NAG Toolbox for MATLAB

d03pv

1 Purpose

d03pv calculates a numerical flux function using Osher'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] = d03pv(uleft, uright, gamma, path)
```

3 Description

d03pv calculates a numerical flux function at a single spatial point using Osher's Approximate Riemann Solver (see Hemker and Spekreijse 1986 and Pennington and Berzins 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 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 d03pv.

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 the Osher 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. Osher's solver carries out an integration along a path in the phase space of U consisting of subpaths which are piecewise parallel to the eigenvectors of the Jacobian of the PDE system. There are two variants of the Osher solver termed O (original) and P (physical), which differ in the order in which the subpaths are taken. The P-variant is generally more efficient, but in some rare cases may fail (see Hemker and Spekreijse 1986 for details). The parameter **path** specifies which variant is to be used. The algorithm for Osher's solver for the Euler equations is given in detail in the Appendix of Pennington and Berzins 1994.

4 References

Hemker P W and Spekreijse S P 1986 Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations *Applied Numerical Mathematics* **2** 475–493

Pennington S V and Berzins M 1994 New NAG Library software for first-order partial differential equations *ACM Trans. Math. Softw.* **20** 63–99

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

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) ≥ 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) ≥ 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.

4: **path – string**

The variant of the Osher scheme.

path = 'O'

Original.

path = 'P'

Physical.

Constraint: **path** = 'O' or 'P'.

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,
or **path** $\neq$ 'O' or 'P'.

ifail $= 2$

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

7 Accuracy

d03pv performs an exact calculation of the Osher numerical flux function, and so the result will be accurate to ***machine precision***.

8 Further Comments

d03pv 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 Osher's scheme, in common with all Riemann solvers, may be unsuitable for some problems (see Quirk 1994 for examples). The time taken depends on the input parameter **path** and on the left and right solution values, since inclusion of each subpath depends on the signs of the eigenvalues. In general this cannot be determined in advance.

9 Example

```
uleft = [1;
     0;
     2.5];
uright = [1;
     0;
     2.5];
gamma = 1.4;
path = 'P';
[flux, ifail] = d03pv(uleft, uright, gamma, path)
```

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