Compressible Flow Subroutine Library Reference Manual (Fortran) DD-00008-110E

Jan Adelsbach

October 2, 2024

Contents

1 About this Guide

1.1 Legal Information

1.2 Feedback and Contact

For feedback on this document, please use the following email address:
techpubs@adelsbach-research.eu

Please include the page number or a link to the page.

For general contact details, please visit https://adelsbach-research.eu/contact.

1.3 Introduction

This manual describes the *Application Programming Interface* (API) of the *Compressible Flow Subroutine Library* for the Fortran programming language families.

1.4 Audience for This Guide

The audience of this guide is assumed to be Fortran programmers who understand the basic concepts of at least one of the aforementioned programming languages.

1.5 How to Use This Guide

This guide first describes some general programming details of the library and then documents each function individually.

1.6 Conventions Used in This Guide

x
: Normal math typesetting represents a normal variable.

$\mathbf{x}$
: Bold math typesetting represents a vector.

`Mono`
: Monospace typesetting represents C function names, variables or data types.

2 Overview

2.1 Performance Characteristics

Each function or subroutine has a declared performance characteristic, these represent how the performance of a function can be predicted.

Fixed The function always executes around the same amount of instructions. Depending upon the actual values passed in the arguments there might be insignificant variations but do not contain any iterative solver. The behavior of standard library functions used is not considered for this performance classification.

Iterative The function uses an iterative root finding algorithm to compute the result. In the library all functions which use iterative solvers have a maximum amount of iterations, after which they will fail.

Iterative solvers are usually non-linear root finding methods such as Newton-Rapshon, False Position or Bisection style solvers or can be solvers for differential equations such as Runge-Kutta methods as described in [**DKahaner1988**].

2.2 Error Behavior

All library subroutines where an error may occur usually have an optional integer pointer argument. If the latter is given and an error occurs the error code will be written into the integer at the pointer's location. If no error occurs the integer will not be changed. As such this allows by design a logical-OR behavior in which one may call several subroutines and check at the very end whether an error code has been set.

This error code is also set into a *thread-local* variable and can be inquired using the `3.1.3` function for the current thread. The latter will also provide the name of the offending function for easier traceback.

Functions that are designed for specific purposes may however return `NaN` (*Not-a-Number*) or infinity ∞. The specific error behavior for those is documented for each of such functions individually.

3 Function Reference

3.1 System functions

3.1.1 Introduction

The functions in this group provide system information about the CFSL library and error handling facilities.

3.1.2 `arsyver` - Get version information

```
SUBROUTINE AR01AA(MAJOR, MINOR, CMPLR)
  INTEGER MAJOR, MINOR
  CHARACTER CMPLR(16)
```

Return the major, minor version and the compiler used to compile the library.

Arguments

`MAJOR` - **SHORT INTEGER** *EXIT:* (*optional*) Pointer to store major version in.

`MINOR` - **SHORT INTEGER** *EXIT:* (*optional*) Pointer to store minor version in.

`CMPLR` - **CHAR POINTER** *EXIT:* (*optional*) Pointer to store a pointer to the compiler version string.

3.1.3 `arsyerr` - Retrieve and reset error code

```
INTEGER FUNCTION AR01AB(FN, CLEAR)
  CHARACTER FN(16)
  LOGICAL CLEAR
```

Returns the last error code and messages that occured in the current thread.

Parameters

`FN` **- CHAR POINTER** *EXIT:* (*optional*) Pointer to store a pointer to the function name string.

`CLEAR` **- BOOLEAN** *ENTRY:* Whether to clear the error code from the *thread-local* context.

3.2 Characteristic Mach Number

3.2.1 Introduction

This function group computes the characteristic Mach Number M^* as given in [**JDAnderson1982**] and [**Shapiro1953**] from a given mach number M or in reverse the latter given M^*. The characteristic mach number has the following properties:

$$\begin{aligned} M^* &= 0 \quad \text{if} \quad \mathrm{M} = 0 \\ M^* &= 1 \quad \text{if} \quad \mathrm{M} = 1 \\ M^* &< 1 \quad \text{if} \quad \mathrm{M} < 1 \\ M^* &> 1 \quad \text{if} \quad \mathrm{M} > 1 \end{aligned}$$

And furthermore the its most useful property:

$$\lim_{M \to \infty} M^* = \sqrt{\frac{\gamma+1}{\gamma-1}}$$

The following properties are computed:

Table 1: Characteristic mach number properties

I	**Symbol**	**Description**
1	M	Mach number
2	M^*	Characteristic mach number

3.2.2 `armcmm` - Compute given normal mach number, M

```
DOUBLE PRECISION FUNCTION AR02AA(G,M,IERR)
  DOUBLE PRECISION G, M
  INTEGER IERR
```

Compute given normal mach number, M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status Codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given $M \leq 0$.

Return Value
Characteristic mach number M^* or `NAN` on error.

3.2.3 `armcms` - Compute given characteristic mach number, M^*

```
DOUBLE PRECISION FUNCTION AR02AB(G,MSTAR,IERR)
  DOUBLE PRECISION G, MSTAR
  INTEGER IERR
```

Compute given the characteristic mach number, M^*. Note that if $M^* = \sqrt{(\gamma+1)(\gamma-1)}$ the function will return $M = \infty =$ `INFINITY`.

Performance
Fixed

Parameters

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`MSTAR` **- REAL** *ENTRY:* Characteristic mach number $0 < M^* \leq \sqrt{(\gamma+1)(\gamma-1)}$.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status Codes

-1 Specific heat ratio $\gamma \leq 1$.

-3 Given M^* out of specified range.

Return Value
Normal mach number M or `NAN` on error.

3.3 Hugoniot compression

3.3.1 Introduction

Hugoniots equation can be used to model a normal shock wave by thermodynamic constants only as given by [**JDAnderson1982**]. As such this equations allows the relation of the pressure factor over the shock wave p_2/p_1 to the density factor over the same ρ_2/ρ_1 given a specific heat constant γ without being related to a velocity or mach number. Note that the the hugoniot based shock wave compression does not yield exactly the same results as normal shock wave relations and in fact deviates from them at higher mach numbers. The following properties are computed:

Table 2: Hugoniot compression properties

I	Symbol	Description
1	p_2/p_1	Pressure factor
2	ρ_2/ρ_1	Density factor

3.3.2 `arshud` - Compute given density factor ρ_2/ρ_1

```
DOUBLE PRECISION FUNCTION AR03AA(G,D,IERR)
  DOUBLE PRECISION G, D
  INTEGER IERR
```

Compute pressure factor based upon density factor ρ_2/ρ_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density fraction $0 < \rho_2/\rho_1 < \sqrt{\frac{\gamma+1}{\gamma-2}}$.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given ρ_2/ρ_1 out of range.

Return Value
Pressure fraction p_2/p_1 or `NAN` on error.

3.3.3 `arshup` - Compute given density factor p_2/p_1

```
DOUBLE PRECISION FUNCTION AR03AB(G,P,IERR)
  DOUBLE PRECISION G, P
  INTEGER IERR
```

Compute pressure factor based upon pressure factor p_2/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure fraction $p_2/p_1 > 0$.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given p_2/p_1 out of range.

Return Value
Density fraction ρ_2/ρ_1 or `NAN` on error.

3.4 Critical mach number

3.4.1 armcrm - Critical pressure coefficient given M_{crit}

```
DOUBLE PRECISION FUNCTION AR04AA(G,MCRIT,IERR)
  DOUBLE PRECISION G, MCRIT
  INTEGER IERR
```

Compute given the critical pressure coefficient given M_{crit}.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`MCRIT` **- REAL** *ENTRY:* Critical Mach number $0 < M_{\text{crit}} < 1$.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_{crit} out of range.

Return Value
Critical pressure coefficient C_p or `NAN` on error.

3.4.2 `armcrc` - Critical mach number given C_p

```
DOUBLE PRECISION FUNCTION AR04AB(G,CP,IERR)
  DOUBLE PRECISION G, CP
  INTEGER IERR
```

Compute given the critical mach number given C_p.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`CP` - **REAL** *ENTRY:* Critical pressure coefficent C_p.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given C_p out of range.

-3 Iterative solver failed.

Return Value
Critical mach number M_{crit} or `NAN` on error.

3.4.3 `armcrpg` - Critical mach number given C_{p0} using Prandtl-Glauert pressure approximation

```
DOUBLE PRECISION FUNCTION AR04AC(G,CP0,IERR)
  DOUBLE PRECISION G, CP0
  INTEGER IERR
```

Compute critical mach number given C_{p0} Prandtl-Glauert pressure approximation.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`CP0` **- REAL** *ENTRY:* Critical pressure coefficent C_{p0}.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Iterative solver failed.

Return Value
Critical mach number M_{crit} or `NAN` on error.

3.4.4 `armcrla` - Critical mach number given C_{p0} using Laitone pressure approximation

```
DOUBLE PRECISION FUNCTION AR04AD(G,CP0,IERR)
  DOUBLE PRECISION G, CP0
  INTEGER IERR
```

Compute critical mach number given C_{p0} Laitone pressure approximation.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`CP0` **- REAL** *ENTRY:* Critical pressure coefficent C_{p0}.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Iterative solver failed.

Return Value
Critical mach number M_{crit} or `NAN` on error.

3.4.5 `armcrkt` - Critical mach number given C_{p0} using Karman-Tsien pressure approximation

```
DOUBLE PRECISION FUNCTION AR04AE(G,CP0,IERR)
  DOUBLE PRECISION G, CP0
  INTEGER IERR
```

Compute critical mach number given C_{p0} Karman-Tsien pressure approximation.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`CP0` **- REAL** *ENTRY:* Critical pressure coefficent C_{p0}.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Iterative solver failed.

Return Value
Critical mach number M_{crit} or `NAN` on error.

3.5 Isentropic flow

3.5.1 Introduction

This function group computes the thermodynamic properties of isentropic flow for a calorically perfect gas as described in [**naca1135**] with one-dimensional variable changes. Isentropic flow assumes the following continuity equation:

$$\rho AV = \text{const}$$

The momentum equation is:

$$p + \rho AV^2 = \text{const}$$

The following properties are computed:

Table 3: Isentropic flow properties

I	Property	Description
1	M	Mach number
2	p_0/p_1	Pressure ratio
3	ρ_0/ρ_1	Density ratio
4	T_0/T_1	Temperature ratio
5	A/A^*	Critical area ratio

3.5.2 `arfism` - Isentropic relations given M

```
SUBROUTINE AR05AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(5)
  INTEGER IERR
```

Compute the isentropic relations given the mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 3.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.5.3 `arfisp` - Isentropic relations given p_2/p_1

```
SUBROUTINE AR05AB(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(5)
  INTEGER IERR
```

Compute the isentropic relations given the pressure ratio p_2/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $0 \leq p_2/p_1 \leq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 3.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p_2/p_1 out of range.

3.5.4 `arfisd` - Isentropic relations given ρ_2/ρ_1

```
SUBROUTINE AR05AC(G,D,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(5)
  INTEGER IERR
```

Compute the isentropic relations given the density ratio ρ_2/ρ_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`R` **- REAL** *ENTRY:* Density ratio $0 \leq \rho_2/\rho_1 \leq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 3.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given ρ_2/ρ_1 out of range.

3.5.5 `arfist` - Isentropic relations given T_2/T_1

```
SUBROUTINE AR05AD(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(5)
  INTEGER IERR
```

Compute the isentropic relations given the temperature ratio T_2/T_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio $0 \leq T_2/T_1 \leq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 3.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T_2/T_1 out of range.

3.5.6 `arfisa` - Isentropic relations given A/A^*

```
SUBROUTINE AR05AE(G,A,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, A, RESULT(5)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the isentropic relations given the critical area ratio A/A^*.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`A` - **REAL** *ENTRY:* Critical area ratio $A/A^* \geq 1$.

`IS_SUB` - **BOOLEAN** *ENTRY:* Whether the mach number correspdonding to A/A^* is assumed to be sub- or super- sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 3.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given A/A^* out of range.

-4 Iterative solver failed

3.6 Fanno flow

3.6.1 Introduction

This function group computes the thermodynamic properties of adiabatic flow with friction for calorically perfect gas as described in [**JDAnderson1982**].

The following properties are computed:

Table 4: Fanno flow properties

I	Property	Description
1	M	Mach number
2	p/p^*	Pressure ratio
3	ρ/ρ^*	Density ratio
4	T/T^*	Temperature ratio
5	V/V^*	Velocity ratio
6	p_{01}/p_{02}	Stagnation pressure ratio
7	$(4fL)/D$	Fanno line
8	$(s-s^*)/c_p$	Change in entropy

3.6.2 `arffam` - Fanno relations given M

```
SUBROUTINE AR06AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(8)
  INTEGER IERR
```

Description
Compute the fanno relations given a mach number, M.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`M` - **REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \le 1$.

-2 Given M out of range.

3.6.3 `arffap` - Fanno relations given p/p^*

```
SUBROUTINE AR06AB(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(8)
  INTEGER IERR
```

Compute the fanno relations given the pressure ratio, p/p^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $0 \leq p/p^*$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p/p^* out of range.

3.6.4 `arffad` - Fanno relations given ρ/ρ^*

```
SUBROUTINE AR06AC(G,D,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(8)
  INTEGER IERR
```

Compute the fanno relations given the density ratio, ρ/ρ^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density ratio $\sqrt{(\gamma-1)/(\gamma+1)} \leq \rho/\rho^*$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given ρ/ρ^* out of range.

3.6.5 `arffat` - Fanno relations given T/T^*

```
SUBROUTINE AR06AD(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(8)
  INTEGER IERR
```

Compute the fanno relations given the temperature ratio, T/T^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio $0 \leq T/T^* \leq (\gamma+1)/(\gamma-1)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T/T^* out of range.

3.6.6 `arffav` - Fanno relations given V/V^*

```
SUBROUTINE AR06AE(G,V,RESULT,IERR)
  DOUBLE PRECISION G, V, RESULT(8)
  INTEGER IERR
```

Compute the fanno relations given the velocity ratio, V/V^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`V` **- REAL** *ENTRY:* Velocity ratio $0 \leq V/V^* \leq \sqrt{(\gamma+1)/(\gamma-1)}$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given V/V^* out of range.

3.6.7 `arffap0` - Fanno relations given p_{01}/p_{02}

```
SUBROUTINE AR06AF(G,P0,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, P0, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the fanno relations given the stagnation pressure ratio, p_{01}/p_{02}.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P0` **- REAL** *ENTRY:* Stagnation pressure ratio $1 \leq p_{02}/p_{01}$.

`IS_SUB` **- BOOLEAN** *ENTRY:* Whether M corresponding to the given p_{02}/p_{01} is assumed to be sub- or super- sonic.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p_{01}/p_{02} out of range.

-4 Iterative solver failed.

3.6.8 `arffaf` - Fanno relations given $(4fL)/D$

```
SUBROUTINE AR06AG(G,L,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, L, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the fanno relations given the fanno line, $(4fL)/D$.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`L` **- REAL** *ENTRY:* Fanno line $0 \leq (4fL)/D \leq ((\gamma+1)\log((\gamma+1)/(\gamma-1))-2)/(2\gamma)$.

`IS_SUB` **- BOOLEAN** *ENTRY:* Whether M corresponding to the given $(4fL)/D$ is assumed to be sub- or super- sonic.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given $(4fL)/D$ out of range.

-4 Iterative solver failed.

3.6.9 `arffads` - Fanno relations given $(s-s^*)/c_p$

```
SUBROUTINE AR06AH(G,DS,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, DS, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the fanno relations given the fanno line, $(s-s^*)/c_p$.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`DS` - **REAL** *ENTRY:* Change in entropy $-\infty < (s-s^*)/c_p < \infty$.

`IS_SUB` - **BOOLEAN** *ENTRY:* Whether M corresponding to the given $(s-s^*)/c_p$ is assumed to be sub- or super- sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 4.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given $(s-s^*)/c_p$ out of range.

-4 Iterative solver failed.

3.7 Rayleigh Flow

3.7.1 Introduction

This function group computes the thermodynamic properties of non-adiabatic flow with heat addition for a calorically perfect gas as described in [**JDAnderson1982**].

The computed properties are as follows:

Table 5: Rayleigh flow properties

I	Property	Description
1	M	Mach number
2	p/p^*	Pressure ratio
3	ρ/ρ^*	Density ratio
4	T/T^*	Temperature ratio
5	V/V^*	Velocity ratio
6	p_{01}/p_{02}	Stagnation pressure ratio
7	T_{01}/T_{02}	Stagnation temperature ratio
8	$(s-s^*)/c_p$	Change in entropy

3.7.2 `arfram` - Rayleigh relations given M

```
SUBROUTINE AR07AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(8)
  INTEGER IERR
```

Compute the rayleigh relations given a mach number, M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.7.3 `arfrap` - Rayleigh relations given p/p^*

```
SUBROUTINE AR07AB(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(8)
  INTEGER IERR
```

Compute the rayleigh relations given the pressure ratio, p/p^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $0 < p/p^* < \gamma + 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p/p^* out of range.

3.7.4 `arfrad` - Rayleigh relations given ρ/ρ^*

```
SUBROUTINE AR07AC(G,D,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(8)
  INTEGER IERR
```

Compute the rayleigh relations given the density ratio, ρ/ρ^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density ratio $\gamma(\gamma+1) < \rho/\rho^*$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given ρ/ρ^* out of range.

3.7.5 `arfrat` - Rayleigh relations given T/T^*

```
SUBROUTINE AR07AD(G,T,TMAX,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(8)
  LOGICAL TMAX
  INTEGER IERR
```

Compute the rayleigh relations given the temperature ratio, T/T^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio $0 < T/T^*$.

`TMAX` **- BOOLEAN** Whether the static temperature ratio supplied is for a low speed `false` or a high speed `true`.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T/T^* out of range.

3.7.6 `arfrav` - Rayleigh relations given V/V^*

```
SUBROUTINE AR07AE(G,V,RESULT,IERR)
  DOUBLE PRECISION G, V, RESULT(8)
  INTEGER IERR
```

Compute the rayleigh relations given the velocity ratio, V/V^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`V` **- REAL** *ENTRY:* Velocity ratio $0 < V/V^* < (\gamma+1)/\gamma$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given V/V^* out of range.

3.7.7 `arfrap0` - Rayleigh relations given p_{02}/p_{01}

```
SUBROUTINE AR07AF(G,P0,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, P0, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the rayleigh relations given the stagnation pressure ratio, p_{02}/p_{01}.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`P0` - **REAL** *ENTRY:* Stagnation pressure ratio $p_{02}/p_{01} > \frac{\gamma^2-1}{\gamma^2}$.

`IS_SUB` - **BOOLEAN** Whether M corresponding to the given p_{02}/p_{01} is assumed to be sub- or super-sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p_{02}/p_{01} out of range.

-4 Iterative solver failed.

3.7.8 `arfrat0` - Rayleigh relations given T_{02}/T_{01}

```
SUBROUTINE AR07AG(G,T0,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, T0, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the rayleigh relations given the stagnation temperature ratio, T_{02}/T_{01}.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`T0` - **REAL** *ENTRY:* Stagnation temperature ratio $T_{02}/T_{01} \geq 0$.

`IS_SUB` - **BOOLEAN** Whether M corresponding to the given T_{02}/T_{01} is assumed to be sub- or super-sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T_{02}/T_{01} out of range.

-4 Iterative solver failed.

3.7.9 `arfrads` - Rayleigh relations given $(s - s^*)/c_p$

```
SUBROUTINE AR07AH(G,DS,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, DS, RESULT(8)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the rayleigh relations given the fanno line, $(s - s^*)/c_p$.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`DS` **- REAL** *ENTRY:* Change in entropy $-\infty < (s - s^*)/c_p < \infty$.

`IS_SUB` **- BOOLEAN** Whether M corresponding to the given $(s - s^*)/c_p$ is assumed to be sub- or super- sonic.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 5.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given $(s - s^*)/c_p$ out of range.

-4 Iterative solver failed.

3.8 Isothermal flow through long duct

3.8.1 Introduction

This function group computes an isothermal flow through a long ducts as described in [**VLStreeter1966**] for which neither fanno nor rayleigh flow is applicable due to the assumptions of the two latter.

The following properties are computed:

Table 6: Isothermal flow properties

I	Property	Description
1	M	Mach number
2	$\frac{p^*}{p}, \frac{r^*}{r}, \frac{M^*}{M}$	Non-stagnative ratios
3	ρ/ρ^*	Density ratio
5	V/V^*	Velocity ratio
6	p_{01}/p_{02}	Stagnation pressure ratio
6	T_{01}/T_{02}	Stagnation temperature ratio
7	$(4fL)/D$	Characteristic line

3.8.2 `arfitm` - Isothermal relations given M

```
SUBROUTINE AR08AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(6)
  INTEGER IERR
```

Compute the isothermal relations given the mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 6.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.8.3 `arfitv` - Isothermal relations given V/V^*

```
SUBROUTINE AR08AB(G,V,RESULT,IERR)
  DOUBLE PRECISION G, V, RESULT(6)
  INTEGER IERR
```

Compute the isothermal relations given the velocity ratio, V/V^*.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`V` **- REAL** *ENTRY:* Velocity ratio $V/V^* > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 6.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given V/V^* out of range.

3.8.4 `arfits` - Isothermal relations given $p^*/p, r^*/r, M^*/M$

```
SUBROUTINE AR08AC(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(6)
  INTEGER IERR
```

Compute the isothermal relations given a non-stagnative ratio, $p^*/p, r^*/r, M^*/M$.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`S` - **REAL** *ENTRY:* Non-stagnative ratio $p^*/p, r^*/r, M^*/M > 0$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 6.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given non-stagnative ratio out of range.

3.8.5 `arfitp0` - Isothermal relations given p_{01}/p_{02}

```
SUBROUTINE AR08AD(G,P0,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, P0, RESULT(6)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the isothermal relations given the stagnation pressure ratio, p_{01}/p_{02}.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`P0` - **REAL** *ENTRY:* Stagnation pressure ratio $1 \leq p_{02}/p_{01}$.

`IS_SUB` - **BOOLEAN** *ENTRY:* Whether M corresponding to the given p_{02}/p_{01} is assumed to be sub- or super- sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 6.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p_{01}/p_{02} out of range.

-4 Iterative solver failed.

3.8.6 `arfitt0` - Isothermal relations given T_{02}/T_{01}

```
SUBROUTINE AR08AE(G,T0,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, T0, RESULT(6)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the isothermal relations given the stagnation temperature T_{02}/T_{01}.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`T0` - **REAL** *ENTRY:* Stagnation temperature $T_{02}/T_{01} > 0$.

`RESULT` - **ARRAY OF REAL** *EXIT*: Array with result properties as described in 6.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given T_{02}/T_{01} out of range.

3.8.7 `arfitf` - Isothermal relations given $(4fL)/D$

```
SUBROUTINE AR08AF(G,F,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, F, RESULT(6)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the isothermal relations given the characteristic line, $(4fL)/D$.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`F` - **REAL** *ENTRY:* Characteristic line $(4fL)/D$.

`IS_SUB` - **BOOLEAN** *ENTRY:* Whether M corresponding to the given $(4fL)/D$ is assumed to be sub- or super- sonic.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 6.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given $(4fL)/D$ out of range.

-4 Iterative solver failed.

3.9 DeLaval nozzle flow

3.9.1 Introduction

This function group computes the properties of *quasi-2D* isentropic flow through a DeLaval nozzle, as described in [**JDAnderson1982**].

The following properties are computed:

Table 7: DeLaval nozzle properties

I	Property	Description
1	M	Mach number
2	p/p_1	Pressure ratio
3	ρ/ρ^*	Density ratio
4	T/T_1	Temperature ratio
5	A/A_1	Critical area ratio

3.9.2 `arfqism` - DeLaval isentropic flow relations given M

```
SUBROUTINE AR09AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(5)
  INTEGER IERR
```

Compute the DeLaval nozzle isentropic relations given the mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 7.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.9.3 `arfqisp` - DeLaval isentropic flow relations given p/p_1

```
SUBROUTINE AR09AB(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(5)
  INTEGER IERR
```

Compute the DeLaval nozzle isentropic relations given the pressure ratio p/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $0 < p/p_1 < 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 7.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \le 1$.

-2 Computed M out of range.

-3 Given p/p_1 out of range.

3.9.4 `arfqisd` - DeLaval isentropic flow relations given ρ/ρ_1

```
SUBROUTINE AR09AC(G,D,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(5)
  INTEGER IERR
```

Compute the DeLaval nozzle isentropic relations given the density ratio p/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density ratio $0 < \rho/\rho_1 < 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 7.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given ρ/ρ_1 out of range.

3.9.5 `arfqist` - DeLaval isentropic flow relations given T/T_1

```
SUBROUTINE AR09AD(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(5)
  INTEGER IERR
```

Compute the DeLaval nozzle isentropic relations given the temperature ratio T/T_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio $0 < T/T_1 < 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 7.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T/T_1 out of range.

3.9.6 `arfqisa` - DeLaval isentropic flow relations given A/A^*

```
SUBROUTINE AR09AE(G,A,IS_SUB,RESULT,IERR)
  DOUBLE PRECISION G, A, RESULT(5)
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the DeLaval nozzle isentropic relations given the critical area ratio A/A^*.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`A` **- REAL** *ENTRY:* Critical area ratio $A/A^* > 1$.

`IS_SUB` **- BOOLEAN** *ENTRY:* Whether the mach number correspdonding to A/A^* is assumed to be sub- or super- sonic.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 7.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \le 1$.

-2 Computed M out of range.

-3 Given A/A^* out of range.

-4 Iterative solver failed.

3.10 Normal Shock Waves

3.10.1 Introduction

These functions compute the thermodynamic properties across a normal shock wave, assuming a calorically perfect gas and an adiabatic and frictionless flow, as described in [**naca1135**]. It can compute the following set of properties given one of the properties in conjunction with a specific heat ratio γ of the gas medium.

The following equations govern normal shock waves as implemented in the function group:

Equation of Mass

$$p_1 u_1 = p_2 u_2$$

Equation of Momentum

$$p_1 + \rho_1 u_1^2 = p_2 + \rho_2 u_2^2$$

Equation of Energy

$$\frac{1}{2} u_1^2 + h_1 = \frac{1}{2} u_2^2 + h_2 \quad \text{[adiab]}$$

Normal Shock Waves are classified by the following properties, where the subscript 1,2 refers to upstream and downstream, respectively.

The following properties are computed:

Table 8: Normal shock properties

I	**Property**	**Description**
1	M_{1n}	Upstream mach number
2	M_{2n}	Downstream mach number
3	p_2/p_1	Pressure ratio
4	ρ_2/ρ_1	Density ratio
5	T_2/T_1	Temperature ratio
6	p_{02}/p_{01}	Stagnation pressure ratio

3.10.2 `arsnsm1` - Normal shock relations given M_{1n}

```
SUBROUTINE AR10AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the upstream mach number M_{1n}.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`M` - **REAL** *ENTRY:* Upstream number $M_{1n} \geq 1$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_{1n} out of range.

3.10.3 `arsnsm2` - Normal shock relations given M_{2n}

```
SUBROUTINE AR10AB(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the downstream mach number M_{2n}.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`M` - **REAL** *ENTRY:* Upstream number $M_{2n} > 0$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_{2n} out of range.

3.10.4 `arsnsp` - Normal shock relations given p_2/p_1

```
SUBROUTINE AR10AC(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the pressure ratio p_2/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $p_2/p_1 < 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M_{1n} out of range.

-3 Given p_2/p_1 out of range.

3.10.5 `arsnsd` - Normal shock relations given ρ_2/ρ_1

```
SUBROUTINE AR10AD(G,D,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the density ratio ρ_2/ρ_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density ratio $1 \leq \rho_2/\rho_1 \leq (\gamma+1)/(\gamma-1)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M_{1n} out of range.

-3 Given ρ_2/ρ_1 out of range.

3.10.6 `arsnst` - Normal shock relations given T_2/T_1

```
SUBROUTINE AR10AE(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the temperature ratio T_2/T_1.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`T` - **REAL** *ENTRY:* Temperature ratio $T_2/T_1 \geq 1$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M_{1n} out of range.

-3 Given T_2/T_1 out of range.

3.10.7 `arsnsp0` - Normal shock relations given p_{02}/p_{01}

```
SUBROUTINE AR10AF(G,P0,RESULT,IERR)
  DOUBLE PRECISION G, P0, RESULT(5)
  INTEGER IERR
```

Compute the normal shock relations given the stagnation pressure ratio p_{02}/p_{01}.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P0` **- REAL** *ENTRY:* Stagnation pressure ratio $0 \leq p_{02}/p_{01} \leq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 8.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M_{1n} out of range.

-3 Given p_{02}/p_{01} out of range.

-4 Iterative solver failed.

3.11 Oblique Shock Relations

3.11.1 Introduction

This function group computes the thermodynamic properties of an oblique shock wave given either a mach number and the surface deflection or wave angle or based upon both the wave and deflection surface angles, using algorithms as described in [**usafv1**], [**naca1135**] and [**NASA187173**].

Oblique Shock Waves are classified by the following properties, where the subscript 1,2 refers to upstream and downstream, respectively. The normal mach numbers M_{1n} and M_{2n} represent the mach numbers through the shock front as by the normal shock wave equations.

The following properties will be available:

Table 9: Oblique shock properties

I	Property	Description
1	M_1	Upstream mach number
2	M_2	Downstream mach number
3	M_{1n}	Normal upstream mach number
4	M_{2n}	Normal downstream mach number
5	θ	Deflection surface angle (°)
6	δ	Wave angle (°)
7	p_2/p_1	Pressure ratio
8	ρ_2/ρ_1	Density ratio
9	T_2/T_1	Temperature ratio
10	p_{02}/p_{01}	Stagnation pressure ratio

3.11.2 `arsolmw` - Oblique shock relations given M_1 and δ

```
SUBROUTINE AR11AA(G,M,D,RESULT,IERR)
  DOUBLE PRECISION G, M, D, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given a mach number M and the wave angle δ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M_1 \geq 1$.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 out of range.

-3 Given δ out of range.

3.11.3 `arsolpw` - Oblique shock relations given p_2/p_1 and δ

```
SUBROUTINE AR11AB(G,P,D,RESULT,IERR)
  DOUBLE PRECISION G, P, D, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given the pressure ratio p_2/p_1 and the wave angle δ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $p_2/p_1 > -(\gamma-1)/(\gamma+1)$.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given p_2/p_1 out of range.

-3 Given δ out of range.

3.11.4 `arsoldw` - Oblique shock relations given ρ_2/ρ_1 and δ

```
SUBROUTINE AR11AC(G,DS,D,RESULT,IERR)
  DOUBLE PRECISION G, DS, D, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given the density ratio ρ_2/ρ_1 and the wave angle δ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`DS` **- REAL** *ENTRY:* Density ratio $0 < \rho_2/\rho_1 < (\gamma+1)/(\gamma-1)$.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given ρ_2/ρ_1 out of range.

-3 Given δ out of range.

3.11.5 `arsoltw` - Oblique shock relations given T_2/T_1 and δ

```
SUBROUTINE AR11AD(G,A,D,RESULT,IERR)
  DOUBLE PRECISION G, A, D, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given the temperature ratio T_2/T_1 and the wave angle δ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio T_2/T_1.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given T_2/T_1 out of range.

-3 Given δ out of range.

3.11.6 `arsolp0w` - Oblique shock relations given p_{02}/p_{01} and δ

```
SUBROUTINE AR11AE(G,P0,D,RESULT,IERR)
  DOUBLE PRECISION G, P0, D, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given the stagnation pressure p_{02}/p_{01} and the wave angle δ.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P0` **- REAL** *ENTRY:* Stagnation pressure ratio p_{02}/p_{01}.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given p_{02}/p_{01} out of range.

-3 Given δ out of range.

-4 Iterative solver failed.

3.11.7 `arsolws` - Oblique shock relations given θ and δ

```
SUBROUTINE AR11AF(G,D,T,RESULT,IERR)
  DOUBLE PRECISION G, D, T, RESULT(10)
  INTEGER IERR
```

Compute the oblique shock relations given the deflection surface angle θ and the wave angle δ.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`T` **- REAL** *ENTRY:* Deflection surface angle $0 < \theta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-3 Given θ or δ out of range.

3.11.8 `arsolms` - Oblique shock relations given M_1 and θ

```
SUBROUTINE AR11AG(G,M,T,STRONG,RESULT,IERR)
  DOUBLE PRECISION G, M, T, RESULT(10)
  LOGICAL STRONG
  INTEGER IERR
```

Compute the oblique shock relations given the mach number M_1 and the deflection surface angle θ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M_1 \geq 1$.

`T` **- REAL** *ENTRY:* Deflection surface angle $0 < \theta < 90$.

`STRONG` **- BOOLEAN** *ENTRY:* Whether this is a strong shock.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 9.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 out of range.

-3 Given θ out of range.

-4 Calculation failed.

3.12 Oblique Shock Limits

3.12.1 Introduction

This set of functions computes the angle limits where an oblique shock would become detached or where the downstream flow would become sonic.

The following properties are computed:

Table 10: Oblique limit properties

I	Property	Description
1	M	Mach number for which the limit applies.
2	δ_{max}	Maximum wave angle (°).
3	θ_{max}	Maximum deflection surface angle (°).

3.12.2 `arsolld` - Compute maximum wave and deflection surface angle before shock detachment

```
SUBROUTINE AR12AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(3)
  INTEGER IERR
```

Compute maximum wave and deflection surface angle before shock detachment given a mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 10.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.12.3 `arsolls` - Compute maximum wave and deflection surface angle that wil result in sonic downstream flow

```
SUBROUTINE AR12AB(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(3)
  INTEGER IERR
```

Compute maximum wave and deflection surface angle that wil result in sonic downstream flow given a mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 10.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.13 Prandtl-Meyer function

3.13.1 Introduction

This function group contains a forward and inverse Prandtl-Meyer function, these two functions are separated from the expansion fan and normal shock function groups for convenience. The properties computed are as follows:

Table 11: Prandtl-Meyer properties

I	Property	Description
1	M	Mach number.
2	$v(M)$	Prandtl-Meyer function (°).
3	θ	Mach angle (°).

3.13.2 `arspmm` - Prandtl-Meyer properties given M

```
SUBROUTINE AR13AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(3)
  INTEGER IERR
```

Compute the Prandtl-Meyer properties given a mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 11.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.13.3 `arspmv` - Prandtl-Meyer properties given $v(M)$

```
SUBROUTINE AR13AB(G,N,RESULT,IERR)
  DOUBLE PRECISION G, N, RESULT(3)
  INTEGER IERR
```

Compute the Prandtl-Meyer properties from the Prandtl-Meyer angle $v(M)$.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`N` **- REAL** *ENTRY:* Prandtl-Meyer angle $v(M) \leq 90\left(\sqrt{\frac{\gamma+1}{\gamma-1}}-1\right)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 11.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-3 Given $v(M)$ out of range.

-4 Iterative solver failed.

3.13.4 `arspmt` - Prandtl-Meyer properties given θ

```
SUBROUTINE AR13AC(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(3)
  INTEGER IERR
```

Compute the Prandtl-Meyer properties given a mach angle θ.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Mach angle $0 \le \theta \le 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 11.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \le 1$.

-2 Computed M out of range.

-3 Given θ out of range.

3.13.5 `arspmfm` - Prandtl-Meyer angle from M

```
DOUBLE PRECISION FUNCTION AR13AD(G,M)
  DOUBLE PRECISION G, M
```

Computes the Prandtl-Meyer angle $v(M)$ from the given mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

Return value
This function returns the Prandtl-Meyer angle $v(M)$ for the given mach number M or `NAN` (Not-a-Number) if an error has occurred).

3.13.6 `arspmfv` - Mach number from $v(M)$

```
DOUBLE PRECISION FUNCTION AR13AE(G,V)
  DOUBLE PRECISION G, V
```

Computes the mach number M given the Prandtl-Meyer angle $v(M)$.

Performance
Iterative

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`V` - **REAL** *ENTRY:* Prandtl-Meyer angle $v(M) \leq 90\left(\sqrt{\frac{\gamma+1}{\gamma-1}}-1\right)$.

Return value
This function returns the mach number M for the given mach Prandtl-Meyer angle $v(M)$ or `NAN` (Not-a-Number) if an error has occurred).

3.14 Expansion Fan (Rarefaction Wave)

3.14.1 Introduction

This class computes the thermodynamic properties of compressible gas flow over a wedge for a calorically perfect gas, as described by [**naca1135**]. It computes the following properties given the specific heat ratio γ and any of the up- or downstream mach number plus any of the other properties. After calling any of the evaluation functions the following properties will be available, where the subscript $1, 2$ refers to upstream and downstream, respectively;

The following properties are computed:

Table 12: Expansion fan properties

I	Property	Description
1	M_1	Upstream mach number.
2	M_2	Downstream mach number.
3	p_2/p_1	Pressure ratio.
4	ρ_2/ρ_1	Density ratio.
5	T_2/T_1	Temperature ratio.

3.14.2 `arsefm` - Expansion fan properties given M_1 and M_2

```
SUBROUTINE AR14AA(G,M1,M2,RESULT,IERR)
  DOUBLE PRECISION G, M1, M2, RESULT(5)
  INTEGER IERR
```

Compute the expansion fan properties given M_1 and M_2.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M1` **- REAL** *ENTRY:* Upstream mach number $M_1 \geq 1$.

`M2` **- REAL** *ENTRY:* Downstream mach number $M_2 > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 12.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 or M_2 out of range.

3.14.3 `arsefp` - Expansion fan properties given p_2/p_1 and either M_1 or M_2

```
SUBROUTINE AR14AB(G,P,M,IS_M2,RESULT,IERR)
  DOUBLE PRECISION G, P, M, RESULT(5)
  LOGICAL IS_M2
  INTEGER IERR
```

Compute the expansion fan properties given p_2/p_1 and either M_1 or M_2.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio p_2/p_1.

`M` **- REAL** *ENTRY:* Either $M_1 \geq 1$ or $M_2 > 0$ depending on whether `is_m2` is `false` or `true`, respectivly.

`IS_M2` **- BOOLEAN** *ENTRY:* Whether the given `m` refers to M_1 or M_2.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 12.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 or M_2 out of range.

-3 Given pressure ratio p_2/p_1 out of range.

3.14.4 `arsefd` - Expansion fan properties given ρ_2/ρ_1 and either M_1 or M_2

```
SUBROUTINE AR14AC(G,D,M,IS_M2,RESULT,IERR)
  DOUBLE PRECISION G, D, M, RESULT(5)
  LOGICAL IS_M2
  INTEGER IERR
```

Compute the expansion fan properties given ρ_2/ρ_1 and either M_1 or M_2.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`D` - **REAL** *ENTRY:* Density ratio ρ_2/ρ_1.

`M` - **REAL** *ENTRY:* Either $M_1 \geq 1$ or $M_2 > 0$ depending on whether `is_m2` is `false` or `true`, respectivly.

`IS_M2` - **BOOLEAN** *ENTRY:* Whether the given `m` refers to M_1 or M_2.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 12.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 or M_2 out of range.

-3 Given pressure ratio ρ_2/ρ_1 out of range.

3.14.5 `arseft` - Expansion fan properties given T_2/T_1 and either M_1 or M_2

```
SUBROUTINE AR14AD(G,T,M,IS_M2,RESULT,IERR)
  DOUBLE PRECISION G, T, M, RESULT(5)
  LOGICAL IS_M2
  INTEGER IERR
```

Compute the expansion fan properties given T_2/T_1 and either M_1 or M_2.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`T` - **REAL** *ENTRY:* Temperature ratio T_2/T_1.

`M` - **REAL** *ENTRY:* Either $M_1 \geq 1$ or $M_2 > 0$ depending on whether `is_m2` is `false` or `true`, respectivly.

`IS_M2` - **BOOLEAN** *ENTRY:* Whether the given `m` refers to M_1 or M_2.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 12.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M_1 or M_2 out of range.

-3 Given pressure ratio T_2/T_1 out of range.

3.15 (Rayleigh-)Pitot Tube Relations

3.15.1 Introduction

Computes the mach number out of the corresponding pressure given or inverse. For the supersonic case it computes the Rayleigh-Pitot equation which takes the bow shock in from of the pitot tube into account.

The following properties are computed: For $M < 1$ the Pitot pressure ratio p_0/p will be computed for $M \geq 1$ the

Table 13: (Rayleigh-)Pitot properties

I	Property	Description
1	M	Mach number.
2	p	Pressure ratio, see below

Rayleigh-Pitot pressure ratio p_{02}/p_1 will be computed.

3.15.2 `arsptm` - (Rayleigh-)Pitot relations given M

```
DOUBLE PRECISION FUNCTION AR15AA(G,M,IERR)
  DOUBLE PRECISION G, M
  INTEGER IERR
```

Compute the Rayleigh-Pitot properties given a mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 13.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

Return Values
The pressure ratio p_0/p or p_{02}/p_1 or `NAN` on error.

3.15.3 `arsptp` - (Rayleigh-)Pitot relations given p_0/p or p_{02}/p_1

```
DOUBLE PRECISION FUNCTION AR15AB(G,P0,IS_SUB,IERR)
  DOUBLE PRECISION G, P0
  LOGICAL IS_SUB
  INTEGER IERR
```

Compute the Rayleigh-Pitot properties given a one of the pressure ratios p_0/p or p_{02}/p_1.

Performance

- Fixed if `is_sub` is `true`.
- Iterative if `is_sub` is `false`.

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`P0` - **REAL** *ENTRY:* The pressure ratio $p_0/p \geq 1$ or $p_{02}/p_1 \geq ((1+\gamma)/2)^{(\gamma/(\gamma-1))}$ depending upon `is_sub`.

`IS_SUB` - **BOOLEAN** *ENTRY:* Whether `p0` refers to a Pitot pressure fraction (`true`) or to a Rayleigh-Pitot pressure fraction (`false`).

`RESULT` - **ARRAY OF REAL** *EXIT*: Array with result properties as described in 13.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given pressure ratio out of range.

-3 Iterative solver failed.

Return Values
The Mach number M or `NAN` on error.

3.16 Reflected Shock Waves

3.16.1 Introduction

This function group computes the mach number of a reflected shock wave as given in [**JDAnderson1982**].

The following properties are computed:

Table 14: Reflected shock wave properties

I	Property	Description
1	M	Mach number of incident shock wave
2	M'	Mach number of reflected shock wave

3.16.2 arsrsms - Reflected shock wave mach number given M

```
DOUBLE PRECISION FUNCTION AR16AA(G,MS,IERR)
  DOUBLE PRECISION G, MS
  INTEGER IERR
```

Computes the reflected shock wave mach number given M'.

Performance
Fixed

Arguments

G - **REAL** *ENTRY:* Specific heat constant γ.

MS - **REAL** *ENTRY:* Incident mach number $M \geq 1$.

RESULT - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 14.

IERR - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

Return values
The reflected shock wave mach number M' or NAN on error.

3.16.3 `arsrsmr` - Incident shock wave mach number given M'

```
DOUBLE PRECISION FUNCTION AR16AB(G,MR,IERR)
  DOUBLE PRECISION G, MR
  INTEGER IERR
```

Computes the incident shock wave mach number given M'.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M' > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 14.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M' out of range.

-3 Iterative solver failed.

Return values
The incident shock wave mach number M or `NAN` on error.

3.17 Quasi-2D Conical Flow

3.17.1 Introduction

This function group computes a solution to the *Taylor-Maccoll* equation for axis symmetric flow around a cone as for example described in [**JDAnderson1982**], that is:

$$\frac{\gamma-1}{2}\left[1-V_r'^2-\left(\frac{dV_r'}{d\theta}\right)^2\right]\left[2V_r'+\frac{dV_r'}{d\theta}\cot\theta+\frac{d^2V_r'}{d\theta^2}\right]-\frac{V_r'}{d\theta}\left[V_r'\frac{dV_r'}{d\theta}+\frac{dV_r'}{d\theta}\frac{d^2V_r'}{d\theta^2}\right]=0$$

The solution to the above can be found given any combination of: the deflection surface angle θ of the cone, an approximated wave angle δ (i.e. using oblique shock relations) and the upstream mach number M. The solution is computed numerically using a Runge-Kutta method of the 4th order.

The following properties are computed:

Table 15: Conic shock properties

I	Property	Description
1	M	Mach number
2	θ	Shock angle (°)
3	σ	Cone angle (°)

3.17.2 `arscomw` - Conical flow given M and δ

```
SUBROUTINE AR17AA(G,M,W,RESULT,IERR)
  DOUBLE PRECISION G, M, W, RESULT(3)
  INTEGER IERR
```

Solve the conical flow given a mach number M and wave angle δ.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

`W` **- REAL** *ENTRY:* Wave angle $0 < \delta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 15.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

-3 Given δ out of range.

-4 Iterative solver failed.

3.17.3 `arscoms` - Conical flow given M and θ

```
SUBROUTINE AR17AB(G,M,S,RESULT,IERR)
  DOUBLE PRECISION G, M, S, RESULT(3)
  INTEGER IERR
```

Solve the conical flow given a mach number M and deflection surface angle θ.

Performance
Iterative

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M \geq 1$.

`S` **- REAL** *ENTRY:* Deflection surface angle $0 < \theta < 90$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 15.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

-3 Given θ out of range.

-4 Iterative solver failed.

3.18 Moving Normal Shock Waves

Moving normal shock waves refer to normal shock waves in linear motion axis symmetric to the flow direction. Two sets of functions are available, one computes the thermodynamic properties of moving normal shock waves as described and the other set computes the mechanical/dynamic properties of such shock waves. These are separated as the functions computing the mechanical properties will require additional arguments in comparison to the thermodynamic property functions.

3.19 Moving Normal Shock Waves (Thermodynamic Properties)

3.19.1 Introduction

This function group computes the thermodynamic properties of a moving normal shock wave.
The following properties are computed:

Table 16: Moving normal shock wave thermodynamic properties

I	Property	Description
1	M	Mach number
2	p_2/p_1	Pressure ratio
3	ρ_2/ρ_1	Density ratio
4	T_2/T_1	Temperature ratio

3.19.2 `arsmnsm` - Moving shock relations given M

```
SUBROUTINE AR18AA(G,M,RESULT,IERR)
  DOUBLE PRECISION G, M, RESULT(4)
  INTEGER IERR
```

Compute the moving normal shock wave relations given the mach number M.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $M > 0$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 16.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.19.3 `arsmnsp` - Moving shock relations given p_2/p_1

```
SUBROUTINE AR18AB(G,P,RESULT,IERR)
  DOUBLE PRECISION G, P, RESULT(4)
  INTEGER IERR
```

Compute the moving normal shock wave relations given the pressure ratio p_2/p_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`P` **- REAL** *ENTRY:* Pressure ratio $p_2/p_1 > (1-\gamma)/(\gamma+1)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 16.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given p_2/p_1 out of range.

3.19.4 `arsmnsd` - Moving shock relations given ρ_2/ρ_1

```
SUBROUTINE AR18AC(G,P,RESULT,IERR)
  DOUBLE PRECISION G, D, RESULT(4)
  INTEGER IERR
```

Compute the moving normal shock wave relations given the density ratio ρ_2/ρ_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`D` **- REAL** *ENTRY:* Density ratio $\rho_2/\rho_1 > -(\gamma^2-1)/(\gamma^2+1)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 16.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given ρ_2/ρ_1 out of range.

3.19.5 `arsmnst` - Moving shock relations given T_2/T_1

```
SUBROUTINE AR18AD(G,T,RESULT,IERR)
  DOUBLE PRECISION G, T, RESULT(4)
  INTEGER IERR
```

Compute the moving normal shock wave relations given the temperature ratio T_2/T_1.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`T` **- REAL** *ENTRY:* Temperature ratio $T_2/T_1 > (\gamma^2+1)/(\gamma+1)^2$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 16.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Computed M out of range.

-3 Given T_2/T_1 out of range.

3.20 Moving Normal Shock Waves (Dynamic Properties)

3.20.1 Introduction

This function group computes the mechanic/dynamic properties of a moving normal shock wave. These not only require the specific heat constant γ but also the speed of sound in the latter medium a_0.

The following properties are computed:

Table 17: Moving normal shock waves dynamic properties

I	**Property**	**Description**
1	p_2/p_1	Pressure ratio
2	V	Velocity
3	U_P	Max-motion velocity, downstream

3.20.2 `arsmnvp` - Dynamic moving shock relations given p_2/p_1

```
SUBROUTINE AR19AA(G,A0,P,RESULT,IERR)
  DOUBLE PRECISION G, A0, P, RESULT(3)
  INTEGER IERR
```

Compute the dynamic moving normal shock wave relations given the the pressure ratio p_2/p_1 and the speed of sound a_0 of the medium.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`A0` **- REAL** *ENTRY:* Speed of sound in the medium $a_0 > 0$.

`P` **- REAL** *ENTRY:* Pressure ratio $p_2/p_1 > (\gamma - 1)/(\gamma + 1)$.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 17.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given a_0 out of range.

-3 Given p_2/p_1 out of range.

3.20.3 `arsmnvw` - Dynamic moving shock relations given V

```
SUBROUTINE AR19AB(G,A0,W,RESULT,IERR)
  DOUBLE PRECISION G, A0, W, RESULT(3)
  INTEGER IERR
```

Compute the dynamic moving normal shock wave relations given the the velocity V and the speed of sound a_0 of the medium.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`A0` - **REAL** *ENTRY:* Speed of sound in the medium $a_0 > 0$.

`W` - **REAL** *ENTRY:* Velocity $V > 0$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 17.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given a_0 out of range.

-3 Computed p_2/p_1 out of range.

-4 Given V out of range.

3.20.4 `arsmnvup` - Dynamic moving shock relations given U_p

```
SUBROUTINE AR19AC(G,A0,UP,RESULT,IERR)
  DOUBLE PRECISION G, A0, UP, RESULT(3)
  INTEGER IERR
```

Compute the dynamic moving normal shock wave relations given the mass-motion velocity U_p and the speed of sound a_0 of the medium.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`A0` - **REAL** *ENTRY:* Speed of sound in the medium $a_0 > 0$.

`UP` - **REAL** *ENTRY:* Max-motion velocity $U_p > -(2\sqrt{2}a_0\sqrt{\gamma(\gamma^2+2\gamma-1)})/(\gamma+1)$.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 17.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given a_0 out of range.

-3 Computed p_2/p_1 out of range.

-4 Given U_p out of range.

3.21 Karman-Tsien Pressure Correction Coefficient

3.21.1 Introduction

The *Karman-Tsien* pressure correlation coefficient is defined as:

$$C_p = \frac{C_{p0}}{\sqrt{1-M^2} + \frac{1}{2}\left(\frac{M^2}{1+\sqrt{1-M^2}}\right) C_{p0}}$$

Where C_{p0} the incompressebility coefficient.

The following properties are computed:

Table 18: Karman-Tsien property table

I	Property	Description
1	M	Mach number
2	C_p	Pressure correction coefficient
3	C_{p0}	Incompressebility coefficient

3.21.2 `arckarcp` - Karman-Tsien pressure correction given M and C_{p0}

```
SUBROUTINE AR20AA(G,M,C,RESULT,IERR)
  DOUBLE PRECISION G, M, C, RESULT(3)
  INTEGER IERR
```

Compute the Karman-Tsien pressure correction coefficient relations given the mach number M and the the incompressebility coefficient C_{p0}.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` **- REAL** *ENTRY:* Incompressebility coefficient C_{p0}.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 18.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.21.3 `arckarci` - Karman-Tsien pressure correction given M and C_p

```
SUBROUTINE AR20AB(G,M,CP,RESULT,IERR)
  DOUBLE PRECISION G, M, CP, RESULT(3)
  INTEGER IERR
```

Compute the Karman-Tsien pressure correction coefficient relations given the mach number M and the the Karman-Tsien pressure correction coefficient C_p.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` **- REAL** *ENTRY:* Pressure correction coefficient C_p.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 18.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \le 1$.

-2 Given M out of range.

3.22 Laitone Pressure Correction Coefficient

3.22.1 Introduction

The *Laitone* pressure correlation coefficient is defined as:

$$C_p = \frac{C_{p0}}{\sqrt{1-M^2} + \left(M^2 \frac{1+(\gamma-1)/2\times M^2}{2\sqrt{1-M^2}}\right) C_{p0}}$$

Where C_{p0} the incompressebility coefficient.

The following properties are computed:

Table 19: Laitone property table

I	Property	Description
1	M	Mach number
2	C_p	Pressure correction coefficient
3	C_{p0}	Incompressebility coefficient

3.22.2 `arclaicp` - Laitone pressure correction given M and C_{p0}

```
SUBROUTINE AR21AA(G,M,C,RESULT,IERR)
  DOUBLE PRECISION G, M, C, RESULT(3)
  INTEGER IERR
```

Compute the Laitone pressure correction coefficient relations given the mach number M and the the incompressebility coefficient C_{p0}.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` **- REAL** *ENTRY:* Incompressebility coefficient C_{p0}.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 19.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.22.3 `arclaici` - Laitone pressure correction given M and C_p

```
SUBROUTINE AR21AB(G,M,CP,RESULT,IERR)
  DOUBLE PRECISION G, M, CP, RESULT(3)
  INTEGER IERR
```

Compute the Laitone pressure correction coefficient relations given the mach number M and the the Laitone pressure correction coefficient C_p.

Performance
Fixed

Arguments

`G` - **REAL** *ENTRY:* Specific heat constant γ.

`M` - **REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` - **REAL** *ENTRY:* Pressure correction coefficient C_p.

`RESULT` - **ARRAY OF REAL** *EXIT:* Array with result properties as described in 19.

`IERR` - **INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.23 Prandtl-Glauert Pressure Correction Coefficient

3.23.1 Introduction

The *Prandtl-Glauert* pressure correlation coefficient is defined as:

$$C_p = \frac{C_{p0}}{\sqrt{1-M^2}}$$

Where C_{p0} the incompressebility coefficient.

The following properties are computed:

Table 20: Prandtl-Glauert property table

I	Property	Description
1	M	Mach number
2	C_p	Pressure correction coefficient
3	C_{p0}	Incompressebility coefficient

3.23.2 `arcpglcp` - Prandtl-Glauert pressure correction given M and C_{p0}

```
SUBROUTINE AR22AA(G,M,C,RESULT,IERR)
  DOUBLE PRECISION G, M, C, RESULT(3)
  INTEGER IERR
```

Compute the Prandtl-Glauert pressure correction coefficient relations given the mach number M and the the incompressebility coefficient C_{p0}.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` **- REAL** *ENTRY:* Incompressebility coefficient C_{p0}.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 20.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.

3.23.3 `arcpglci` - Prandtl-Glauert pressure correction given M and C_p

```
SUBROUTINE AR22AB(G,M,CP,RESULT,IERR)
  DOUBLE PRECISION G, M, CP, RESULT(3)
  INTEGER IERR
```

Compute the Prandtl-Glauert pressure correction coefficient relations given the mach number M and the the Prandtl-Glauert pressure correction coefficient C_p.

Performance
Fixed

Arguments

`G` **- REAL** *ENTRY:* Specific heat constant γ.

`M` **- REAL** *ENTRY:* Mach number $0 < M < 0.8$.

`CP` **- REAL** *ENTRY:* Pressure correction coefficient C_p.

`RESULT` **- ARRAY OF REAL** *EXIT:* Array with result properties as described in 20.

`IERR` **- INTEGER** *EXIT:* (*optional*) Return status code.

Status codes

-1 Specific heat ratio $\gamma \leq 1$.

-2 Given M out of range.