NAG Library Routine Document

D03PVF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

D03PVF 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 D03PFF, D03PLF or D03PSF, but may also be applicable to other conservative upwind schemes requiring numerical flux functions.

2 Specification

```
SUBROUTINE D03PVF (ULEFT, URIGHT, GAMMA, PATH, FLUX, IFAIL)
INTEGER             IFAIL
REAL (KIND=nag_wp) ULEFT(3), URIGHT(3), GAMMA, FLUX(3)
CHARACTER(1)        PATH
```

3 Description

D03PVF 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 routines D03PFF, D03PLF and D03PSF, the left and right solution values are derived automatically from the solution values at adjacent spatial points and supplied to the subroutine argument NUMFLX from which you may call D03PVF.

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 routine 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 argument 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 Arguments

1: ULEFT(3) – REAL (KIND=nag_wp) array *Input*

On entry: 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) – REAL (KIND=nag_wp) array *Input*

On entry: 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 – REAL (KIND=nag_wp) *Input*

On entry: the ratio of specific heats, γ.

Constraint: GAMMA > 0.0.

4: PATH – CHARACTER(1) *Input*

On entry: the variant of the Osher scheme.

PATH = 'O'
Original.

PATH = 'P'
Physical.

Constraint: PATH = 'O' or 'P'.

5: FLUX(3) – REAL (KIND=nag_wp) array *Output*

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

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

Note: if the left and/or right values of ρ or p (from (3)) are found to be negative, then the routine will terminate with an error exit (IFAIL $= 2$). If the routine is being called from the NUMFLX etc., then a **soft fail** option (IFAIL $= 1$ or -1) is recommended so that a recalculation of the current time step can be forced using the NUMFLX argument IRES (see D03PFF or D03PLF).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

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.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

D03PVF performs an exact calculation of the Osher numerical flux function, and so the result will be accurate to ***machine precision***.

8 Parallelism and Performance

D03PVF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in How to Use the NAG Library and its Documentation for more information on thread safety.

D03PVF is not threaded in any implementation.

9 Further Comments

D03PVF 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 argument 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.

10 Example

See Section 10 in D03PLF.