NAG Toolbox for MATLAB

d03pu

1 Purpose

d03pu calculates a numerical flux function using Roe's Approximate Riemann Solver for the Euler equations in conservative form. It is designed primarily for use with the upwind discretization schemes d03pf, d03pl or d03ps, but may also be applicable to other conservative upwind schemes requiring numerical flux functions.

2 Syntax

```
[flux, ifail] = d03pu(uleft, uright, gamma)
```

3 Description

d03pu calculates a numerical flux function at a single spatial point using Roe's Approximate Riemann Solver (see Roe 1981) for the Euler equations (for a perfect gas) in conservative form. You must supply the *left* and *right* solution values at the point where the numerical flux is required, i.e., the initial left and right states of the Riemann problem defined below.

In the functions d03pf, d03pl and d03ps, the left and right solution values are derived automatically from the solution values at adjacent spatial points and supplied to the (sub)program argument user-supplied (sub)program **numflx** from which you may call d03pu.

The Euler equations for a perfect gas in conservative form are:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0, \tag{1}$$

with

$$U = \begin{bmatrix} \rho \\ m \\ e \end{bmatrix} \text{ and } F = \begin{bmatrix} m \\ \frac{m^2}{\rho} + (\gamma - 1)\left(e - \frac{m^2}{2\rho}\right) \\ \frac{me}{\rho} + \frac{m}{\rho}(\gamma - 1)\left(e - \frac{m^2}{2\rho}\right) \end{bmatrix}, \tag{2}$$

where ρ is the density, m is the momentum, e is the specific total energy, and γ is the (constant) ratio of specific heats. The pressure p is given by

$$p = (\gamma - 1)\left(e - \frac{\rho u^2}{2}\right), \tag{3}$$

where $u = m/\rho$ is the velocity.

The function calculates the Roe approximation to the numerical flux function $F(U_L, U_R) = F(U^*(U_L, U_R))$, where $U = U_L$ and $U = U_R$ are the left and right solution values, and $U^*(U_L, U_R)$ is the intermediate state $\omega(0)$ arising from the similarity solution $U(y, t) = \omega(y/t)$ of the Riemann problem defined by

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial y} = 0, \tag{4}$$

with U and F as in (2), and initial piecewise constant values $U = U_L$ for $y < 0$ and $U = U_R$ for $y > 0$. The spatial domain is $-\infty < y < \infty$, where $y = 0$ is the point at which the numerical flux is required. This implementation of Roe's scheme for the Euler equations uses the so-called parameter-vector method described in Roe 1981.

4 References

LeVeque R J 1990 *Numerical Methods for Conservation Laws* Birkhäuser Verlag

Quirk J J 1994 A contribution to the great Riemann solver debate *Internat. J. Numer. Methods Fluids* **18** 555–574

Roe P L 1981 Approximate Riemann solvers, parameter vectors, and difference schemes *J. Comput. Phys.* **43** 357–372

5 Parameters

5.1 Compulsory Input Parameters

1: **uleft(3) – double array**

uleft(i) must contain the left value of the component U_i, for $i = 1, 2, 3$. That is, **uleft**(1) must contain the left value of ρ, **uleft**(2) must contain the left value of m and **uleft**(3) must contain the left value of e.

Constraints:

uleft$(1) \geq 0.0$;
Left pressure, $pl \geq 0.0$, where pl is calculated using (3).

2: **uright(3) – double array**

uright(i) must contain the right value of the component U_i, for $i = 1, 2, 3$. That is, **uright**(1) must contain the right value of ρ, **uright**(2) must contain the right value of m and **uright**(3) must contain the right value of e.

Constraints:

uright$(1) \geq 0.0$;
Right pressure, $pr \geq 0.0$, where pr is calculated using (3).

3: **gamma – double scalar**

The ratio of specific heats, γ.

Constraint: **gamma** > 0.0.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **flux(3) – double array**

flux(i) contains the numerical flux component $\hat{F}_i$, for $i = 1, 2, 3$.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **gamma** ≤ 0.0.

ifail = 2

On entry, the left and/or right density or pressure value is less than 0.0.

7 Accuracy

d03pu performs an exact calculation of the Roe numerical flux function, and so the result will be accurate to ***machine precision***.

8 Further Comments

d03pu must only be used to calculate the numerical flux for the Euler equations in exactly the form given by (2), with **uleft**(i) and **uright**(i) containing the left and right values of ρ, m and e, for $i = 1, 2, 3$, respectively. It should be noted that Roe's scheme, in common with all Riemann solvers, may be unsuitable for some problems (see Quirk 1994 for examples). In particular Roe's scheme does not satisfy an 'entropy condition' which guarantees that the approximate solution of the PDE converges to the correct physical solution, and hence it may admit non-physical solutions such as expansion shocks. The algorithm used in this function does not detect or correct any entropy violation. The time taken is independent of the input parameters.

9 Example

```
uleft = [1;
     0;
     2.5];
uright = [1;
     0;
     2.5];
gamma = 1.4;
[flux, ifail] = d03pu(uleft, uright, gamma)
```

```
flux =
         0
    1.0000
         0
ifail =
           0
```