{\frac{k+1}{k}} \right]^{\frac{1}{2}} = \left[ \left( \frac{p_2}{p_{2}} \right)^{\frac{k}{k}} - \left( \frac{p_2}{p_{12}} \right)^{\frac{k+1}{k}} \right]^{\frac{1}{2}}
$$

Eq. 6 relates an expression of the static pressure at point 1 to the same at point 2. That expression of pressure is called the pressure ratio function $P$. It is useful to normalize the pressure ratio function by differentiating to determine its maximum value, and dividing by the maximum value. At the same time, this procedure reveals the critical pressure ratio $\frac{P}{p_{12}}$ which maximizes $P$. It is convenient to define the pressure ratio $R$ to be the ratio of static pressure at a point to the isentropic reference pressure:

$$
R = \frac{P}{p_{12}}
$$

The critical pressure ratio $R_*$ is found to be

$$
R_* = \left( \frac{2}{k+1} \right)^{\frac{k}{k-1}}
$$

which results in the maximum value of the pressure ratio function

$$
P_* = \left[ \left( \frac{2}{k+1} \right)^{\frac{k}{k-1}} \left( \frac{k-1}{k+1} \right) \right]^{\frac{1}{2}}
$$

The normalized gas dynamics function is $P$ divided by its maximum value:
\[ \Gamma = \frac{P}{P_*} = \left[ \frac{R^{\frac{2k}{k+1}} - R^{\frac{k+1}{k}}}{\left( \frac{2}{k+1} \right)^{\frac{2k-1}{(k-1)(k+1)}}} \right]^{\frac{1}{2}}. \]  

Equation 6 is now more succinctly written as

\[ \left( \frac{T_{r2}}{T_{r1}} \right)^{\frac{k}{2}} \left( \frac{p_{r1}}{p_{r2}} \right) \left( \frac{A_1}{A_2} \right) \Gamma_1 = \Gamma_2 \]  

This equation provides the solution to any one-dimensional compressible flow problem. Note that the use of isentropic stagnation states, also known as reference states, does not restrict the applicability of the gas dynamics function to isentropic flows. Every point along the flow has a corresponding conceptual reference state, whether the flow is isentropic (reversible) or not. Equation 11 admits one unknown value or one unknown ratio, but additional information can be obtained if assumptions about the process linking the two states can be reasonably made.

There are several ways of finding \(\Gamma\) for a point along the flow. If the static pressure and the isentropic reference pressure are known at the point, then \(\Gamma\) can be calculated immediately. If the reference pressure is not otherwise known, the pressure ratio can be found by the relation

\[ \frac{p}{p_*} = R = \left( 1 + \frac{k-1}{2} M^2 \right)^{\frac{k}{k-1}} \]  

which requires only that the Mach number to be known. The inverse of this relation is also useful for future reference:

\[ M = \left( \frac{2}{k-1} \left( R^{\frac{1}{k-1}} - 1 \right) \right)^{\frac{1}{2}}. \]  

\(\Gamma\) may also be found directly from the Mach number:

\[ \Gamma = \frac{\Gamma}{\left[ \left( \frac{2}{k+1} \right)^{\frac{2}{k+1}} \left( 1 + \frac{k-1}{2} M^2 \right)^{\frac{k+1}{2(k-1)}} \right]} \]  

The Mach number in terms of average linear velocity is simply

\[ M = \frac{V}{\sqrt{kRT}} \]
One notes that this involves temperature $T$. It becomes apparent that $\Gamma$ represents the full state of
the fluid at a point in the flow, requiring a certain amount of information before a specific problem
may be solved.

Incompressible fluid dynamics is the comparatively simple process of applying geometry to a
pressure difference to determine the flow rate, or applying the flow rate to determine the pressure
drop. The knowledge of an unchanging density simplifies the problem and is often taken for
granted. Compressible flow requires greater rigor, since density is variable, requiring that the
complete state of the fluid be known at a point in the flow.

In addition, the type of thermodynamic process relating two adjacent states in the flow must be
known or assumed. This could be an isothermal process in which the addition or subtraction of
heat through the boundary maintains a constant temperature throughout the process. It could also
be an insulated or adiabatic process in which exchange of heat is excluded. One of the simplest
processes to analyze is an isentropic process, in which no irreversible changes occur. This excludes
friction, heat transfer, and standing shock waves. Knowledge of the process provides the
additional information necessary for the solution of the basic flow problem. For example, in an
isentropic (frictionless) process, the isentropic reference state happens to be constant for every
point along the flow. Thus,

\[ T_{i} = T_{2} ; \quad p_{i} = p_{2} \]  \hspace{1cm} (16)

and Eq. 11 reduces to the following:

\[ \frac{A_{1}}{A_{2}} \Gamma_{i} = \Gamma_{2} . \] \hspace{1cm} (17)

After a few more useful relations are introduced, some examples of solved problems will be given.

The static flow number provides a useful link between the Mach number and the mass flow rate.
Substituting $\dot{m} = \rho AV$ into Eq. 15 results in

\[ \left( \frac{\dot{m}}{Ap} \right) \sqrt{RT} = \sqrt{kM} \] \hspace{1cm} (18)

in which the left hand side is termed the static flow number. The total flow number (or the
isentropic reference flow number) is given the symbol $\alpha$ and is related to $M$ by

\[ \alpha \equiv \left( \frac{\dot{m}}{Ap} \right) \sqrt{RT} = \frac{\sqrt{kM}}{1 + \frac{k-1}{2} M^{2}} \left( \frac{1}{k+1/2(k-1)} \right) \] \hspace{1cm} (19)

If one defines the critical total flow number $\alpha_{*}$ to be the value of $\alpha$ for which $M = 1$, then one can
show that $\Gamma$ relates to $\alpha$ according to
\[ \Gamma \equiv \frac{P}{P_*} = \frac{\alpha}{\alpha_*} = \frac{\dot{m}}{\dot{m}_*} \] (20)

**Example: Static Pressure at a Point in the Flow**

Given isentropic reference conditions \((p_t, T_t)\), the mass flow rate (same for all points in the flow) and the area at some point, use \(\Gamma\) to calculate the static pressure at that point.

Solution: First, use Eq. 19 to compute \(\alpha\):

\[ \alpha = \left( \frac{\dot{m}}{Ap_t} \right) \sqrt{RT_t} \] (21)

Now use the right-hand side of Eq. 19 to compute \(\alpha_*\) (i.e. \(\alpha\) at \(M = 1\)):

\[ \alpha_* = \frac{\sqrt{k}}{\left(1 + \frac{k-1}{2} \frac{k-1}{k+1} \right)} \] (22)

Compute \(\Gamma\) from Eq. 20:

\[ \Gamma = \frac{\alpha}{\alpha_*} \] (23)

Now that \(\Gamma\) is known at the point of interest, everything is known about the flow at that point, and any required quantity can be found. To find the static pressure \(p\), find the pressure ratio \(R\) from the numerical inverse of \(\Gamma\) (see MATLAB function invgamma.m), select between the subsonic and supersonic solutions, then multiply by the reference pressure \(p_t\). This example calculation is the method used to determine a pressure distribution in a flow with a known area profile. Such a pressure distribution (pressure profile) is integrated to determine net forces against flow boundaries.

**Example: Choked Flow**

The flow at one point has a known area, mass flow rate and reference conditions. The flow area narrows somewhat downstream from this point. Will the flow become choked?

Use the procedure outlined in previous example to compute \(\Gamma\) for the known point. If the narrowed flow area is known, use the following, obtained from Eq. 11, to compute \(\Gamma\) at the narrowed point given isentropic flow between the two points:
If $\Gamma_2$ is less than 1, the flow is subsonic at the new point. If the result is greater than 1, this is not a valid value of $\Gamma$, and the isentropic assumption is invalid. It may be assumed that the flow is choked, and that the value of $\Gamma_2$ is equal to 1. One uses that information to determine other correct information about the flow.

If the downstream area is not known, one can determine the throat area that would result in choked flow. This is a useful design exercise. Assume a value of 1 for $\Gamma_2$, and compute $A_2$ from Eq. 24.

**Example: Isentropic Flow with Area Change**

The term “isentropic” is usually used to mean specifically an adiabatic (no heat transfer) reversible (no entropy change) process. Reversibility implies that there is no viscous friction. This is not a bad approximation for many engineering applications, since the viscosity of most gases is very small. The common reference to “wind resistance” does not belie that fact, since that refers not to friction but to aerodynamic drag, a different phenomenon altogether. For short, internal flows, a frictionless assumption results in fair approximations.

Given two points in a one-dimensional flow labeled 1 and 2, the area at each point ($A_1$ and $A_2$), the pressure at each point ($p_1$, $p_2$), determine the flow rate.

Using the isentropic assumption, the isentropic reference state remains constant throughout the flow. Thus,

$$\left(\frac{T_{12}}{T_{11}}\right) = 1; \quad \left(\frac{p_{11}}{p_{12}}\right) = 1$$

and Eq. 11 reduces to

$$\left(\frac{A_1}{A_2}\right) \Gamma_1 = \Gamma_2$$

(26)

The ratio of $\Gamma_2$ to $\Gamma_1$ is now known, but how does this tell us the state at either point? The one value needed that is unknown is the isentropic reference pressure, $p_t$. The task is therefore to determine a value of $p_t$ which, with the given values $p_1$ and $p_2$, results in a ratio $\Gamma_2/\Gamma_1$ given by $A_1/A_2$. To do this, consider the normalized gas dynamics equation

$$\Gamma = \left[ \frac{\frac{R^{\frac{2}{k}}}{k+1} - \frac{R^{\frac{k+1}{k}}}{k+1}}{2 \left(\frac{k-1}{k+1}\right)^{\frac{2}{k-1}}} \right]^{\frac{1}{2}}$$

(27)
and therewith define the number

$$\gamma = \left( \frac{\Gamma_1}{\Gamma_2} \right)^2 = \frac{R_1^a - R_1^b}{R_2^a - R_2^b}$$  \hspace{1cm} (28)$$

in which $a \equiv \gamma / \kappa$; $b \equiv k+\gamma / \kappa$. If given the pressure at points 1 and 2 along a one-dimensional compressible flow path with area change, the pressure ratios $R_1$ and $R_2$ are defined using the isentropic reference pressure (stagnation pressure):

$$R_1 = \frac{p_1}{p_t} \hspace{1cm} R_2 = \frac{p_2}{p_t}.$$  \hspace{1cm} (29)$$

For convenience, let

$$r \equiv \frac{R_1}{R_2} = \frac{p_1}{p_2}$$  \hspace{1cm} (30)$$

so that

$$R_1 = r R_2.$$  \hspace{1cm} (31)$$

The task is to determine $p_t$, given the ratio $\Gamma_1 / \Gamma_2$ and the values of $p_1$ and $p_2$. It can be done by substituting Eq. 31 into Eq. 28 and solving numerically for the value $R_2$. Both $R_1$ and $p_t$ follow directly afterwards. A MATLAB function has been devised to solve the equation

$$\gamma = \left( \frac{\Gamma_1}{\Gamma_2} \right)^2 = \frac{r^a R_2^a - r^b R_2^b}{R_2^a - R_2^b}$$  \hspace{1cm} (32)$$

in which $r$ is given by Eq. 30, $\gamma$ is defined in Eq. 28, and $R_2$ is sought. The numerical solution is limited to values of $r$ and $\gamma$ between 0.02 and 50. It should also be noted that for values of $r$ or $\gamma$ close to 1 or for values resulting in $R_2$ close to 0 or 1, solutions are difficult to obtain and convergence is not assured. For $r = 1$ or $\gamma = 1$, the solution is undefined. However, this situation corresponds to a trivial physical interpretation. Once Eq. 32 is solved, the resulting value of $R_2$ consistent with the given constraints is used to determine $p_t$. (See the MATLAB functions gratio.m and g2ratio.m. Also, the script grsol3.m is useful for visualizing the form of Eq. 32.)

**Example: Flow Through An Orifice**

Given an orifice of area $A$, supply pressure $p_s$ and downstream static pressure $p$, what is the mass flow rate through the orifice? Begin by assuming that the flow process from the stagnant supply to the orifice throat is isentropic. Then, $p_s$ is identified as the isentropic stagnation reference pressure $p_t$, and $R$ is the ratio $p/p_t = p/p_s$. Compute $\Gamma$ based on $R$, but require that the flow be sonic or subsonic (i.e. it cannot be supersonic, since the flow up to this point converges monotonically). So, if $R$ is less than $R_*$ (see Eq. 8), set $R = R_*$ and take $\Gamma$ to be unity. With $\Gamma$ known, use Eqs. 21 and 23 to obtain
\[ \alpha \Gamma(R) = \frac{m}{A p_s \sqrt{RT_s}} \]  

(33)

recalling that \( \alpha^* \) is a function only of \( k \) (see Eq. 22) and that “plain R” is the specific gas constant. Now solve for the mass flow rate:

\[ m = \frac{A p_s}{\sqrt{RT_s}} \alpha \Gamma(R) \]  

(34)

However, the process is not isentropic (frictionless), and the reference pressure at the throat must actually be less than the upstream reference pressure:

\[ p_i = C_d p_s \]  

(35)

where \( C_d \) is the orifice discharge loss coefficient, a value less than unity and determined empirically for various orifices (normally 0.4 – 0.8). Now, the mass flow rate is given by

\[ m = \frac{AC_d p_s}{\sqrt{RT_s}} \alpha \Gamma(R) \]  

(36)

It can be shown analytically that this identical to the equation used in Jacob, J. S., D. E. Bently and J. J. Yu, 2001, “A hydrostatic bearing with compressible fluid for broad application,” Proc. ASME Turbo Expo, New Orleans, LA, June 4-7, paper 2001-GT-0250:

\[
\dot{m} = \begin{cases} 
C_{d_0} A p_s \left[ \frac{2k}{(k-1)RT} \left( \frac{p}{p_s} \right)^{\frac{k}{k-1}} - \left( \frac{p}{p_s} \right)^{\frac{k+1}{k-1}} \right]^{\frac{1}{2}} & \text{for } \frac{p}{p_s} > R_s \\
C_{d_0} A p_s \left[ \frac{2k}{(k-1)RT} \left( \left( R_s \right)^{\frac{k}{k-1}} - \left( R_s \right)^{\frac{k+1}{k-1}} \right) \right]^{\frac{1}{2}} & \text{for } \frac{p}{p_s} \leq R_s 
\end{cases} 
\]  

(37)

\[ R_s = \left( \frac{\alpha^*}{\alpha_k} \right)^{\frac{k}{k-1}} \]

The advantages of using the normalized gas dynamics function are apparent in the comparatively succinct form of Eq. 36.
A MATLAB Algorithm for the Inverse Normalized Gas Dynamics Function

A method of robustly iterating to a solution of the inverse-$\Gamma$ problem is based on a method of linearizing a non-linear feedback control system. Full use is made of a priori knowledge of the system’s range, its double-valued domain, and the limit values.

The first aspect of the inverse-$\Gamma$ problem is its double-valued nature (see Fig. 1). For each value of $\Gamma$, the inverse-$\Gamma$ function must return two values. For the case $\Gamma=1$, those values are the same, equal to the critical pressure ratio. For the trivial case of $\Gamma=0$, the values are 0 and 1.

![Figure 2. Control system representation of algorithm.](image)

The process of determining $R$ given a value of $\Gamma$ is shown schematically in terms of a feedback control network in Figure 2. An initial value of $R$ is assumed. The required value of $\Gamma$ is compared to the computed value, and the difference is integrated with each iteration. The tanh function is applied to the result, and scaled to the range between $R_a$ and the limiting value of $R$ (1 or 0). The purpose of tanh is to guarantee that no matter how large the integral of the error becomes, the output only approaches the limiting value of $R$ (1 or 0). Since $R$ never exceeds its limiting value, no limit cycle oscillation in the value of $R$ occurs and the algorithm is stable. The tanh linearization also improves convergence times. Convergence time is further improved by using an initial value $R_a$ close to the probable final value. Thus, $R_a$ is set to $R_c$ if $\Gamma \geq 0.5$ and 1 or 0 if $\Gamma < 0.5$.

The steady-state output of this network is the value of $R$ that corresponds to the desired value of $\Gamma$. The residual or steady state error can be shown to be identically zero. Two parallel networks are used, one for each branch of the $\Gamma$ function. They differ from each other only in the integration sign and in the limiting $R$-value used (1 for the top branch, 0 for the lower). The integral gain $K$ may also be specified separately for the two branches to optimize convergence.
A similar approach is used in the numerical solution of Eq. 32 as well. It is unique in having a matrix of gains corresponding to different regions of the input space.

MATLAB Function Listings
Following are MATLAB m-file implementations of some of the functions described above.

% Normalized Gas Dynamics Function Gamma J.S. Jacob 31 May 2001
%  G = gamma(R,k)
% Where G is the normalized gas dynamics function,
% R is the pressure ratio. R may be a vector.
% k is the optional scalar ratio of specific heats (default = 1.4 for air)
% eq. 4.19.

function G = gamma(R,k)
if nargin == 1
    k = 1.4;
end
if nargin < 1 | nargin > 2
    error('Not enough or too many arguments');
end
k = k(1);                           % ensure k is a scalar
a = 2/k;  b = (k+1)/k;  c = (2/(k+1))^(2/(k-1)); d = (k-1)/(k+1);
G = sqrt((R.^a - R.^b)/(c*d));

% Inverse Gas Dynamics Function J.S. Jacob 31 May 2001
% [Rsubsonic,Rsupersonic] = invgamma(G,k)
%
% Rsubsonic is the solution on the upper (subsonic) branch of gamma.
% Rsupersonic is the solution on the lower (supersonic) branch.
% Rxxxxx is the ratio of static (local) pressure to the isentropic
% stagnation (reference) pressure or "total" pressure.
% G is the normalized gas dynamics function. Valid values are 0 to 1.
% Only scalar values are valid for G.
% k is the optional ratio of specific heats (default = 1.4)
% See gamma.m, gammaM.m.

function [Rh,Rl] = invgamma(G,k)

if G == 0           % Test the trivial case & return
    Rh = 1;
    Rl = 0;
    return
end

if G < 0 | G > 1
    error('Input value out of range.')
end

if nargin == 1      % default value of k
    k = 1.4;
end

k = k(1);           % ensure k is a scalar

% compute critical ratio & other oft-used expressions
Rcrit = (2/(k+1))^(k/(k-1));
a = 2/k;  b = (k+1)/k;  c = (2/(k+1))^(2/(k-1))*(k-1)/(k+1);
Rl = Rcrit;  Rh = Rcrit; dh = 1-Rcrit; dl = Rcrit;

if G == 1         % The other trivial solution: Rl = Rh = Rcrit valid for G = 1.
    return
end

% Proportional and integral gains set; precision limits set.
kph = 0; kpl = 0; kih = 2*G; kil = 2*G; Ih = 0; Il = 0;
hlim = .00001;  llim = .00001; stillgoing = 1; n = 0;

errh = 1;  errl = 1; % initialize errors

if G > 0.5
    % Iterate to a dual solution
    while stillgoing
        n = n + 1;
        g = gamma([min(1,Rh),max(0,Rl)],a,b,c);
        errh = g(1)-G;  errl = g(2)-G;
        stillgoing = (abs(errh)>hlim) & (abs(errl)>llim);
        Ih = Ih + errh;  Il = Il + errl;  % integrate the error
        rh = Rcrit + dh*tanh(kih*Ih + kph*errh);
        rl = Rcrit - dl*tanh(kil*Il + kpl*errl);
        if rh < Rcrit | rh > 1
            kih = .9*kih; stillgoing = 1;
        else
            Rh = rh;
        end
        if rl > Rcrit | rl < 0
            kil = .9*kil; stillgoing = 1;
        end
    end
end
else
    \texttt{Rl = rl;}
end

\% \texttt{disp([errh errl]);} \\
\% pause
end

\texttt{else} \quad \% \texttt{G <= 0.5}

\texttt{kph = 0; kpl = 0; kih = 1*G; kil = 2*G; Ih = 0; I1 = 0;}
\texttt{hlim = .00001; llim = .00001; stillgoing = 1; n = 0;}
\texttt{Rh = 1; Rl = 0;} \quad \% \texttt{Start from G = 0 values}

\% \texttt{Iterate to a dual solution}

\texttt{while stillgoing}
\texttt{n = n + 1;}
\texttt{g = gamma([min(1,Rh),max(0,Rl)],a,b,c);}
\texttt{errh = g(1)-G; errl = g(2)-G;}
\texttt{stillgoing = (abs(errh)>hlim) | (abs(errl)>llim);}
\texttt{Ih = Ih + errh; I1 = I1 + errl; \% integrate the error}
\texttt{rh = 1 + dh*tanh(kih*Ih + kph*errh);}
\texttt{rl = 0 - dl*tanh(kil*I1 + kpl*errl);}
\texttt{if rh < Rcrit \texttt{\mid rh > 1}}
\texttt{kih = .9*kih; stillgoing = 1;}
\texttt{else}
\texttt{Rh = rh;}
\texttt{end}
\texttt{if rl > Rcrit \texttt{\mid rl < 0}}
\texttt{kil = .9*kil; stillgoing = 1;}
\texttt{else}
\texttt{Rl = rl;}
\texttt{end}

\% \texttt{disp([errh errl]);} \\
\% pause
end \quad \% \texttt{while}

\texttt{end} \quad \% \texttt{if-else}
\texttt{disp(n)} \quad \% \texttt{Debugging \& optimization - outputs number of its}
\texttt{return} \quad \% \texttt{function invgamma}

\% The Normalized Gas Dynamics Function:
\texttt{function G = gamma(R,a,b,c)}
\texttt{G = sqrt((R.^a - R.^b)/c);}

\% Ratio of Gamma solution function - reduced equation
\% \texttt{J. S. Jacob} \\
\% 10 Sept 2001
\% \texttt{Iterates to a solution of the g equation (see notes).}
\% This is a reduced form of the gamma ratios function, which
\% simplifies finding a numerical solution somewhat.
% This solution is limited to an input space of r & g between
% 0.02 and 50. It could be modified to take values outside
% this input space, but solution times will be very long
% and convergence is not necessarily assured.
%
function R = g2ratio(r,g,k)

    if nargin == 2  % Default value for k is 1.4, valid for air.
        k = 1.4;
    end

    a = 2/k;  b = (k+1)/k;  % oft-used constants defined here.
    Rmax = min(1,1/r);  % define input space grid:
    ghi = [.1 1 5 10 50];  glo = [.02 .1 1 5 10];
    rhi = [.1 1 2 5 10];  rlo = [.02 .1 1 2 5 10];

    Ks = [10.0 0.7 .13 .055 .011; 10.0 2.00 0.70 0.10 0.02;
         -.40 -.30 -1.0 -1.0 -1.0;
         -.095 -.17 -.4 -.3 -.1;
         -.072 -.05 -.07 -.15 -.32;
         -.0095 -.0085 -.009 -.0095 -.022];

    Rst = [0.5 0.9 0.96 0.99 0.99; 0.2 0.6 0.7 0.96 0.999;
           Rmax Rmax Rmax Rmax Rmax;  % Initial Values
           Rmax Rmax Rmax Rmax Rmax;
           Rmax Rmax Rmax Rmax Rmax;
           Rmax Rmax Rmax Rmax Rmax];

    ri = find(r <= rhi & r > rlo);
    gi = find(g <= ghi & g > glo);
    %disp([ri gi]);  % display g and r indexes for debugging.

    if isempty(ri)
        error(['r = ' num2str(r) '. Input out of range.'])
    end
    if isempty(gi)
        error(['g = ' num2str(g) '. Input out of range.'])
    end

    Rstart = Rst(ri,gi);
    K = Ks(ri,gi);

    % Check to see if solutions exist:

    if r == 1  % R is undefined.
        error('P1/P2 = 1; no general solution.')
    end

    if r > 1 & g > g2fun(eps,r,a,b)  % no solution
        error(['Ratio G1/G2 too high for this value of r = ' num2str(r)])
    end

    if r < 1 & g < g2fun(eps,r,a,b)  % no solution
        error(['Ratio G1/G2 too low for this value of r = ' num2str(r)])
end

% here's where I iterate to a solution.

R1 = Rstart; tol = .0001; % initialize iteration
err = 1; e1 = 1; Int = 0; count = 0; kcount = 0;

while abs(err) > tol % loop
count = round(count + 1);
kcount = round(kcount + 1);
g2 = g2fun(R1,r,a,b);
%disp([R1 g err]); % display values during debugging
%pause % wait for debugging
err = (g-g2);
if e1*err < 0 % changing sign indicates oscillation
    K = K*.95; % so gain is reduced to aid convergence
end
if e1*err > 0 % increase gain during monotonic convergence to
    K = K*1.01; % reduce convergence time
end
I2 = Int + K*err;
R2 = (tanh(I2) + 1)/2; % tanh function is bounded by -1 and 1. This is a % handy way to bound R by 0 and 1.
if R2 <= 0 | R2 >=1
    K = K*.99; % If by chance you land on or out of bounds, reduce % the gain & try again.
else
    R1 = R2; Int = I2; e1 = err;
end % but if R is OK, regress all values and iterate.
%disp(R1)
end % End of WHILE loop
%disp([R1 g2 g err]); %disp(count);
R = R1; % This is the end of this function.
%reslope(R,r,a,b) %output the slope as check for debugging

% A few oft-used expression placed here as built-in functions:
function g2 = g2fun(R,r,a,b)
d = r*R;
g2 = (d^a - d^b)/(R^a - R^b);
end

function S = reslope(R,r,a,b) % reciprocal of the slope function
    c = a-1; d = b-1; % see my notes for this derivation:
    S = (R^a - R^b)/(a^r^a*R^c - b*r^b*R^d - g2fun(R,r,a,b)*(a*R^c - b*R^d));
end

% State Point Flow Number J.S. Jacob
% 31 May 2001
%
% Computes the state point flow number from the Normalized Gas
% Dynamics Function (gamma).
%  a = alpha(G,k)
%  G is the normalized gas dynamics function, gamma.  Valid values are
%  0 to 1.  G may be vector.
%  k is the optional scalar specific heats ratio (default value 1.4 for air).
%  a is defined as (m/APt)sqrt(RT) where m is the mass flow rate, A is the flow
%  area, Pt is the isentropic stagnation pressure (total pressure or reference
%  pressure), T is abs. temperature, R is specific gas constant (287 for air).
%  When a is known, one has a relation for m in terms of A and Pt, solving the
%  basic fluid flow question.
%  eq. 4.21, 4.24, 4.25.
%  See gamma.m, invgamma.m

function a = alpha(G,k)
    if nargin == 1
        k = 1.4;
    end
    if nargin < 1 | nargin > 2
        error('Not enough or too many arguments');
    end
    k = k(1);                           % ensure k is a scalar
    b = 2/(k+1);  c = (k+1)/(k-1);
    acrit = sqrt(k*b^c);                % eq. 4.24
    a = acrit*G;                        % eq. 4.25

% Pressure ratio from Mach number

function R = m2r(M,k)
    if nargin == 1
        k = 1.4;
    end
    M is the Mach number.  M may be a vector.
    k is the optional scalar ratio of specific heats (default = 1.4 for air)
    m2r.m Relates the pressure ratio, defined as p/pt, to the Mach number.
    pt is the isentropic stagnation pressure, aka the reference pressure
    or total pressure.  Mach number is V/c where V is local flow velocity
    and c is speed of sound, sqrt(kRT).
    eq. 3.13.
    See r2m.m, gamma.m, gammaM.m, invgamma.m, alpha.m

function R = m2r(M,k)
    if nargin == 1
        k = 1.4;
if nargin < 1 | nargin > 2
    error('Not enough or too many arguments');
end
k = k(1); % ensure k is a scalar
a = (k-1)/2;  b = k/(k-1);
R = (1+a*M.^2).^(-b);

% Mach number from pressure ratio  J.S. Jacob
% eq. 3.14.  Note - printed equation is incorrect.
% See m2r.m
%
function M = r2m(R,k)
if nargin == 1
    k = 1.4;
end
if nargin < 1 | nargin > 2
    error('Not enough or too many arguments');
end
k = k(1); % ensure k is a scalar
a = 2/(k-1);  b = (k-1)/k;
M = sqrt(a*(R.^(-b) - 1));