NAG Toolbox for MATLAB

d03pw

1 Purpose

d03pw calculates a numerical flux function using a modified HLL (Harten–Lax–van Leer) 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] = d03pw(uleft, uright, gamma)
```

3 Description

d03pw calculates a numerical flux function at a single spatial point using a modified HLL (Harten–Lax–van Leer) Approximate Riemann Solver (see Toro 1992, Toro 1996 and Toro *et al.* 1994) 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 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 d03pw.

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

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

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 an 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.

4 References

Toro E F 1992 The weighted average flux method applied to the Euler equations *Phil. Trans. R. Soc. Lond.* **A341** 499–530

Toro E F 1996 *Riemann Solvers and Upwind Methods for Fluid Dynamics* Springer–Verlag

Toro E F, Spruce M and Spears W 1994 Restoration of the contact surface in the HLL Riemann solver *J. Shock Waves* **4** 25–34

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 derived pressure value is less than 0.0.

7 Accuracy

d03pw performs an exact calculation of the HLL (Harten–Lax–van Leer) numerical flux function, and so the result will be accurate to ***machine precision***.

8 Further Comments

d03pw 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. The time taken is independent of the input parameters.

9 Example

```
uleft = [5.99924;
     117.5701059;
     2304.275075187626];
uright = [5.99924;
     117.5701059;
     2304.275075187626];
gamma = 1.4;
[flux, ifail] = d03pw(uleft, uright, gamma)
```

```
flux =
   1.0e+04 *
    0.0118
    0.2765
    5.4190
ifail =
           0
```