NAG 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 discretization 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

```c
#include <nag.h>
#include <nagd03.h>

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 (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 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 you 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

\[
U = \begin{bmatrix} \rho \\ m \\ e \end{bmatrix} \quad \text{and} \quad F = \begin{bmatrix} \rho \frac{u_2}{\rho} + (\gamma - 1)\left(e - \frac{u_2^2}{2\rho}\right) \\ \rho u_1 \frac{u_2}{\rho} + \frac{\rho}{\gamma - 1}\left(e - \frac{u_2^2}{2\rho}\right) \end{bmatrix}, \tag{2}
\]

where \(\rho\) is the density, \(m\) is the momentum, \(e\) is the specific total energy, and \(\gamma\) 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

\[
F(U_L, U_R) = F(U^*(U_L, U_R)), \quad \text{where} \ U = U_L \quad \text{and} \ U = U_R \quad \text{are the left and right solution values,}\n\]

and

\[
U^*(U_L, U_R) \quad \text{is the intermediate state} \ \omega(0) \ \text{arising from the similarity solution} \ U(y, t) = \omega(y/t) \ \text{of the Riemann problem defined by}
\]

\[
\frac{\partial U}{\partial t} + \frac{\partial F}{\partial y} = 0, \tag{4}
\]
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 argument-vector method described in Roe (1981).

4 References


5 Arguments

1: \texttt{uleft[3]} – const double \hspace{1cm} Input

\textit{On entry:} \texttt{uleft[i - 1]} must contain the left value of the component \( U_i \), for \( i = 1, 2, 3 \). That is, \texttt{uleft[0]} must contain the left value of \( \rho \), \texttt{uleft[1]} must contain the left value of \( m \) and \texttt{uleft[2]} must contain the left value of \( e \).

\textit{Constraints:}

\begin{align*}
\texttt{uleft[0]} & \geq 0.0; \\
\text{Left pressure, } p_l & \geq 0.0, \text{ where } p_l \text{ is calculated using (3)}.
\end{align*}

2: \texttt{uright[3]} – const double \hspace{1cm} Input

\textit{On entry:} \texttt{uright[i - 1]} must contain the right value of the component \( U_i \), for \( i = 1, 2, 3 \). That is, \texttt{uright[0]} must contain the right value of \( \rho \), \texttt{uright[1]} must contain the right value of \( m \) and \texttt{uright[2]} must contain the right value of \( e \).

\textit{Constraints:}

\begin{align*}
\texttt{uright[0]} & \geq 0.0; \\
\text{Right pressure, } p_r & \geq 0.0, \text{ where } p_r \text{ is calculated using (3)}.
\end{align*}

3: \texttt{gamma} – double \hspace{1cm} Input

\textit{On entry:} the ratio of specific heats, \( \gamma \).

\textit{Constraint:} \texttt{gamma} > 0.0.

4: \texttt{flux[3]} – double \hspace{1cm} Output

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

5: \texttt{saved} – Nag_D03_Save * \hspace{1cm} Communication Structure

\texttt{saved} may contain data concerning the computation required by \texttt{nag_pde_parab_1d_euler_roe} (d03puc) as passed through to \texttt{numfix} from one of the integrator functions \texttt{nag_pde_parab_1d_cd} (d03pfc), \texttt{nag_pde_parab_1d_cd_ode} (d03plc) or \texttt{nag_pde_parab_1d_CD_ode_remesh} (d03psc). You should not change the components of \texttt{saved}.

6: \texttt{fail} – NagError * \hspace{1cm} Input/Output

The NAG error argument (see Section 3.6 in the Essential Introduction).
6 Error Indicators and Warnings

NE_ALLOC_FAIL
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM
On entry, argument \(\text{value}\) 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 contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_REAL
Left pressure value \(pl < 0.0\): \(pl = \langle value\rangle\).
On entry, \(\text{gamma} = \langle value\rangle\).
Constraint: \(\text{gamma} > 0.0\).

On entry, \(\text{uleft}[0] = \langle value\rangle\).
Constraint: \(\text{uleft}[0] \geq 0.0\).

On entry, \(\text{uright}[0] = \langle value\rangle\).
Constraint: \(\text{uright}[0] \geq 0.0\).

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

7 Accuracy

\text{nag_pde_parab_1d_euler_roe} (d03puc) performs an exact calculation of the Roe numerical flux function,
and so the result will be accurate to \textit{machine precision}.

8 Parallelism and Performance

Not applicable.

9 Further Comments

\text{nag_pde_parab_1d_euler_roe} (d03puc) must only be used to calculate the numerical flux for the Euler
equations in exactly the form given by (2), with \text{uleft}[i-1] and \text{uright}[i-1] containing the left and
right values of \(\rho, m\) 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 arguments.
10 Example

See Section 10 in nag_pde_parab_1d_cd_ode (d03plc).