NAG C Library Function Document

nag_pde_parab_1d_euler_roe (d03puc)

1 Purpose

nag_pde_parab_1d_euler_roe (d03puc) 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 discretisation schemes nag_pde_parab_1d_cd (d03pfc), nag_pde_parab_1d_cd_ode (d03plc) or nag_pde_parab_1d_cd_ode_remesh (d03psc), but may also be applicable to other conservative upwind schemes requiring numerical flux functions.

2 Specification

```
void nag_pde_parab_1d_euler_roe (const double uleft[], const double uright[],
     double gamma, double flux[], Nag_D03_Save *saved, NagError *fail)
```

3 Description

nag_pde_parab_1d_euler_roe (d03puc) calculates a numerical flux function at a single spatial point using Roe's Approximate Riemann Solver (Roe (1981)) for the Euler equations (for a perfect gas) in conservative form. The user 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 nag_pde_parab_1d_cd (d03pfc), nag_pde_parab_1d_cd_ode (d03plc) and nag_pde_parab_1d_cd_ode_remesh (d03psc), the left and right solution values are derived automatically from the solution values at adjacent spatial points and supplied to the function argument **numflx** from which the user may call nag_pde_parab_1d_euler_roe (d03puc).

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 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

1: **uleft**[3] – const double *Input*

On entry: **uleft**$[i-1]$ must contain the left value of the component U_i for $i = 1, 2, 3$. That is, **uleft**[0] must contain the left value of ρ, **uleft**[1] must contain the left value of m and **uleft**[2] must contain the left value of e.

Constraints:

uleft[0] ≥ 0.0;
Left pressure, pl, calculated using (3) ≥ 0.0.

2: **uright**[3] – const double *Input*

On entry: **uright**$[i-1]$ must contain the right value of the component U_i for $i = 1, 2, 3$. That is, **uright**[0] must contain the right value of ρ, **uright**[1] must contain the right value of m and **uright**[2] must contain the right value of e.

Constraints:

uright[0] ≥ 0.0;
Right pressure, pr, calculated using (3) ≥ 0.0.

3: **gamma** – double *Input*

On entry: the ratio of specific heats γ.

Constraint: **gamma** > 0.0.

4: **flux**[3] – double *Output*

On exit: **flux**$[i-1]$ contains the numerical flux component $\hat{F}_i$ for $i = 1, 2, 3$.

5: **saved** – Nag_D03_Save *Input*

Note: saved is a NAG defined structure. See Section 2.2.1.1 of the Essential Introduction.

On entry: data concerning the computation required by nag_pde_parab_1d_euler_roe (d03puc) and passed through to **numflx** from one of the integrator functions nag_pde_parab_1d_cd (d03pfc), nag_pde_parab_1d_cd_ode (d03plc), or nag_pde_parab_1d_cd_ode_remesh (d03psc).

6: **fail** – NagError * *Input/Output*

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_REAL

Right pressure value $pr < 0.0$: $pr = \langle value \rangle$.

Left pressure value $pl < 0.0$: $pl = \langle value \rangle$.

On entry, **uright**[0] < 0.0: **uright**[0] $= \langle value \rangle$.

On entry, **uleft**[0] < 0.0: **uleft**[0] $= \langle value \rangle$.

On entry, **gamma** = $\langle value \rangle$.
Constraint: **gamma** > 0.0.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ had an illegal value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

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

8 Further Comments

The function must only be used to calculate the numerical flux for the Euler equations in exactly the form given by (2), with **uleft**$[i-1]$ and **uright**$[i-1]$ 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

See Section 9 of the document for nag_pde_parab_1d_cd_ode (d03plc).