Consiglio Nazionale delle Ricerche

Istituto di Matematica Applicata e Tecnologie Informatiche

PUBBLICAZIONI

Enrico Bertolazzi, Gianmarco Manzini

DIAGONALLY IMPLICIT–EXPLICIT RUNGE KUTTA METHODS FOR MULTIDIMENSIONAL HYPERBOLIC SYSTEMS.
PART I: FORMULATION OF THE METHOD

N. 16-PV 2004

Diagonally Implicit–Explicit Runge Kutta Methods for Multidimensional Hyperbolic Systems. Part I: Formulation of the Method

Enrico Bertolazzi [a] Gianmarco Manzini [b]

[a] *Dipartimento di Ingegneria Meccanica e Strutturale, Università di Trento, via Mesiano 77, I – 38050 Trento, Italy*

[b] *Istituto di Matematica Applicata e Tecnologie Informatiche, IMATI – CNR, via Ferrata 1, I – 27100 Pavia, Italy*

Abstract

We propose a general framework for the semi-implicit discretization of multidimensional hyperbolic systems in conservative (divergence) form on unstructured grids. This approach is based on the method-of-line strategy, which decouples the discretization in time and space. The spatial divergence term of the flux function is discretized by an unstructured Finite Volume (FV) technique and the spatial accuracy is provided by including into the scheme a generic non-oscillatory reconstruction procedure which may attain orders higher than one. An original splitting of the numerical flux function into a convective and a non-convective part is introduced to discretize the time derivative by the Diagonally Implicit-Explicit Runge-Kutta (DIMEX-RK) time stepping scheme. The convective term is treated implicitly by mimicking the upwinding of a scalar linear flux, while the non-convective one is treated explicitly. Finally, several numerical examples illustrate the behavior of the method.

Key words: Finite Volume, Runge-Kutta, Implicit-Explicit, Partial Differential Equation, M-matrix, Unstructured Grid

1 Introduction

Convected-dominated flows are usually described in terms of the following multi-dimensional hyperbolic system in divergence conservative form

$$\frac{\partial}{\partial t}\mathbf{u} + \nabla \cdot \mathbf{F}(\mathbf{u}) = \mathbf{0}, \qquad \text{in } \Omega \times (0,\, T), \tag{1}$$

where Ω is a bounded open connected subset of $\mathbb{R}^d$, $d = 1, 2, 3$, $\mathbf{u}$ is a vector-valued function from $\mathbb{R}^d \times [0,\, \infty]$ into the open subset $\mathfrak{U} \subseteq \mathbb{R}^m$, and $\mathbf{F}(\mathbf{u})$ is a non-linear vector-valued mapping from $\mathfrak{U}$ into $\mathbb{R}^m$. Throughout the paper, we will refer to $\mathbf{u}$

as *the solution vector function*, to $\mathfrak{U}$ as *the set of admissible solution states*, and to $\mathbf{F}(\mathbf{u})$ as *the flux vector function*. As usual, the components $\mathbf{F}(\mathbf{u})$ are assumed smooth — say of class C^∞.

An Initial Boundary Value Problem (IBVP) is defined by providing system (1) with an initial solution $\mathbf{u}|_{t=0} = \mathbf{u}_0$, where $\mathbf{u}_0$ is a vector function from $\mathbb{R}^d$ into $\mathfrak{U} \subseteq \mathbb{R}^m$, and with a suitable set of problem dependent boundary conditions.

Let us denote the set of the d Jacobian matrices of the flux vector functions in (1) by

$$\mathbf{J}(\mathbf{u}) = \frac{\partial \mathbf{F}(\mathbf{u})}{\partial \mathbf{u}}, \qquad \text{for each } \mathbf{u} \in \mathfrak{U}.$$

Equation (1) is a multidimensional hyperbolic system of equations in divergence form if the matrix $\mathbf{J}(\mathbf{u}, \mathbf{n}) = \mathbf{n} \cdot \mathbf{J}(\mathbf{u})$ has m real eigenvalues $\lambda^{\min}(\mathbf{u}, \mathbf{n}) = \lambda_1(\mathbf{u}, \mathbf{n}) \leq \cdots \leq \lambda_m(\mathbf{u}, \mathbf{n}) = \lambda^{\max}(\mathbf{u}, \mathbf{n})$ and a complete set of eigenvectors for any $\mathbf{u} \in \mathfrak{U}$ and any non-zero vector $\mathbf{n} \in \mathbb{R}^d$. We will denote the set of the eigenvalues of the Jacobian matrix $\mathbf{n} \cdot \mathbf{J}(\mathbf{u})$ by $\Lambda(\mathbf{u}, \mathbf{n})$, and the minimum and maximum eigenvalues by $\lambda^{\min}(\mathbf{u}, \mathbf{n})$ and $\lambda^{\max}(\mathbf{u}, \mathbf{n})$. No strict hyperbolicity assumption will be retained in this work.

The approximation method that we investigate in this work can be naturally considered in the framework of the Implicit-Explicit Runge-Kutta (IMEX-RK) methods [1–3]. This approach is especially suitable to the circumstances which force the use of an implicit scheme even in evolutionary calculations. This occurs for instance when fast reactive processes take place in the flow or when the flow configuration features very high speed flows in some restricted regions, and moderate to low speed flow elsewhere. Regions of high speed flows may drastically reduce the time-step size allowable in an explicit formulation, consequently degrading the performance of a numerical flow solver from the viewpoint of the computational costs. On the other hand, an implicit first order discretization in time produces a poor accurate approximation of the time-dependent solution. Thus, high-order semi-implicit techniques should be devised to improve the overall accuracy of the model prediction.

Basically, we consider a decomposition of the physical flux function into a *convective* and a *non-convective* term, respectively denoted by $\mathbf{F}^{(c)}(\mathbf{u})$ and $\mathbf{F}^{(nc)}(\mathbf{u})$. Essentially, the IMEX-RK strategy that we propose consists in applying an implicit discretization to the convective term $\mathbf{F}^{(c)}(\mathbf{u})$ and an explicit one to the non-convective term $\mathbf{F}^{(nc)}(\mathbf{u})$. For this reason, we focus our attention on problems of the form (1) whose physical flux vector function satisfies the following formal assumption.

Assumption 1 *The flux vector function* $\mathbf{F}(\mathbf{u})$ *can be split as*

$$\mathbf{F}(\mathbf{u}) = \mathbf{F}^{(c)}(\mathbf{u}) + \mathbf{F}^{(nc)}(\mathbf{u}), \tag{2}$$

where the convective part takes the form

$$\mathbf{F}^{(c)}(\mathbf{u}) = \mathbf{u} \otimes \mathbf{v}(\mathbf{u}),$$

$\mathbf{v}(\mathbf{u})$ *being the* convective velocity field *and satisfying*

$$\lambda^{min}(\mathbf{u},\mathbf{n}) \leq \mathbf{n}\cdot\mathbf{v}(\mathbf{u}) \leq \lambda^{max}(\mathbf{u},\mathbf{n}), \qquad \textit{for any } \mathbf{u}\in\mathfrak{U}, \quad \textit{and } \mathbf{n}\in\mathbb{R}^d.$$

The symbol $\otimes$ indicates the standard tensor product, defined as follows. Given two matrices $\mathbf{A}$ and $\mathbf{B}$ of order $m\times n$ and $p\times q$, $\mathbf{A}\otimes\mathbf{B}$ is the block matrix of order $mp\times nq$ whose block i,j is $(\mathbf{A}\otimes\mathbf{B})_{i,j} = A_{ij}\mathbf{B}$. The tensor product has some noteworthy properties, see for instance Reference [4]. We just mention the one most used in the paper, thus $(\mathbf{A}\otimes\mathbf{B})(\mathbf{C}\otimes\mathbf{D}) = \mathbf{AC}\otimes\mathbf{BD}$, with $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ four generic matrices (with compatible dimensions).

Clearly, the functional form of $\mathbf{F}^{(c)}(\mathbf{u})$, $\mathbf{v}(\mathbf{u})$, and $\mathbf{F}^{(nc)}(\mathbf{u})$ is strictly problem dependent. Nonetheless, as pointed out in References [5,6], it turns out that the decomposition (2) does not rely on an arbitrary formal trick but has a thorough physical meaning. It is indeed the consequence of the assumption that the balance laws described by the model equations (1) are invariant under Galileian transformations. Furthermore, it is possible to demonstrate that this invariance property also implies the uniqueness of the convective/non-convective decomposition and also specifies the functional form of the (possibly non-linear) convective velocity field $\mathbf{v}(\mathbf{u})$.

For many important problems usually considered by the CFD community, we verified that $\mathbf{v}(\mathbf{u})$ is the real fluid velocity, $\mathbf{n}\cdot\mathbf{v}(\mathbf{u})$ is usually an eigenvalue of the Jacobian matrix $\mathbf{J}(\mathbf{u},\mathbf{n})$, and the non-convective term $\mathbf{F}^{(nc)}(\mathbf{u})$ is also given a precise physical interpretation. For instance, Assumption 1 is satisfied by the Compressible Gas Dynamic Euler Equations,

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \end{bmatrix}, \quad \mathbf{F}^{(c)}(\mathbf{u}) = \mathbf{u}\otimes\mathbf{v}(\mathbf{u}), \quad \mathbf{F}^{(nc)}(\mathbf{u}) = p(\mathbf{u}) \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \\ \mathbf{v}(\mathbf{u}) \end{bmatrix},$$

where ρ is the density, $\mathbf{v}$ the velocity, E the total energy, and $p(\mathbf{u})$ the pressure given by the thermodynamical relation

$$\frac{p}{\gamma-1} = \rho E - \frac{1}{2}\rho|\mathbf{v}|^2,$$

and $\gamma = 1.4$, as usual for air at standard condition. Notice that the splitting in Assumption 1 has a direct physical interpretation because the non-convective flux $\mathbf{F}^{(nc)}(\mathbf{u})$ is the pressure contribution to the physical flux.

Following a method-of-line approach, we numerically solve (1) by combining a shock-capturing cell-centered Finite Volume (FV) spatial discretization of the divergence term and an Diagonally Implicit-Explicit Runge-Kutta (DIMEX-RK) scheme for the time derivative.

If Assumption 1 holds for the physical flux, it becomes natural to wonder wheter a similar splitting may hold at the numerical flux function level. We formalize this issue in Section 2 by introducing the *numerical convective splitting*. Let us emphasize that our numerical convective splitting fits particularly well in the DIMEX-RK framework just because it allows to treat the numerical convective term implicitly, and the non-convective one explicitly. To this purpose, the numerical convective

part of the flux is defined via a suitable numerical upwind velocity field, in order to mimick the upwinding techniques utilized in difference schemes for the linear scalar advection equation. As pointed out in Reference [7], the integral FV form of the divergence operator applied to the linear advection flux produces a discrete operator that satisfies the properties of a *singular M-matrix*. The singularity can be eventually removed by the diagonal term arising from the discretization of the time derivative as well as by the introduction of the boundary conditions.

One of the major goals of this paper is to extend this issue to the more general case of a multidimensional non-linear hyperbolic system of equations. More precisely, we will show that some peculiar block structures can be identified in the time evolution matrix operator and that the matrices underlying them are still M-matrices. This fact has two noteworthy implications, that we briefly list below. First, we can prove the existence and uniqueness of the solution of the non-linear algebraic problem arising at each internal stage of the DIMEX-RK scheme. Then, we show that a less expensive iterative approximation at the correct time accuracy can be easily calculated. This fact makes it possible the development of simple and efficient resolution algorithms for the implicit algebraic equations corresponding to $\mathbf{F}^{(c)}(\mathbf{u})$.

We outline that this approach is compatible to many piecewise polynomial reconstruction techniques that are usually considered to improve the spatial order of the cell-centered FV methods — like MUSCL, ENO, etc — and many limiting procedures that are applied to control the numerical oscillations. Furthermore, it is worth noting that any numerical flux that can be re-formulated in accord with the numerical convective splitting can be effectively incorporated into our discretization framework. The requirement stated by the numerical convective splitting definition is not too restrictive because we prove that it is satisfied by many important numerical fluxes available in literature. We mention the HLLE-like fluxes and several widely known flux splitting schemes in the case of the Compressible Euler Equations, such as the Steger-Warming, the Van Leer and the family of the AUSM methods.

The paper is organized as follows. In Section 2 we introduce the basic definition of the *numerical convective splitting* upon which the family of discretizations investigated in this paper is actually based. We also show that several classes of shock-capturing methods fit within this definition. In Section 3 we formulate the semi-discrete FV method and in Section 4 the coupled DIMEX-RK FV integrators. In this last section, we also state the main theoretical properties of the full discrete formulation. The demonstration of these properties will be the subject of the forthcoming paper. Finally, in Section 5 conclusions are offered.

2 The numerical convective splitting

A FV scheme is usually defined by a numerical flux function, which will be denoted by $\boldsymbol{H}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$. Its entries are the solution states $\mathbf{u}_L$ and $\mathbf{u}_R$, and the vector

$\mathbf{n}$, which is perpendicular to the element interface over which the flux integral is estimated. We adopt the symbols $\mathbf{u}_L$ and $\mathbf{u}_R$ because these states are often considered as the "left" and the "right" states of a Riemann problem. The meaning of the words "left" and "right" is uniquely defined by using the orientation of the vector $\mathbf{n}$, which is supposed to be always pointing from the left side of a cell interface to the right one.

Following the physical convective/non-convective decomposition of Assumption 1, we consider numerical fluxes that can be decomposed as the sum of two terms, namely $\boldsymbol{H}^{(c)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ and $\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$. The former one, $\boldsymbol{H}^{(c)}$, is the numerical correspondent of $\mathbf{F}^{(c)}$ and the latter one, $\boldsymbol{H}^{(nc)}$, of $\mathbf{F}^{(nc)}$. Let us introduce the formal definition of *numerical convective splitting*.

Definition 2 *A numerical flux admits a numerical convective splitting if it can be decomposed into the sum of a convective and non-convective part,*

$$\boldsymbol{H}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = \boldsymbol{H}^{(c)}(\mathbf{u}, \mathbf{v}, \mathbf{n}) + \boldsymbol{H}^{(nc)}(\mathbf{u}, \mathbf{v}, \mathbf{n}).$$

The convective part takes the form

$$\boldsymbol{H}^{(c)}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = \mathsf{a}(\mathbf{u}, \mathbf{v}, \mathbf{n})\mathbf{u} - \mathsf{a}(\mathbf{v}, \mathbf{u}, -\mathbf{n})\mathbf{v}, \tag{3}$$

and is such that for each $\mathbf{u}, \mathbf{v} \in \mathfrak{U}$ *and* $\mathbf{n} \in \mathbb{R}^d$ *with* $\|\mathbf{n}\| = 1$ *the numerical convective velocity satisfies the following conditions:*

(i) $\mathsf{a}(\mathbf{u}, \mathbf{v}, \mathbf{n}) \geq 0$ *(non-negativity);*
(ii) $|\mathsf{a}(\mathbf{u}, \mathbf{v}, \mathbf{n}) - \mathsf{a}(\mathbf{u}', \mathbf{v}', \mathbf{n})| \leq L\left(\|\mathbf{u} - \mathbf{u}'\| + \|\mathbf{v} - \mathbf{v}'\|\right)$ *(Lipschitz continuity);*
(iii) $\mathbf{v}(\mathbf{u}) \cdot \mathbf{n} = \mathsf{a}(\mathbf{u}, \mathbf{u}, \mathbf{n}) - \mathsf{a}(\mathbf{u}, \mathbf{u}, -\mathbf{n})$ *(consistency).*

Thus, we formally require that both the convective and the non-convective part of the numerical flux function $\boldsymbol{H}(\mathbf{u}, \mathbf{v}, \mathbf{n})$ satisfies a regularity condition in the arguments $\mathbf{u}$ and $\mathbf{v}$ and that a stronger consistency condition holds on the convective flux function in place of the usual one, thus $\mathbf{F}(\mathbf{u}) \cdot \mathbf{n} = \boldsymbol{H}(\mathbf{u}, \mathbf{u}, \mathbf{n})$.

Notice that in the scalar multi-dimensional linear case, a numerical flux function in accord with Definition 2 simply reduces to the linear upwind formula [7]. Let us consider the physical flux function $\mathbf{F}(\mathbf{u}) = \mathbf{v}\mathbf{u}$, where $\mathbf{u}$ is to be taken in this case as a scalar quantity rigidly advected by the constant velocity field $\mathbf{v}$, and introduce the standard upwind projections on the normal vector $\mathbf{n}$, that are $a(\mathbf{n})^{\pm} = (\mathbf{v} \cdot \mathbf{n} \pm |\mathbf{v} \cdot \mathbf{n}|)/2$. Since $\mathbf{F}^{(nc)}(\mathbf{u}) = 0$, it obviously holds that $\boldsymbol{H}^{(nc)}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = 0$, and we can clearly identify $\mathsf{a}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = a(\mathbf{n})^{+}$ because $a(\mathbf{n})^{-} = -a(-\mathbf{n})^{+}$.

Several important families of numerical fluxes proposed in the literature of the last two decades satisfy Definition 2. In the rest of this section, we briefly review some of the most important numerical flux formulae that are widely used by the CFD community. The first two formulae are given for a general physical flux function, $\mathbf{F}(\mathbf{u})$, while the others are specifically derived from the flux splitting schemes for the multi-dimensional Compressible Euler Equations.

Table 1
Some possible choices of $S(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$

$S(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$	Reference
$\max\{\lambda_L^{max}, \lambda_R^{max}, 0\}$	[12,13]
$\max\left\{\widetilde{\lambda_{LR}^{max}}, \lambda_R^{max}, 0\right\}$	[9,10]
$\max\{\sigma_L, \sigma_R\}$	[11]

2.1 Rusanov-like numerical fluxes

The members of the family of Rusanov-like numerical fluxes take the following *viscosity form*

$$\boldsymbol{H}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \frac{1}{2}\left[\mathbf{n}\cdot\mathbf{F}(\mathbf{u}_L) + \mathbf{n}\cdot\mathbf{F}(\mathbf{u}_R)\right] - \frac{1}{2}\mathbf{Q}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})(\mathbf{u}_R - \mathbf{u}_L)$$

where the diagonal numerical viscosity tensor $\mathbf{Q}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ is a scalar multiple of the $m \times m$ identity matrix $\mathbf{I}_m$, thus

$$\mathbf{Q}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \xi(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})\mathbf{I}_m.$$

These schemes can be rewritten in numerical convective splitting, provided that

$$\boldsymbol{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \frac{1}{2}\left[\mathbf{v}(\mathbf{u}_L)\cdot\mathbf{n} + \xi(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})\right],$$

$$\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \frac{1}{2}(\mathbf{F}^{(nc)}(\mathbf{u}_L) + \mathbf{F}^{(nc)}(\mathbf{u}_R))\cdot\mathbf{n}.$$

2.2 HLLE-like numerical fluxes

These numerical fluxes are discussed in References [8–11]. They can be reformulated in numerical convective splitting by setting:

$$\boldsymbol{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \frac{S_R}{S_R - S_L}\left(v_{nL} - S_L\right),$$

$$\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \frac{(S_R\mathbf{F}^{(nc)}(\mathbf{u}_L) - S_L\mathbf{F}^{(nc)}(\mathbf{u}_R))\cdot\mathbf{n}}{S_R - S_L},$$

where $v_{nL} = \mathbf{v}(\mathbf{u}_L)\cdot\mathbf{n}$, $S_R = S(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ and $S_L = -S(\mathbf{u}_R, \mathbf{u}_L, -\mathbf{n})$, and some possible choices of the function $S(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ are listed in Table 1, with the corresponding reference. In Table 1, $\widetilde{\lambda_{LR}^{max}}$ is the maximum eigenvalue estimated using the intermediate Roe averaged state, and

The choice of $S(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ in the first row of Table 1 gives the so-called *local Lax-Friedrichs* numerical flux, while the one in the second row the "classical" HLLE scheme. This follows immediately by noting that $\lambda^{\max}(\mathbf{u}, \mathbf{n}) = -\lambda^{\min}(\mathbf{u}, -\mathbf{n})$ and using the identity $\min(a, b) = -\max(-a, -b)$, which holds for any pair of real numbers a and b.

2.3 Steger & Warming flux splitting

This splitting is described in Reference [14] and can be re-formulated in accord with the numerical convective splitting by setting $\mathsf{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \mathsf{a}_{SW}(\mathbf{u}_L, \mathbf{n})$ and $\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \boldsymbol{G}_{SW}(\mathbf{u}_L, \mathbf{n}) - \boldsymbol{G}_{SW}(\mathbf{u}_R, -\mathbf{n})$, where

$$\mathsf{a}_{SW}(\mathbf{u}, \mathbf{n}) = (2\gamma)^{-1}\left(2(\gamma-1)\lambda_1^+ + \lambda_2^+ + \lambda_3^+\right),$$

$$\boldsymbol{G}_{SW}(\mathbf{u}, \mathbf{n}) = \frac{p}{2c}\left[0,\ \left(\lambda_2^+ - \lambda_3^+\right)\mathbf{n},\ \lambda_2^+ - \lambda_3^+ + \frac{c}{\gamma}\left(\lambda_2^+ + \lambda_3^+ - 2\lambda_1^+\right)\right]^T,$$

$c = (\gamma p/\rho)^{1/2}$ denotes the frozen speed of sound, and

$$\lambda_1^+ = {v_n}^+, \quad \lambda_2^+ = (v_n + c)^+, \quad \lambda_3^+ = (v_n - c)^+.$$

2.4 Van Leer flux splitting

This splitting is described in References [15–17] and can be re-formulated in accord with the numerical convective splitting by setting $\mathsf{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \mathsf{a}_{VL}(\mathbf{u}_L, \mathbf{n})$ and $\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \boldsymbol{G}_{VL}(\mathbf{u}_L, \mathbf{n}) - \boldsymbol{G}_{VL}(\mathbf{u}_R, -\mathbf{n})$, where

$$\mathsf{a}_{VL}(\mathbf{u}, \mathbf{n}) = \begin{cases} (c + v_n)^2/(4c) & \text{if } |v_n| \le c, \\ (v_n)^+ & \text{otherwise.} \end{cases}$$

$$\boldsymbol{G}_{VL}(\mathbf{un}) = \mathsf{a}_{VL}(\mathbf{u}, \mathbf{n})\left[0,\ \frac{\rho}{\gamma}(2c - v_n)\mathbf{n},\ \frac{p + \rho v_n(2c - v_n)}{\gamma + 1}\right]^T.$$

2.5 AUSM+ flux splitting

The AUSM+ flux has been originally presented in Reference [18], where a derivation from the Van Leer flux difference splitting is also discussed. It can be re-formulated in accord with the numerical convective splitting by setting $\mathsf{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \mathsf{a}_{ausm+}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ and $\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \boldsymbol{G}_{ausm+}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) - \boldsymbol{G}_{ausm+}(\mathbf{u}_R, \mathbf{u}_L, -\mathbf{n})$ where

$$\mathsf{a}_{ausm+}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = \left(\mathcal{M}^+(v_{nL}/c) + \mathcal{M}^-(v_{nR}/c)\right)^+,$$

$$\boldsymbol{G}_{ausm+}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n}) = p(\mathbf{u}_L)\left[0,\ \mathcal{P}^+(v_{nL}/c),\ \mathsf{a}_{ausm+}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})\right]^T,$$

$$c^*_{L/R} = \left(\frac{2c^2_{L/R} + (\gamma - 1)v^2_{nL/R}}{\gamma + 1}\right)^{1/2},$$

$$c = \min\left\{c^*_L,\ c^*_R,\ \frac{c^{*2}_L}{v_{nL}},\ \frac{c^{*2}_R}{v_{nR}}\right\},$$

and

$$\mathcal{M}^{\pm}(M) = \begin{cases} (M \pm |M|)/2 & \text{if } |M| > 1; \\ (M \pm 1)^2/4 \pm \beta(M^2 - 1)^2 & \text{if } |M| \le 1. \end{cases}$$

$$\mathcal{P}^{\pm}(M) = \begin{cases} (M \pm |M|)/(2M) & \text{if } |M| > 1; \\ (M \pm 1)^2(2 \pm M)/4 \pm \alpha M(M^2 - 1)^2 & \text{if } |M| \le 1. \end{cases}$$

A number of AUSM-like fluxes and variants have been successively proposed and analyzed theoretically and their performance investigated experimentally, see for instance References [18,19]. These numerical fluxes can all be re-formulated in accord with the numerical convective splitting.

3 The semi-discrete Finite Volume formulation

In this section, we introduce the semi-discrete FV method and discuss several properties that will be useful in the analysis of the DIMEX-RK FV method presented in the next section.

3.1 Mesh notations and conventions

Our basic FV scheme is defined on a mesh that completely covers the computational domain $\Omega \in \mathbb{R}^d$. The mesh is defined as the union of a set of n non-overlapping *cells*, which actually are *intervals* in the 1-D case, of *triangles* in the 2-D one, and *tetrahedrons* in the 3-D one. The mesh is assumed to satisfy some constraints, i.e. it is to be *regular* and *conformal* in the sense specified by Reference [20].

Cells are conventionally labeled by an integer identifier ranging from 1 to n. The identifier is generically indicated by the index letters i, j or k. For the generic cell T_i we indicate by $|\mathsf{T}_i|$ the d dimensional measure of the cell (volume in 3-D, area in 2-D, length in 1-D).

The intersection of two cells or the intersection of a cell and the border of Ω with positive $(d-1)$ dimensional measure is called a face (edge in 2-D, point in 1-D). The internal face shared by the cells T_i and T_j is addressed by the pair ij and denoted by the symbol f_{ij}. For the sake of notation consistency, a boundary face is also addressed by a pair of indices, namely ik, i being the unique cell that the face belongs to, and k a specific boundary face identifier — like a fictitious "external" cell. This convention allows us to refer to either internal or boundary faces by means of an index pair. For the generic face f_{ij}, we indicate by $|\mathsf{f}_{ij}|$ its $(d-1)-$dimensional measure (area in 3-D, length in 2D, conventionally 1 in 1-D), and by $\mathbf{n}_{ij}$ its normal vector (± 1 in 1D). The normal vector is assumed to be oriented from cell i to cell j when the face is internal and outward directed when the face is on the boundary.

For each cell T_i, we denote the set of internal faces by $\sigma(i)$ and the subset of the cell faces located at the boundary by $\sigma'(i)$. The symbol $|\mathsf{T}_i|$ denotes the d-dimensional measure of the cell, i.e. the tetrahedron volume in 3-D, the triangle area in 2-D, and the interval length in 1-D. These notations and convections are suitable for practical implementation of Finite Volume solvers by using the object oriented library P2MESH [21].

3.2 Vector/Matrix notations and conventions

Let us now introduce the vector and matrix notations that will be utilized in this section and in the following ones. The symbol $\mathbf{I}_k$ indicates the $k \times k$ identity matrix, $\mathbf{1}_k$ the k size vector all of whose components are equal to 1, while the symbol $\mathbf{0}_k$ indicates the k size vector all of whose components are equal to 0. A positive vector $\mathbf{v}$ satisfies $\mathbf{v} \gg \mathbf{0}$, a non-negative vector $\mathbf{v}$ satisfies $\mathbf{v} \geq \mathbf{0}$, where the compact notation means

$$\mathbf{v} \gg \mathbf{0} \iff v_i > 0 \quad \text{for all } i$$

$$\mathbf{v} \geq \mathbf{0} \iff v_i \geq 0 \quad \text{for all } i$$

$$\mathbf{v} > \mathbf{0} \iff \mathbf{v} \geq 0 \quad \text{and } \mathbf{v} \neq \mathbf{0}.$$

Similar notations and definitions also apply to matrices; for instance, the matrix inequality $\mathbf{M} \leq \mathbf{N}$ means that $M_{ij} \leq N_{ij}$ for every pair ij, and a positive (non-negative) real matrix $\mathbf{M}$ is a matrix all of whose components are positive (non-negative) real numbers. Block vectors are denoted by underlined bold symbols; i.e. $\underline{\mathbf{u}}$ is a block vector. Their block components are indicated by indexed (non underlined) bold symbols; i.e. $\mathbf{u}_i$ is the i-th sub-vector block of $\underline{\mathbf{u}}$.

3.3 The basic semi-discrete Finite Volume scheme

The i-th cell-averaged solution state is denoted by $\mathbf{u}_i$, and the global collection of n cell-averaged data by $\underline{\mathbf{u}}$. Thus, this latter one is the $n \times m$-size block vector $\underline{\mathbf{u}}^T = (\mathbf{u}_1^T, \mathbf{u}_2^T, \ldots, \mathbf{u}_n^T)$, whose i-th block is the m-size vector $\mathbf{u}_i$. We can also address the cell-averaged k-th equation variable for $k = 1, 2, \ldots, m$ within the i-th cell, for $i = 1, 2, \ldots, n$ by the notation $\mathbf{u}_i|_k = \underline{\mathbf{u}}|_{k+mi}$. This is the k-th component of the i-th vector block.

Reformulating equation (1) in an integral form for each cell of the mesh, applying the Gauss divergence theorem and introducing suitable numerical flux functions to discretize the physical flux yield the semi-discrete FV numerical scheme in the form

$$|\mathsf{T}_i| \frac{d\mathbf{u}_i}{dt} + \sum_{j \in \sigma(i)} \boldsymbol{H}_{ij}(\underline{\mathbf{u}}) + \sum_{j' \in \sigma'(i)} \boldsymbol{H}_{ij'}^{(bc)}(\underline{\mathbf{u}}) = \mathbf{0}, \tag{4}$$

for each $i = 1, \ldots, n$. The terms $\boldsymbol{H}_{ij}(\underline{\mathbf{u}})$ and $\boldsymbol{H}_{ij'}^{(bc)}(\underline{\mathbf{u}})$ denote the numerical flux integrals estimated on internal and boundary faces. In equation (4), $\boldsymbol{H}_{ij}(\underline{\mathbf{u}})$ is estimated by using the cell-average approximations $\mathbf{u}_i$ and $\mathbf{u}_j$ within the elements i and j sharing the face f_{ij} and within an appropriate set of neighbor cells close to them. Instead, the term $\boldsymbol{H}_{ij'}^{(bc)}(\underline{\mathbf{u}})$ may also depend in some suitable form on a set of *external* data $\mathbf{u}_{j'}^{(bc)}$. We point out that boundary conditions may differ at distinct boundary faces, also implying a different functional form for the numerical fluxes.

Introducing the numerical flux (3) into (4) yields the final form of the basic semi-

discrete FV scheme, thus

$$|\mathsf{T}_i| \frac{d\mathbf{u}_i}{dt} + \sum_{j\in\sigma(i)} (\mathsf{a}_{ij}(\underline{\mathbf{u}})\mathbf{u}_i - \mathsf{a}_{ji}(\underline{\mathbf{u}})\mathbf{u}_j) + \sum_{j\in\sigma(i)} \boldsymbol{H}_{ij}^{(nc)}(\underline{\mathbf{u}}) + \sum_{j'\in\sigma'(i)} \boldsymbol{H}_{ij'}^{(bc)}(\underline{\mathbf{u}}) = \mathbf{0}, \quad (5)$$

where the scalar functions a_{ij} and the vectors $\boldsymbol{H}_{ij}^{(nc)}$ are indeed the terms of $\mathsf{a}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ and $\boldsymbol{H}^{(nc)}(\mathbf{u}_L, \mathbf{u}_R, \mathbf{n})$ evaluated at the cell interface ij.

This indexed notation is introduced because the functional form of a_{ij} and $\boldsymbol{G}_{ij}$ strongly depends on the spatial accuracy order of the approximation. In the next section, we focus on this issue and explain the dependence on high-order cell-average polynomial reconstructions.

3.4 The semi-discrete scheme

The simplest case concerns with a FV scheme thus of first order of accuracy in space. This scheme is obtained by using a single-point approximation to the numerical flux integral in (5), and is given by

$$\begin{aligned} \mathsf{a}_{ij}(\underline{\mathbf{u}}) &= |\mathsf{f}_{ij}|\, \mathsf{a}(\mathbf{u}_i, \mathbf{u}_j, \mathbf{n}_{ij}), \\ \boldsymbol{H}_{ij}^{(nc)}(\underline{\mathbf{u}}) &= |\mathsf{f}_{ij}|\, \boldsymbol{H}^{(nc)}(\mathbf{u}_i, \mathbf{u}_j, \mathbf{n}_{ij}). \end{aligned} \quad (6)$$

A higher-order accurate discretization is formally achieved by using a higher-order numerical quadrature for the numerical flux integrals. Notice, however, that a higher-order quadrature formula would imply the explicit knowledge of the value of the solution within the cells i and j at the different quadrature points located on the cell-interface. Since the first order accurate piecewise-constant representation of the solution given by the cell-average approximations is actually not enough, a better solution representation must be devised, which makes use of suitable higher-order piecewise polynomials. To this purpose, a number of polynomial reconstruction procedures have been proposed in literature. Their formal properties, listed for example in References [22–24], ensure the control of the spurious oscillations that can appear in the numerical solution.

In principle, any *non-oscillatory* reconstruction procedure which is capable of increasing the accuracy of the numerical flux integral can be inserted into our FV scheme. Let us write the higher-order approximation of the numerical flux integral as

$$\boldsymbol{H}_{ij}(\underline{\mathbf{u}}) = |\mathsf{f}_{ij}| \sum_{k=1}^{N_q} \omega_k \boldsymbol{H}(\mathbf{u}_i(\cdot, \mathbf{x}_{ij}^k), \mathbf{u}_j(\cdot, \mathbf{x}_{ij}^k), \mathbf{n}_{ij}),$$

where $\mathbf{x}_{ij}^k$ is the k^{th} quadrature point on the face f_{ij}, ω_k the corresponding quadrature weight, and $\mathbf{u}_i(t, \mathbf{x})$ the solution reconstructed in the i-th cell, which satisfies the following assumption.

Assumption 3 *The reconstructed solution $\mathbf{u}_i(\cdot, \mathbf{x})$ within the i-th cell satisfies*

$$\min_{j\in\sigma(i)} \mathbf{u}_j \le \mathbf{u}_i(\cdot, \mathbf{x}) \le \max_{j\in\sigma(i)} \mathbf{u}_j, \qquad \mathbf{x} \in \mathsf{f}_{ij},$$

for $i = 1, \ldots, n$.

Assumption 3 ensures that the values taken at quadrature nodes on the face f_{ij} of the piecewise polynomial representation of the solution within the cell T_i are always between the cell averages of T_i and the adjacent cell T_j. This issue follows quite naturally from the limiting strategies usually adopted to guarantee the satisfaction of a maximum principle [25–28]

The further assumption that we need is the following one.

Assumption 4 *All of the weights used in quadrature formulae are* positive*, i.e.* $\omega_i > 0$ *for all* $i = 1, \ldots, N_q$*.*

This assumption is quite naturally satisfied by a number of quadrature formulae, such as the Gaussian ones.

Then, we introduce the numerical convective splitting for FV schemes of order higher than one by using the following definitions:

$$\mathsf{a}_{ij}(\underline{\mathbf{u}}) = |\mathsf{f}_{ij}| \sum_{k=1}^{N_q} \omega_k \mathsf{a}_{ij}^k(\underline{\mathbf{u}}), \tag{7a}$$

$$\boldsymbol{H}_{ij}^{(nc)}(\underline{\mathbf{u}}) = \boldsymbol{G}_{ij}^{(a)}(\underline{\mathbf{u}}) + |\mathsf{f}_{ij}| \sum_{k=1}^{N_q} \omega_k \boldsymbol{H}^{(nc)}(\mathbf{u}_i(\cdot, \mathbf{x}_{ij}^k), \mathbf{u}_j(\cdot, \mathbf{x}_{ij}^k), \mathbf{n}_{ij}), \tag{7b}$$

where

$$\mathsf{a}_{ij}^k(\underline{\mathbf{u}}) = \mathsf{a}(\mathbf{u}_i(\cdot, \mathbf{x}_{ij}^k), \mathbf{u}_j(\cdot, \mathbf{x}_{ij}^k), \mathbf{n}_{ij}), \tag{7c}$$

$$\boldsymbol{G}_{ij}^{(a)}(\underline{\mathbf{u}}) = |\mathsf{f}_{ij}| \sum_{k=1}^{N_q} \omega_k (\mathsf{a}_{ij}^k(\underline{\mathbf{u}})(\mathbf{u}_i(\cdot, \mathbf{x}_{ij}^k) - \mathbf{u}_i) - \mathsf{a}_{ji}^k(\underline{\mathbf{u}})(\mathbf{u}_j(\cdot, \mathbf{x}_{ij}^k) - \mathbf{u}_j)). \tag{7d}$$

In the definitions above, the a-term has been split in two contributions, namely a_{ij} and $\boldsymbol{G}_{ij}^{(a)}$. The former one is the scalar a-function estimated at the quadrature nodes by using the cell reconstructions and contributes to the linear advective term in (3). The latter one takes into account how much the reconstructed values at quadrature nodes differ from the cell-average solutions. This term is added to $\boldsymbol{H}^{(nc)}$, which may seem unusual at a first sigth. Nonetheless, the reconstruction step introduces a well-known anti-diffusive effect [29], which justifies that $\boldsymbol{G}_{ij}^{(a)}(\underline{\mathbf{u}})$ be considered like a "non-convective" contribution to the scheme. When a first order piecewise constant reconstruction is considered and the numerical flux integral is estimated by the mid-point rule, i.e. $N_q = 1$, we have that $\boldsymbol{G}_{ij}^{(a)} = 0$, and the definition of $\mathsf{a}_{ij}(\underline{\mathbf{u}})$ and $\boldsymbol{H}_{ij}^{(nc)}(\underline{\mathbf{u}})$ in (7) coincides with the one in (6).

3.5 Matrix notation for the semi-discrete scheme

In this section, we introduce a compact matrix notation to manipulate the scheme (5). Let us first introduce the $n \times n$ matrix $\mathbf{A}(\underline{\mathbf{u}})$ defined as

$$A_{ij}(\underline{\mathbf{u}}) = \begin{cases} \sum_{l \in \sigma(i)} \mathsf{a}_{il}(\underline{\mathbf{u}}) & \text{if } i = j; \\ -\mathsf{a}_{ji}(\underline{\mathbf{u}}) & \text{if } ij \text{ addresses a mesh face, namely } \mathsf{f}_{ij}; \\ 0 & \text{otherwise;} \end{cases} \tag{8}$$

the diagonal matrix $\mathbf{D} = \text{diag}(|\mathsf{T}_1|, \ldots, |\mathsf{T}_N|)$, whose i-th component is the d-measure of the cell T_i, and the block vector $\underline{\mathbf{b}}(\underline{\mathbf{u}})^T = [\,\mathbf{b}_1(\underline{\mathbf{u}})^T, \ldots, \mathbf{b}_n(\underline{\mathbf{u}})^T\,]$, whose i-th block is given by

$$\mathbf{b}_i(\underline{\mathbf{u}}) = -\sum_{j \in \sigma(i)} \boldsymbol{H}_{ij}^{(nc)}(\underline{\mathbf{u}}) - \sum_{j' \in \sigma'(i)} \boldsymbol{H}_{ij'}^{(bc)}(\underline{\mathbf{u}}).$$

Finally, by using the definitions introduced in (8), and the diagonal matrix $\mathbf{D}$, we reformulate the semi-discrete FV scheme (4) in the more compact matrix form

$$(\mathbf{D} \otimes \mathbf{I}_m) \frac{d\,\underline{\mathbf{u}}}{d\,t} = \underline{\mathbf{b}}(\underline{\mathbf{u}}) - [\mathbf{A}(\underline{\mathbf{u}}) \otimes \mathbf{I}_m]\,\underline{\mathbf{u}}. \tag{9}$$

The theoretical properties of the discretization matrices $\mathbf{A}(\underline{\mathbf{u}})$ and $\mathbf{A}(\underline{\mathbf{u}}) \otimes \mathbf{I}_m$ are investigated in second part of this work.

4 Implicit Explicit Runge Kutta discretization in time

In order to develop a numerical method that is stable when a source of stiffness is present in the equation, a fully implicit discretization should be adopted. This approach would be surely successful, but would also be likely to be quite expensive. For this reason, we focus our attention to the implicit-explicit discretization stategy.

For the sake of presentation, let us consider the ordinary differential equation

$$\begin{cases} y' = f(x, y) + g(x, y), \\ y(a) = y_a, \end{cases}$$

for the variable y, which is a function of the independent variable x; $f(x, y)$ and $g(x, y,)$, are two given right-hand side terms, and y_a the initial condition for the dependent variable y. Let also assume that $f(x, y)$ be the stiff part of the source term, and $g(x, y)$ be the non-stiff one. The idea is that an implicit Runge-Kutta scheme can be applied to discretize the stiff term and an explicit Runge-Kutta scheme can be used for the non-stiff one. Combining the two distinct Runge-Kutta schemes yields the ”composite” algorithm:

— solve the s–size non linear system for $y^{(1)}, y^{(2)}, \ldots, y^{(s)}$:

$$\begin{pmatrix} y^{(1)} - y_n - h\sum_{j=1}^{s} \alpha_{1j}^{\text{IM}} f(x_n + c_j^{\text{IM}} h, y^{(j)}) - h\sum_{j=1}^{i-1} \alpha_{1j}^{\text{EX}} g(x_n + c_j^{\text{EX}} h, y^{(j)}) \\ y^{(2)} - y_n - h\sum_{j=1}^{s} \alpha_{2j}^{\text{IM}} f(x_n + c_j^{\text{IM}} h, y^{(j)}) - h\sum_{j=1}^{i-1} \alpha_{2j}^{\text{EX}} g(x_n + c_j^{\text{EX}} h, y^{(j)}) \\ \vdots \\ y^{(s)} - y_n - h\sum_{j=1}^{s} \alpha_{sj}^{\text{IM}} f(x_n + c_j^{\text{IM}} h, y^{(j)}) - h\sum_{j=1}^{i-1} \alpha_{sj}^{\text{EX}} g(x_n + c_j^{\text{EX}} h, y^{(j)}) \end{pmatrix} = \mathbf{0};$$

— then compute

$$y_{n+1} = y_n + h\sum_{i=1}^{s} \omega_i^{\text{IM}} f(x_n + c_i^{\text{IM}} h, y^{(i)}) + h\sum_{i=1}^{s} \omega_i^{\text{EX}} g(x_n + c_i^{\text{EX}} h, y^{(i)}).$$

It is important to note that using this splitting instead of using the same fully implicit Runge-Kutta method for both terms $f(x, y)$ and $g(x, y)$ does not provide any real advantage. In fact, the non-linear algebraic system that must be solved in the former case is of the same size of the the system that results in the latter one. Thus, we conclude that this combination is still too general for the aims of computational efficiency and stability that we intend to attain. For this purpose, a *diagonally* implicit Runge-Kutta discretization of the stiff term turns out to be more suitable instead of the fully implicit one previously considered:

— for each $i = 1, \ldots, s$ solve for $y^{(i)}$:

$$y^{(i)} - h\alpha_{ii}^{\text{IM}} f(x_n + c_j^{\text{IM}} h, y^{(j)}) = r^{(i)},$$

where

$$r^{(i)} = y_n + h\sum_{j=1}^{i-1} \left[\alpha_{ij}^{\text{IM}} f(x_n + c_j^{\text{IM}} h, y^{(j)}) + \alpha_{ij}^{\text{EX}} g(x_n + c_j^{\text{EX}} h, y^{(j)})\right];$$

— then compute

$$y_{n+1} = y_n + h\sum_{i=1}^{s} \omega_i^{\text{IM}} f(x_n + c_i^{\text{IM}} h, y^{(i)}) + h\sum_{i=1}^{s} \omega_i^{\text{EX}} g(x_n + c_i^{\text{EX}} h, y^{(i)}).$$

Specific values of $\boldsymbol{\alpha}^{\text{EX}}$, $\boldsymbol{\alpha}^{\text{IM}}$, $\boldsymbol{\omega}^{\text{EX}}$, $\boldsymbol{\omega}^{\text{IM}}$, $\mathbf{c}^{\text{EX}}$, $\mathbf{c}^{\text{IM}}$ can be found in Reference [2,3].

Using the splitting in this form instead of using a single diagonally implicit Runge-Kutta scheme allows us to simplify the resolution procedure because the final non linear system only contains the contribution of f.

Let us rewrite equation (9) as

$$\frac{d\,\underline{\mathbf{u}}}{d\,t} = \underline{\mathbf{b}}\,(\underline{\mathbf{u}}) - \underline{\mathbf{a}}\,(\underline{\mathbf{u}})\,, \tag{10}$$

where

$$\begin{aligned} \mathcal{A}(\underline{\mathbf{u}}) &= \mathbf{D}^{-1}\mathbf{A}(\underline{\mathbf{u}}) \otimes \mathbf{I}_m, \\ \underline{\mathbf{a}}\,(\underline{\mathbf{u}}) &= \mathcal{A}(\underline{\mathbf{u}})\underline{\mathbf{u}}, \\ \underline{\mathbf{b}}\,(\underline{\mathbf{u}}) &= \left(\mathbf{D}^{-1} \otimes \mathbf{I}_m\right) \underline{\mathbf{b}}(\underline{\mathbf{u}}). \end{aligned} \tag{11}$$

The DIMEX-RK method provides a natural framework to discretize (10). In fact, since $\underline{\mathbf{a}}(\underline{\mathbf{u}})$ contains the transport contribution to the flux, it can be interpreted as the *stiff* part of the system of ODEs (10), and an implicit discretization can be seek. On the other hand, the second r.h.s. term, i.e. $\underline{\mathbf{b}}(\underline{\mathbf{u}})$, can be discretized explicitly.

The general s-stage DIMEX-RK scheme can be formulated as:

— for each $i = 1, \ldots, s$ solve for $\underline{\mathbf{w}}^i$:

$$\underline{\mathbf{w}}^i + \Delta t\, \alpha_{ii}^{\text{IM}}\, \underline{\mathbf{a}}\left(\underline{\mathbf{w}}^i\right) = \underline{\mathbf{u}}^n + \Delta t \sum_{j=1}^{i-1} \left(\alpha_{ij}^{\text{EX}} \underline{\mathbf{b}}\left(\underline{\mathbf{w}}^j\right) - \alpha_{ij}^{\text{IM}} \underline{\mathbf{a}}\left(\underline{\mathbf{w}}^j\right)\right), \quad \text{(12a)}$$

— then compute

$$\underline{\mathbf{u}}^{n+1} = \underline{\mathbf{u}}^n + \Delta t \sum_{i=1}^{s} \left(\omega_i^{\text{EX}} \underline{\mathbf{b}}\left(\underline{\mathbf{w}}^i\right) - \omega_i^{\text{IM}} \underline{\mathbf{a}}\left(\underline{\mathbf{w}}^i\right)\right). \quad \text{(12b)}$$

The DIMEX-RK method in (12a)–(12b) requires the solution of s non-linear systems of the form

$$\underline{\mathbf{w}} + \Delta t\, a\, \underline{\mathbf{a}}(\underline{\mathbf{w}}) = \mathbf{r}, \qquad a > 0, \quad \text{(13)}$$

where $\mathbf{r}$ is the right-hand-side of (12a). In view of equation (11), let us first define the map

$$\boldsymbol{\Phi}(\underline{\mathbf{w}}) = \left(\mathbf{I} + \Delta t\, a\, \mathcal{A}(\underline{\mathbf{w}})\right)^{-1} \mathbf{r}. \quad \text{(14)}$$

Let us then observe that every solution $\underline{\mathbf{w}}$ of (13) is equivalently a fixed point of (14). The main results of this section are resumed in the two following theorems. The first one formally states the existence and uniqueness of the solution to the non-linear equation (13). The other one defines a recursive procedure for solving (13). The proof of both theorems is based on some theoretical properties of the discretization matrices, such as the Lipschitz continuity of $\mathcal{A}(\underline{\mathbf{w}})$. The investigation of these properties will be the topic of the second part of this work.

Theorem 5 *The map* (14) *admits a fixed point for all* $\Delta t > 0$*. The fixed point is unique when*

$$\Delta t < \frac{1}{L\, a\, \kappa^2\, \|\mathbf{r}\|_1},$$

where

$$\kappa = \frac{\max_{i=1,2\ldots,n} |\mathsf{T}_i|}{\min_{i=1,2\ldots,n} |\mathsf{T}_i|},$$

and L *is the Lipschitz constant of the map* $\mathcal{A}(\mathbf{W})$.

Equation (13) depends on $\underline{\mathbf{w}}$ in a non-linear fashion and, thus, the IMEX-RK method as it has been proposed so far may be very expensive from the viewpoint of the computational costs. However, as it is stated in the following theorem, the solution $\underline{\mathbf{w}}$ can be approximated up to order $\mathcal{O}\left(\Delta t^{k+1}\right)$ by a straightforward iterative procedure.

Theorem 6 *Let $\underline{\mathbf{w}}^i$ be defined iteratively for $i = 0, \ldots, k$ as*

$$\begin{aligned} \underline{\mathbf{w}}^0 &= \underline{\mathbf{u}}^n, \\ \underline{\mathbf{w}}^i &= \left(\mathbf{I} + \Delta t\, a\, \mathcal{A}(\underline{\mathbf{w}}^{i-1})\right)^{-1} \mathbf{r}, \qquad i = 1, 2, \ldots, k. \end{aligned} \tag{15}$$

Then, the k-th iterate $\underline{\mathbf{w}}^k$ is an $\mathcal{O}\left(\Delta t^{k+1}\right)$ approximation of $\underline{\mathbf{w}}$, which is the exact solution to (13)*.*

In view of Theorem 6, solving non-linear systems of the form (13) is equivalent to solving a set of linear systems with the same coefficient matrix. This coefficient matrix usually has a significantly smaller size and is an M-matrix. Indeed, after some straightforward algebraic manipulations it turns out that the recursive procedure in (15) formally requires the solution of the k linear algebraic problems

$$\left(\mathcal{M}(\underline{\mathbf{w}}^{i-1}) \otimes \mathbf{I}_m\right) \underline{\mathbf{w}}^i = (\mathbf{D} \otimes \mathbf{I}_m)\mathbf{r} \qquad i = 1, 2, \ldots, k, \tag{16}$$

where

$$\mathcal{M}(\underline{\mathbf{w}}^{i-1}) = \mathbf{D} + \Delta t\, a\, \mathbf{A}(\underline{\mathbf{w}}^{i-1}). \tag{17}$$

This remarkable fact makes it possible a noteworthy simplification of the whole solution procedure. As for any $\mathbf{W}$ there holds that

$$\left(\mathcal{M}(\mathbf{W}) \otimes \mathbf{I}\right)^{-1} = \mathcal{M}(\mathbf{W})^{-1} \otimes \mathbf{I},$$

it is clear that at any step the scheme (15) requires the formal inversion of the matrix (17), which is the same for all of the m linear systems in (16). The size of this matrix is $n \times n$ and is thus smaller than the one of the whole coefficient matrix $\mathcal{M}(\mathbf{W}) \otimes \mathbf{I}$, which is $m\,n \times m\,n$.

As anticipated above, the matrix $\mathcal{M}(\mathbf{W})$ is an M-matrix; the proof of this property is given in the second part of this work. This fact has also some very important consequences as far as computational efficiency is concerned in solving the linear algebraic problems (16).

First, we point out that an M-matrix $\mathbf{A}$ such that $\mathbf{y}^T\mathbf{A} \geq \mathbf{0}$ for some vector $\mathbf{y} \gg \mathbf{0}$ admits an LU factorization whose triangular factors $\mathbf{L}$ and $\mathbf{U}$ are also M-matrices, see Reference [30]. Thus, no numerical pivoting is necessary to ensure stability in the LU factorization process, as would be the case of a general matrix, see Reference [31].

Then, the structural pattern of the matrix $\mathcal{M}(\mathbf{W})$ i.e. its non-zeros, only depends on the topological neighborhood relationships among the cells of the mesh. Thus, a non-zero matrix entry always corresponds to a connection between two adjacent cells of the mesh. If the mesh does not change in the time stepping calculation, i.e. no grid adaptation is carried out during the run, the non-zero pattern of $\mathcal{M}(\mathbf{W})$ must remain constant. Then, it follows that the symbolic pivoting of the matrix, i.e. the re-ordering of rows and columns to reduce and control the fill-in phenomenon of the factorization, can be performed only once at the beginning of each calculation. This fact may clearly have a strong impact in reducing both computational costs and memory storage requirements when direct algebraic methods for sparse matrices are used to solve (16).

The previous arguments based on the M-matrix nature of $\mathcal{M}(\mathbf{W})$ still remain valid for incomplete factorizations, i.e. when a direct re-solution method is used to pre-condition an iterative method. To this purpose, two important results are true. First, the incomplete LU-decomposition of an M-matrix is stable at least as the complete one without any numerical pivoting – see Theorem 3.2 in Reference [32]. Second, if we consider an incomplete factorization of the form $\mathbf{LU} = \mathbf{A} + \mathbf{R}$ for the M-matrix $\mathbf{A}$, then the following iterative scheme can be considered

$$\mathbf{LU}\mathbf{x}^{n+1} = \mathbf{R}\mathbf{x}^{n} + \mathbf{b}, \quad \Rightarrow \quad \mathbf{x}^{n+1} = \mathbf{x}^{n} + \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{q}^{n},$$

where $\mathbf{q}^{n} = \mathbf{b} - \mathbf{A}\mathbf{x}^{n}$. This iterative scheme requires a matrix-vector multiplication and the solution of two sparse triangular systems at each iterative step, that implies $\mathcal{O}(n^2)$ floating-point arithmetic operations. This iterative scheme can be proved to be convergent to the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ for every choice of the initial iterate $\mathbf{x}^0$ – see Theorem 2.5 of Reference [32].

5 Final Remarks

This paper is a contribution to the evidence for and against semi-implicit approaches on unstructured meshes in discretizing multi-dimensional hyperbolic systems. The method that we propose has been developed to perform integration of IBVPs in conservative divergence form. It is based on a special splitting of the physical and numerical flux vector function into a convective and a non-convective part, namely the numerical convective splitting. In the framework of IMEX-RK schemes, the convective part is discretized in an implicit way, while the non-convective one in an explicit way. This coupled IMEX-RK FV integrator is strictly conservative, shock-capturing, formally n-th order accurate in space and time, and does not require the evaluation of any Jacobian matrix.

In this paper, we discuss some theoretical issues of the method by introducing a general formalism that describes how a cell-centered FV discretization can be coupled to the IMEX-RK time-stepping schemes. Thanks to this formalism, all the theoretical results are independent of the spatial dimension of the problem, of the numerical flux — provided that this latter one can be re-formulated in accord with the numerical convective splitting — and of the cell-average polynomial reconstruction used to achieve higher order accuracy in space.

Basically, the time evolution matrix operator shows a peculiar block structure common to a wide family of numerical fluxes. The underlying block matrices are M-matrices and the analysis which can be carried out within this context makes it possible to show that simple and efficient resolution algorithms even for high-order schemes can be built. Although this FV scheme has been developed for unstructured mesh calculations, all these theoretical results can be straightforwardly extended to schemes defined on structured cartesian or curvilinear meshes. The proofs of these theorems as well as the numerical testing of the methods regarding both the computational efficiency and the quality of the approximation will be the topic of the second part of this work.

References

[1] U. M. Ascher, S. J. Ruuth, B. T. R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823.

[2] U. M. Ascher, S. J. Ruuth, R. J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, Appl. Numer. Math. 25 (2-3) (1997) 151–167, special issue on time integration (Amsterdam, 1996).

[3] L. Pareschi, G. Russo, Implicit-Explicit Runge-Kutta methods and applications to hyperbolic systems with relaxation, J. Sci. Comp.To appear.

[4] P. R. Halmos, Finite-Dimensional Vector Spaces, Van Nostrand, Princeton, N.J., 1958.

[5] I. Müller, T. Ruggeri, Extended thermodynamics, Springer-Verlag, New York, 1993.

[6] T. Ruggeri, Galilean invariance and entropy principle for systems of balance laws. The structure of extended thermodynamics, Contin. Mech. Thermodyn. 1 (1) (1989) 3–20.

[7] E. Bertolazzi, G. Manzini, High-order IMEX-RK finite volume methods for multidimensional hyperbolic systems, in: ENUMATH 2001 conference, ENUMATH, 2001, pp. 1–10.

[8] S. F. Davis, Simplified second-order Godunov-type methods, SIAM J. Sci. Statist. Comput. 9 (3) (1988) 445–473.

[9] B. Einfeldt, On Godunov-type methods for gas dynamics, SIAM J. Numer. Anal. 25 (2) (1988) 294–318.

[10] B. Einfeldt, C.-D. Munz, P. L. Roe, B. Sjögreen, On Godunov-type methods near low densities, J. Comput. Phys. 92 (2) (1991) 273–295.

[11] C.-W. Shu, S. Osher, Efficient implementation of essentially nonoscillatory shock-capturing schemes. II, J. Comput. Phys. 83 (1) (1989) 32–78.

[12] A. Kurganov, G. Petrova, Central schemes and contanct discontinuities, M2AN. Mathematical Modelling and Numerical Analysis 34 (6) (2000) 1259–1275.

[13] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection-diffusion equations, J. Comput. Phys. 160 (1) (2000) 241–282.

[14] J. L. Steger, R. F. Warming, Flux-vector splitting of the inviscid gas dynamic equations with application to finite-difference methods, J. Comput. Phys. 40 (1981) 263–293.

[15] B. V. Leer, Towards the ultimate conservative difference scheme, J. Comput. Phys. 32 (1979) 101–136.

[16] B. V. Leer, Flux-vector splitting for the Euler equations, Lecture Notes in Physics 170 (1982) 507–512.

[17] B. van Leer, Lecture Notes in Physics, Vol. 170, Sringer Verlag, 1992.

[18] M.-S. Liou, A sequel to AUSM: AUSM$^+$, J. Comput. Phys. 129 (2) (1996) 364–382.

[19] Y. Wada, M.-S. Liou, An accurate and robust flux splitting scheme for shock and contact discontinuities, SIAM J. Sci. Comput. 18 (3) (1997) 633–657.

[20] P. G. Ciarlet, The finite element method for elliptic problems, North-Holland Publishing Company, Amsterdam, Holland, 1980.

[21] E. Bertolazzi, G. Manzini, Algorithm 817 P2MESH: generic object-oriented interface between 2-D unstructured meshes and FEM/FVM-based PDE solvers, ACM TOMS 28 (1) (2002) 101–132.

[22] F. Aràndiga, R. Donat, A. Harten, Multiresolution based on weighted averages of the hat function. II. Nonlinear reconstruction techniques, SIAM J. Sci. Comput. 20 (3) (1999) 1053–1093 (electronic).

[23] F. Aràndiga, R. Donat, A. Harten, Multiresolution based on weighted averages of the hat function. I. Linear reconstruction techniques, SIAM J. Numer. Anal. 36 (1) (1999) 160–203 (electronic).

[24] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high-order accurate essentially nonoscillatory schemes. III, J. Comput. Phys. 71 (2) (1987) 231–303.

[25] T. Barth, D. Jespersen, The design and application of upwind schemes on unstructured meshes, in: 27th Aerospace Sciences Meeting, AIAA 89-0366, Reno, Nevada, 1989, pp. 1–12.

[26] M. Wierse, A new theoretically motivated higher order upwind scheme on unstructured grids of simplices, Adv. Comput. Math. 7 (1997) 303–335.

[27] M. E. Hubbard, Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids, J. Comput. Phys. 155 (1) (1999) 54–74.

[28] E. Bertolazzi, G. Manzini, A second-order maximum principle preserving finite volume method for steady convection-diffusion problems, SIAM J. Numer. Anal.Submitted.

[29] E. Bertolazzi, G. Manzini, Polynomial reconstructions and limiting strategies in finite volume approximations, in: Finite Volume for Complex Application, 2002 conference, FVCA-3, 2002, pp. 1–8.

[30] R. E. Funderlic, R. J. Plemmons, LU decomposition of M-matrices by elimination without pivoting, Linear Algebra and Appl. 41 (1981) 99–110.

[31] M. Neumann, On the Schur complement and the LU-factorization of a matrix, Linear and Multilinear Algebra 9 (4) (1980/81) 241–254.

[32] J. A. Meijerink, H. A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comp. 31 (137) (1977) 148–162.