Strong Stability Preserving and $A(\alpha)$-Stable Two-Derivative Time Discretization for Discontinuous Galerkin Method

A. Thenery Manikantan, J. Zeifang and J. Schütz

UHasselt Computational Mathematics Preprint Nr. UP-23-02

February 14th, 2023

Strong Stability Preserving and $A(\alpha)$-Stable Two-Derivative Time Discretization for Discontinuous Galerkin Method

Arjun Thenery Manikantan[a,*], Jonas Zeifang[a], Jochen Schütz[a]

[a] *Faculty of Sciences & Data Science Institute, Hasselt University, Agoralaan Gebouw D, BE-3590, Diepenbeek, Belgium.*

Abstract

In this paper, we analyze stability properties of the two-derivative strong stability preserving schemes in [Gottlieb et al., High order strong stability preserving multiderivative implicit and IMEX Runge-Kutta methods with asymptotic preserving properties, SIAM Journal on Numerical Analysis 60, 2022]. Stability analysis shows that the diagonally implicit two-derivative two-stage third order strong stability preserving scheme can never be A-stable. We implement the two-derivative strong stability preserving schemes for partial differential equations with a discontinuous Galerkin spectral element spatial discretization. We use Newton's method for non-linear stage equations and the generalized minimal residual method with a matrix-free approach for solving linear algebraic equations under suitable preconditioning. The method is applied for compressible Euler and Navier-Stokes equations with orders of consistency up to four. Numerical results show that the second and fourth order strong stability preserving schemes attain their desired order of convergence for relatively large timesteps; however, the third order scheme suffers from severe convergence issues from the (non-)linear solver, which is most likely due to worse stability properties.

Keywords: Strong stability preserving, Implicit time stepping, Multiderivative schemes, Discontinuous Galerkin spectral element method

*Corresponding author

Email addresses: arjun.thenerymanikantan@uhasselt.be (Arjun Thenery Manikantan), jonas.zeifang@uhasselt.be (Jonas Zeifang), jochen.schuetz@uhasselt.be (Jochen Schütz)

2000 MSC: 65L05, 65M20, 65M22, 65M60

1. Introduction

Consider the PDE defined on the spatial domain $\Omega \subset \mathbb{R}^n$,

$$\mathbf{w}_t + \nabla \cdot \mathbf{F}(\mathbf{w}) = 0, \tag{1}$$

where $\mathbf{w}(x,t) : \Omega \times \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ is the state vector and $\mathbf{F}(\mathbf{w}) : \mathbb{R}^m \to \mathbb{R}^m$ is the flux. For the Navier-Stokes equation, the flux function is given by

$$\mathbf{F}(\mathbf{w}) := \mathbf{F}^{inv}(\mathbf{w}) - \mathbf{F}^{\nu}(\mathbf{w}, \nabla\mathbf{w}), \tag{2}$$

with the inviscid flux $\mathbf{F}^{inv}$ and the viscous flux $\mathbf{F}^{\nu}$, where fluxes are given in Eqs. (33) and (37). The Euler equations are obtained by setting $\mathbf{F}^{\nu} \equiv 0$ in Eq. (2). Formally, the PDE (1) can be cast into an ODE

$$\mathbf{w}_t = \mathbf{R}^{(1)}(\mathbf{w}) \tag{3}$$

in some infinite dimensional function space where the function $\mathbf{R}^{(1)}(\mathbf{w})$ is defined as

$$\mathbf{R}^{(1)}(\mathbf{w}) := -\nabla \cdot \mathbf{F}(\mathbf{w}). \tag{4}$$

Widely used time stepping methods in the field of compressible computational fluid dynamics are explicit schemes. However, the conditional stability of the explicit integration schemes imposes timestep restrictions, which arise from the CFL condition. In order to overcome these timestep restrictions, we use *implicit time stepping schemes* which have less severe or no restrictions on the timestep. The multistep methods such as backward difference formulae (BDF) or the multistage methods such as diagonally implicit Runge-Kutta (DIRK) methods are standard examples for implicit time stepping methods. In [1], some efficient implicit schemes based on Rosenbrock type Runge-Kutta methods [2] and BDF methods can be found.

Classical implicit time stepping schemes in literature mostly use only the first order derivative ($\mathbf{w}_t$) for temporal integration [1]. The orders of consistency for one-derivative Runge-Kutta type methods can be increased only by increasing the number of stages, and hence by requiring more memory and more involved order conditions. To overcome this situation, one can include higher-order derivatives in the time stepping procedures, resulting

in *multiderivative methods*. The multistage multiderivative schemes were first considered half a century ago in [3, 4, 5]. In general, these methods belong to the larger class of multistep-multistage-multiderivative methods [5]. Including higher order derivatives in time stepping gives more flexibility to the method; hence it can achieve higher orders of consistency with the same number of stages as used in the one-derivative schemes [4, 6]. In this paper, we limit to only *two-derivative schemes*. Hence it is required to compute also the second order derivative for the temporal integration,

$$\mathbf{w}_{tt} = (\mathbf{R}^{(1)}(\mathbf{w}))_t = -\nabla \cdot \left(\frac{\partial \mathbf{F}}{\partial \mathbf{w}}(\mathbf{w}) \mathbf{R}^{(1)}(\mathbf{w}) \right) := \mathbf{R}^{(2)}(\mathbf{w}, \mathbf{R}^{(1)}(\mathbf{w})). \tag{5}$$

An asymptotic preserving higher order implicit two-derivative method for stiff ODEs was developed in a predictor-corrector fashion (HBPC) [7, 8]. In each correction step, the order of consistency is raised by one until some maximal order is reached; order reduction for low stiffness parameters was mitigated by adding more correction steps. The HBPC scheme was optimized in [9] to have $A(\alpha)$ stability almost up to $\alpha = 90^o$.

In [10], the authors have developed a higher order implicit two-derivative Runge-Kutta type scheme, which preserves the strong stability property. Higher order SSP schemes like the ones given in [10], are constructed by considering convex combinations of strong stability preserving forward Euler steps [11]. Assuming additional conditions on the second derivative such as *second derivative, backward derivative or Taylor series conditions*, the two-derivative SSP schemes were also developed, see the schemes [10, 12, 13, 14]. Besides multistage methods, there are higher order SSP two-derivative general linear methods (GLMs) in literature, see, e.g. [15].

To solve the PDE (1), the equations are discretized to form a large system of ODEs, which will be integrated via suitable timestepping methods. In [13, 14], explicit two-derivative SSP schemes are combined with WENO methods to solve conservation laws. Application of implicit multiderivative methods combined with DG spatial discretization for (non-)linear PDEs can be found in [16, 17, 18]. In [19], the HPBC schemes [7, 8] are combined with discontinuous Galerkin spectral element method (DGSEM) [20] and achieve order of accuracy up to eight. The scheme is parallelized in space and in [21] also in time.

In this work, we combine the two-derivative implicit SSP [10] time discretization approach with a spatial discretization of the DGSEM to solve Navier-Stokes equations. Linear stability properties of the SSP schemes [10]

are investigated with Dahlquist's equation. The implicit stages of the discretized system are solved using Newton's method and the generalized minimal residual method (GMRES) is used with appropriate preconditioning to solve the linear system arising in the Newton iterations. As the SSP schemes are structurally very similar to the stages of the HPBC schemes, we closely follow the preconditioning and the matrix-free approach implemented in [19]. Comparison of the SSP [10] schemes with the HBPC schemes [19] on the basis of wall-clock time and error convergence in Sec. 4.5 shows an advantage in performance of HBPC schemes over SSP schemes for Navier-Stokes equations.

The sections of this paper are structured as follows: In Sec. 2 the semi-descretization of the two-derivative SSP are explained with Butcher form, followed by the investigation of the stability properties. The fully discrete formulation is briefed in Sec. 3 with the weak formulation of the PDEs. Effect of preconditioning and the implementation of the preconditioner using a matrix-free approach is shortly recalled in Sec. 3.2. Validation of the SSP schemes on Euler and Navier-Stokes equations is done in Sec. 4 with the convergence results and the number of (non-)linear iterations plots. In the final section (Sec. 5), conclusions are made and an outlook is given.

2. Semi-discrete formulation

In this section, we discuss the semi-discrete formulation of the implicit two-derivative SSP Runge-Kutta method. We use the spatial operators $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$, see Eq. (3) and Eq. (5), for the semi-discretization. The aforementioned spatial operators are discretized using DGSEM, which will be described in Sec. 3. Given the approximate solution $\mathbf{w}^n$ at time t^n, the s-stage implicit two-derivative SSP Runge-Kutta method can be written in the following Butcher form,

$$\mathbf{w}^{(i)} := \mathbf{w}^n + \Delta t \sum_{j=1}^{i} a_{ij} \mathbf{R}^{(1)}(\mathbf{w}^{(j)}) + \Delta t^2 \sum_{j=1}^{i} \dot{a}_{ij} \mathbf{R}^{(2)}(\mathbf{w}^{(j)}, \mathbf{R}^{(1)}(\mathbf{w}^{(j)})) \quad (6)$$

for the stages $i = 1, \ldots, s$. The solution at time t^{n+1} is updated with

$$\mathbf{w}^{n+1} := \mathbf{w}^{(s)}. \quad (7)$$

The implicit two-derivative SSP scheme was derived for orders up to four; see Tab. 1. The Butcher coefficients $\mathbf{A} = \{a_{ij}\}$ and $\dot{\mathbf{A}} = \{\dot{a}_{ij}\}$ for the schemes

can be found in [10, Sec. 2.3]. In Sec. 2.1, we analyze the linear stability of the implicit SSP schemes (Tab. 1).

Scheme	**Order**	**Stages**	A(α)-**Stability**
SSP-I2DRK2-1	2	1	$A(90.00^o)$
SSP-I2DRK3-2	3	2	$A(79.94^o)$
SSP-I2DRK4-5	4	5	$A(84.51^o)$

Table 1: Strong stability preserving schemes from Gottlieb et al. [10, Sec. 2.3]. The A(α) stability analysis can be found in Sec. 2.1 of the current paper.

2.1. Linear stability of the implicit two-derivative SSP schemes

In this section, the linear stability of the implicit two-derivative SSP method has been analyzed using Dahlquist's equation $y' = \lambda y$, where $\lambda \in \mathbb{C}$. We first derive the stability function for an s-stage SSP scheme, and then the linear stability for each of the schemes (see Tab. 1) will be analyzed in the following subsections.

Plugging in $\mathbf{R}^{(1)} = \lambda y$ and $\mathbf{R}^{(2)} = \lambda y' = \lambda^2 y$ in Eq. (6) we get

$$y^{(i)} = y^n + \sum_{j=1}^{i} \left(a_{ij}\Delta t\lambda + \dot{a}_{ij}\Delta t^2\lambda^2\right) y^{(j)} \tag{8}$$

for each of the stages up to s. Define $z := \Delta t\lambda$ and functions $\mathcal{S}_i(z)$, so that the stage values $y^{(i)}$ in Eq. (8) can be explicitly written as

$$y^{(i)} = \mathcal{S}_i(z) y^n.$$

For $i \geq 2$, these functions can be recursively written as

$$\mathcal{S}_i(z) = \frac{1 + \sum_{j=1}^{i-1} \left(a_{ij}z + \dot{a}_{ij}z^2\right)\mathcal{S}_j(z)}{\left(1 - a_{ii}z - \dot{a}_{ii}z^2\right)} \text{ with } \mathcal{S}_1(z) = (1 - a_{11}z - \dot{a}_{11}z^2)^{-1}.$$

Then the update y^{n+1} can be written as

$$y^{n+1} = \mathcal{S}_s(z) y^n = \left(\frac{1 + \sum_{j=1}^{s-1} \left(a_{sj}z + \dot{a}_{sj}z^2\right)\mathcal{S}_j(z)}{\left(1 - a_{ss}z - \dot{a}_{ss}z^2\right)}\right) y^n =: \mathcal{S}(z) y^n, \tag{9}$$

where $\mathcal{S}(z)$ is the stability function.

2.1.1. Second order SSP scheme

The second order SSP scheme *SSP-I2DRK2-1* is none other than second order Taylor method. From Eq. (9), the stability function for *SSP-I2DRK2-1* is given by

$$\mathcal{S}(z) = \mathcal{S}_1(z) = \frac{2}{2 - 2z + z^2} \text{ for } z \in \mathbb{C}. \tag{10}$$

Then for the modulus value $|\mathcal{S}(z)|$

$$\max_{z \in \mathbb{C}^-} |\mathcal{S}(z)| \leq \max_{y \in \mathbb{R}} |\mathcal{S}(iy)| = \max_{y \in \mathbb{R}} \frac{2}{|(2 - y^2) - 2yi|} = \max_{y \in \mathbb{R}} \frac{2}{\sqrt{y^4 + 4}} \leq 1,$$

which implies that the second order SSP method is **A-stable**, see [22, Chap. 5]. The method is also **L-stable** because,

$$\lim_{z \to \infty} \mathcal{S}(z) = 0.$$

2.1.2. Third order SSP scheme

The stability function for *SSP-I2DRK3-2* is given by

$$\mathcal{S}(z) = \mathcal{S}_2(z) = \frac{18}{(6 + z^2)(3 - 3z + z^2)} \text{ for } z \in \mathbb{C}. \tag{11}$$

It can be found from Eq. (11) that $\mathcal{S}(z)$ has singularities on the boundary of the domain $\mathbb{C}^-$ at $z = \pm i\sqrt{6}$.

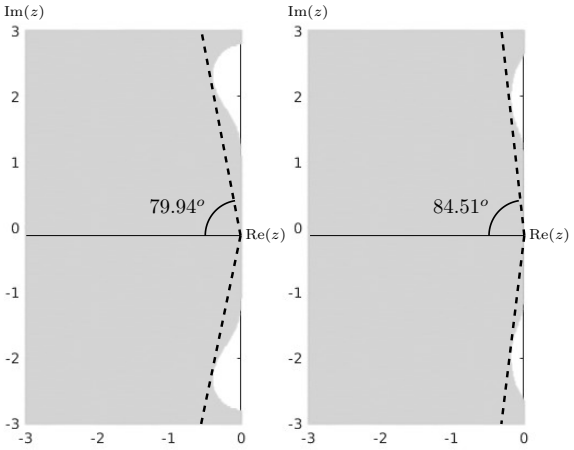
In Fig. 1, the stability region ($|\mathcal{S}(z)| < 1$) has been plotted for the values $-3 \leq \text{Re}(z) < 0$ and $-3 \leq \text{Im}(z) \leq 3$. The stability region is the gray shaded region in Fig. 1 (a). It can be seen that the stability region does not cover the entire negative complex half-plane, which implies that the method is **not A-stable**. Using the algorithm from [9, Sec. 3], it can be found that the method is approximately $A(79.94^o)$ stable.

2.1.3. Fourth order SSP scheme

The stability function for *SSP-I2DRK4-5* is given by

$$\mathcal{S}(z) = \mathcal{S}_5(z) = \frac{1 + \sum_{j=1}^{4} \left(a_{5j} z + \dot{a}_{5j} z^2\right) \mathcal{S}_j(z)}{\left(1 - a_{55} z - \dot{a}_{55} z^2\right)} \text{ for } z \in \mathbb{C} \tag{12}$$

with the Butcher coefficients given in [10, Sec. 2.3].



(a) SSP-I2DRK3-2 (b) SSP-I2DRK4-5

Figure 1: Stability region ($|\mathcal{S}(z)| < 1$) for third order (a) and fourth order (b) SSP schemes sketched using the algorithm from [9, Sec. 3] via the stability functions given in Eq. (11) and Eq. (12) respectively.

Similarly to the third order SSP scheme, the stability region (gray shaded region in Fig. 1 (b)) of the fourth order scheme also does not cover the entire negative complex half-plane, which implies that the method is **not A-stable**; the method is approximately $A(84.51^o)$ stable.

2.1.4. Third order A-stable scheme

It can be seen from the stability analysis, that the two-stage third order SSP scheme is $A(\alpha)$-stable only for a relatively low angle $\alpha = 79.94^o$, which can create timestep restrictions when applied to conservation equations. Analyzing the stability function (11), it can be inferred that the absence of an implicit term of the first derivative ($a_{11} = 0$) in the first stage leads to the formation of singularities for $|\mathcal{S}(z)|$ (11) on the imaginary axis. So a two-stage third order implicit two-derivative method with $a_{11} \neq 0$ might resolve this issue. Keeping implicitness on the first stage for the first derivative, we have the following third order **A-stable** scheme with coefficients,

$$\mathbf{A} = \begin{bmatrix} \frac{1}{3} & 0 \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} -\frac{1}{18} & 0 \\ -\frac{1}{12} & -\frac{1}{12} \end{bmatrix}. \tag{13}$$

However it does *not satisfy the SSP property* as per [10, Theorem 1]. In Sec. 4 it can be seen that the third order SSP scheme suffers from severe issues with

the linear solver. Hence, we will use the A-stable third order Runge-Kutta (*AS-I2DRK3-2*) scheme (13) for comparing the numerical results and also for understanding the severity of the number of (non-)linear iterations for the third order SSP scheme *SSP-I2DRK3-2*, with respect to *AS-I2DRK3-2* scheme (13).

2.2. A-stable implicit two-derivative two-stage third order SSP scheme

Consider the diagonally implicit two-derivative two-stage Runge-Kutta scheme with the coefficients

$$\mathbf{A} = \begin{bmatrix} \rho & 0 \\ \phi & \eta \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} \alpha & 0 \\ \gamma & \beta \end{bmatrix}. \tag{14}$$

Lemma 1. *There is no diagonally implicit two-derivative two-stage third order scheme* (14) *which is both* ***A-stable*** *and* ***SSP*** *according to the conditions in [10, Theorem 1].*

The Butcher coefficients (14) are to be modified as per [10, Theorem 1] to fulfill the SSP property. So, for $0 < k \leq 1$ we have

$$\phi = k\rho \text{ and } \gamma = k\alpha \tag{15}$$

where $\rho \geq 0$, $\eta \geq 0$, $\alpha \leq 0$ and $\beta \leq 0$. Now, rewrite the Butcher coefficients (14) with the aforementioned SSP conditions and the first order condition ($\phi + \eta = 1$) to obtain

$$\mathbf{A} = \begin{bmatrix} \rho & 0 \\ k\rho & 1 - k\rho \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} \alpha & 0 \\ k\alpha & \beta \end{bmatrix}. \tag{16}$$

Remark 1. *We assume that $k \neq 0$ because the third order conditions (see [10, Sec. 2.1]) give rise to an inconsistent system of equations for $k = 0$ (see Eqs.* (18) *and* (19)*).*

Now, applying the second order condition [10, Sec. 2.1, $p = 2$] we get

$$\beta = k(\rho - \rho^2 - \alpha) - \frac{1}{2}. \tag{17}$$

Consider Eq. (17) and apply the third order conditions [10, Sec. 2.1, $p = 3$] to obtain

$$k(\rho^3 - 2\rho^2 + 2\rho\alpha + \rho - 2\alpha) = \frac{1}{3}, \tag{18}$$

$$k(2\rho^3 - 2\rho^2 + 4\rho\alpha + \rho - 2\alpha) = \frac{1}{3}. \tag{19}$$

As $k \neq 0$ (of Remark 1), the equations (18) and (19) can be solved for ρ by eliminating k. Hence we have

$$\rho^3 + 2\alpha\rho = 0 \;\Rightarrow\; \rho = 0, \text{ or } \rho = \pm\sqrt{-2\alpha}.$$

We start with the non-zero solution. From the condition that $\alpha \leq 0$, take $\alpha = -2\mu^2$, for $\mu > 0$. Since the SSP property [10, Theorem 1] necessitates a non-negative ρ, the non-zero solution is

$$\rho = 2\mu, \;\; \mu > 0. \tag{20}$$

Substituting for α and ρ in (18) or (19) gives,

$$k = \frac{1}{6(\mu - 2\mu^2)}, \;\; \mu > 0. \tag{21}$$

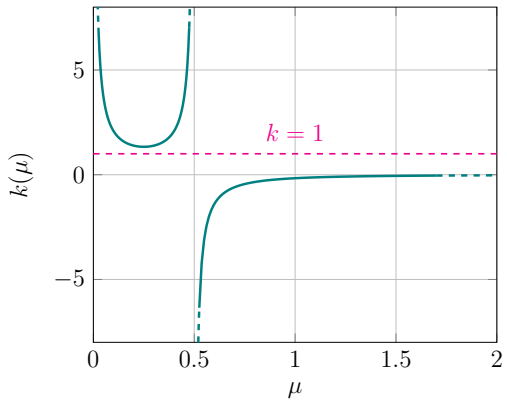


Figure 2: Graph of variable k (21) from the Butcher coefficients (16), obtained as a function of μ using the order equations (18) and (19). The function $k(\mu) = \frac{1}{6(\mu - 2\mu^2)}$ never reaches values from (0,1] for any values of $\mu > 0$.

The values of k have been plotted in Fig. 2 for $\mu > 0$. It can been seen clearly from Fig. 2 that

$$k = \begin{cases} > 1, \; 0 < \mu < 0.5 \\ < 0, \; \mu > 0.5 \end{cases} \tag{22}$$

So, there are no values for $\mu > 0$ such that $0 < k \leq 1$. Now consider the case when $\rho = 0$. Then from Eq. (18) and Eq. (17) we get

$$k\alpha = -\frac{1}{6}, \text{ and } \beta = -\frac{1}{3}. \tag{23}$$

Hence there are infinitely many possibilities for third order SSP schemes when $\rho = 0$ and it is given by

$$\mathbf{A} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} -\frac{1}{6k} & 0 \\ -\frac{1}{6} & -\frac{1}{3} \end{bmatrix}, \ 0 < k \leq 1. \tag{24}$$

The SSP scheme (24) is the same third order SSP scheme referred to in [10] for $k = 1$. The stability function for the scheme (24) is

$$\mathcal{S}(z) = \frac{18k + 3(1-k)z^2}{(6k + z^2)(3 - 3z + z^2)}, \tag{25}$$

and the scheme (24) can **never be A-stable** as $\mathcal{S}(z)$ has singularities at $\pm i\sqrt{6k}$. However, the schemes (24) are $A(\alpha)$ stable for $0 < k \leq 1$. The α angles are plotted against different k values in Fig. 3. It can be noted from Fig. 3 that the schemes (24) are tending towards $A(90^o)$ as $k \to 0$, but the Butcher coefficient $\dot{a}_{11} = -\frac{1}{6k} \to -\infty$ makes the scheme impractical for use.

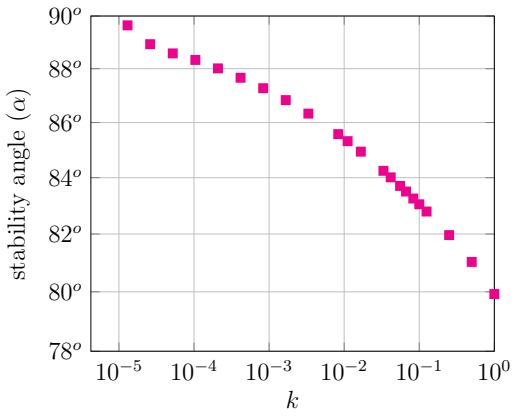


Figure 3: Stability angles α for the $A(\alpha)$-stable two-derivative two-stage third order scheme (24) is plotted for different values of the free variable $0 < k \leq 1$.

3. Fully discrete formulation

We use the discontinuous Galerkin spectral element method which has been introduced in [20] for the spatial discretization. The spatial domain Ω under consideration is subdivided into $\mathcal{N}_E$ quadrangular (2D) or hexahedral (3D) elements Ω_e. The discrete formulation of the DGSEM is recalled briefly in the following section by closely following the papers [19, 23].

3.1. Evaluation of the temporal derivatives with the DGSEM

The DGSEM utilizes the weak formulation of Eq. (1),

$$\sum_{e=1}^{\mathcal{N}_E}(\mathbf{w}_t,\phi)_{\Omega_e}-(\mathbf{F}(\mathbf{w}),\nabla\phi)_{\Omega_e}+\Big\langle\mathbf{F}^*(\mathbf{w}^L,\mathbf{w}^R)\cdot\mathbf{n},\phi\Big\rangle_{\partial\Omega_e}=0,\ \forall\phi\in\Pi_{\mathcal{N}_p}\quad(26)$$

where $\Pi_{\mathcal{N}_p}$ is the space of test functions constructed by the the tensor product of the one-dimensional Lagrange interpolation polynomials of degree $\mathcal{N}_p$. The scalar product $(\cdot,\cdot)_{\Omega_e}$ denotes the element-wise integration over the elements Ω_e and $\langle\cdot,\cdot\rangle_{\partial\Omega_e}$ is the integration along the cell-edges $\partial\Omega_e$. The flux function is replaced by a numerical flux function $F^*(\mathbf{w}^L,\mathbf{w}^R)$ on the cell-edges, which depends on the left and right states with respect to the cell-edge, and $\mathbf{n}$ is the outward pointing normal to the cell-edge. The global Lax-Friedrichs flux is used as the numerical flux (see [19, Eq. (13) and Eq. (17)]). The spatial DGSEM operator for the first time derivative $\mathbf{R}_h^{(1)}(\mathbf{w}_h)$ is given in [19, Eq. (12)].

As we use two-derivative time stepping schemes (6), it is required to compute the spatial DGSEM operator for the second derivative. The spatial operator for the second derivative is evaluated by introducing the artificial quantity

$$\boldsymbol{\sigma}:=\mathbf{R}^{(1)}(\mathbf{w})\equiv\mathbf{w}_t\quad(27)$$

as done in [16]. Differentiating Eq. (26) with respect to time and applying $\boldsymbol{\sigma}$, we get,

$$\begin{aligned}\sum_{e=1}^{\mathcal{N}_E}(\mathbf{w}_{tt},\phi)_{\Omega_e}-\Big(\frac{\partial\mathbf{F}(\mathbf{w})}{\partial\mathbf{w}}\boldsymbol{\sigma},\nabla\phi\Big)_{\Omega_e}+\Big\langle\frac{\partial\mathbf{F}^*(\mathbf{w}^L,\mathbf{w}^R)}{\partial\mathbf{w}^L}\boldsymbol{\sigma}^L\cdot\mathbf{n}\\+\frac{\partial\mathbf{F}^*(\mathbf{w}^L,\mathbf{w}^R)}{\partial\mathbf{w}^R}\boldsymbol{\sigma}^R\cdot\mathbf{n},\phi\Big\rangle_{\partial\Omega_e}=0,\ \forall\phi\in\Pi_{\mathcal{N}_p}.\end{aligned}\quad(28)$$

The spatial DGSEM operator for the second time derivative $\mathbf{R}_h^{(2)}(\mathbf{w}_h,\mathbf{R}^{(1)})$ is also evaluated similarly to that of the first time derivative and it is given in [19, Eq. (16)].

The SSP schemes [10] for the conservation laws (1) are implemented in the open source code FLEXI[1], which was developed for solving hyperbolic-parabolic conservation equations in a discontinuous Galerkin setting [24].

[1] `http://www.flexi-project.org`

3.2. Preconditioning of the extended linear system

The stage values in Eq. (6) can be solved using Newton's method. For the i^{th} stage, the equation can be cast into the following non linear form,

$$\mathbf{G}(\mathbf{w}^{(i)}) := \mathbf{g}(\mathbf{w}^{(i)}) - \mathbf{b} = 0, \tag{29}$$

where the function $\mathbf{g}$ and $\mathbf{b}$ are given by

$$\mathbf{g}(\mathbf{w}^{(i)}) = \mathbf{w}^{(i)} - \Delta t a_{ii}\mathbf{R}^{(1)}(\mathbf{w}^{(i)}) - \Delta t^2 \dot{a}_{ii}\mathbf{R}^{(2)}(\mathbf{w}^{(i)}, \mathbf{R}^{(1)}(\mathbf{w}^{(i)})) \text{ and}$$

$$\mathbf{b} = \mathbf{w}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathbf{R}^{(1)}(\mathbf{w}^{(j)}) + \Delta t^2 \sum_{j=1}^{i-1} \dot{a}_{ij}\mathbf{R}^{(2)}(\mathbf{w}^{(j)}, \mathbf{R}^{(1)}(\mathbf{w}^{(j)})).$$

More details over the non-linear formulation (29) can be found in [19, Sec. 3.2.1]. The linear system which arises in every Newton step is solved using the GMRES method. For faster convergence of the GMRES, application of a preconditioner is necessary.

In [19, Sec. 4], the authors have compared the effect of different preconditioners on linear and non-linear iterations (see [19, Fig. 2 and Fig. 4]) and found that a problem-tailored preconditioner $\mathbf{BJ}^H_{ext}$ and $\mathbf{BJ}_{ext}$ performs better than the others, in terms of the GMRES iterations per timestep and the wall-clock time. The major difference between $\mathbf{BJ}^H_{ext}$ and $\mathbf{BJ}_{ext}$ is that the first one uses the Hessian contribution, whereas the second does not (see [19, Sec. 3.2.3]). There was no notable difference in the GMRES iterations per timestep and the wall clock-time when compared between the preconditioners $\mathbf{BJ}^H_{ext}$ and $\mathbf{BJ}_{ext}$. Therefore, the authors have thereafter used a matrix-free approach with the $\mathbf{BJ}_{ext}$ preconditioner for the numerical investigations in [19]. See the papers [25] and [26] for the matrix-free implementation. In this paper, we also use the matrix free approach with $\mathbf{BJ}_{ext}$ preconditioning as implemented in [19, Sec. 4.2].

The effect of preconditioning on the GMRES iterations and Newton iterations can be seen in Fig. 4 for the linear-advection equation (30) on the spatial domain discretized with $\mathcal{N}_E = 16^2$ elements with $\mathcal{N}_p = 5$. A notable decrease in iterations of the linear solver and Newton's method can be seen in Fig. 4, when the linear equations are solved without and with preconditioning.

4. Numerical investigations

For the numerical testing, we consider the cases in Tab. 2 with the tolerances for Newton's method (ε_{Newton}) and the linear solver (ε_{GMRES}). For a

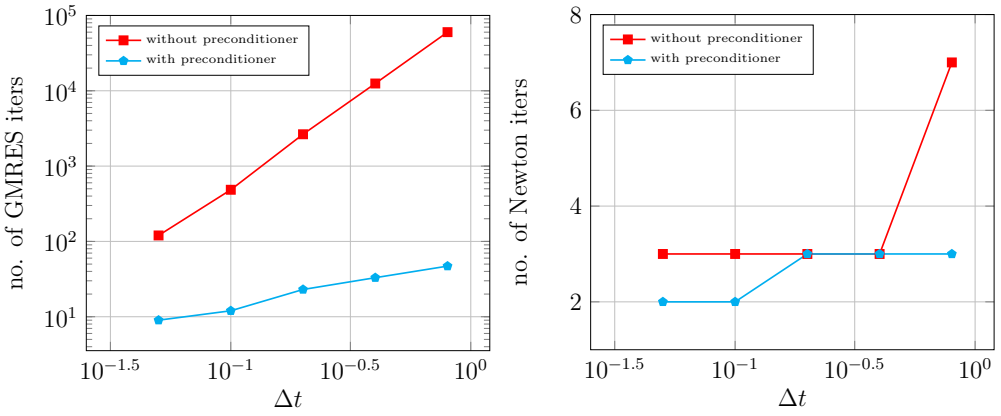


Figure 4: Linear scalar advection equation: the cumulated iterations required for the linear solver (left) and Newton's method (right) to complete one timestep for varying timestep sizes. The Eq. (30) with $a = (0.3, 0.3)^T$ is considered with the initial condition (31). The timestepping is *SSP-I2DRK2-1* and the setup is Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$ and $\varepsilon_{GMRES} = 10^{-3}$, within a limit of 70000 GMRES iterations per Newton step.

Case	**Domain** (Ω)	$\mathcal{N}_E$	$\mathcal{N}_p$	ε_{Newton}	ε_{GMRES}
1	$[-1, 1]^2$	16×16	5	10^{-7}	10^{-3}
2	$[-1, 1]^2$	32×32	7	10^{-12}	10^{-5}

Table 2: The different setups considered for the numerical testing of the SSP schemes with DGSEM spatial discretization with different mesh sizes ($\mathcal{N}_E$) and polynomial degree ($\mathcal{N}_p$).

given ε_{Newton}, the stopping criterion for the k^{th} Newton's iteration is given by

$$\left\|\mathbf{G}(\mathbf{X}^k)\right\|_2 < \varepsilon_{Newton} \cdot \left\|\mathbf{G}(\mathbf{X}^0)\right\|_2,$$

where $\left\|\mathbf{G}(\mathbf{X}^k)\right\|_2$ and $\|\mathbf{G}(\mathbf{X}^0)\|_2$ are the L_2 norms of the k^{th} and initial residuals respectively. For a given ε_{GMRES}, the stopping criterion for the GMRES iterations for the k^{th} Newton increment is

$$\left\|\mathbf{r}^k\right\|_2 < \varepsilon_{GMRES} \cdot \left\|\mathbf{G}(\mathbf{X}^{k-1})\right\|_2,$$

where $\mathbf{r}^k$ is the residual of the linear equation, see [19, Eq. 18]. Testcases are described in the following sections.

4.1. Linear advection equation

Consider the two dimensional linear advection equation

$$\mathbf{w}_t + \nabla \cdot (\mathbf{a}\mathbf{w}) = 0, \tag{30}$$

where $\mathbf{a} \in \mathbb{R}^2$ is a constant vector. Then the exact solution for the equation (30) with initial condition

$$\mathbf{w}(x, 0) = \sin\Big(\pi \sum_{j=1}^{2} x_j\Big) \tag{31}$$

is given by

$$\mathbf{w}(x, t) = \sin\Big(\pi \sum_{j=1}^{2} (x_j - a_j t)\Big).$$

4.2. Euler equations

We consider the two dimensional Euler equations of gas dynamics

$$\mathbf{w}_t + \nabla \cdot \mathbf{F}^{inv}(\mathbf{w}) = 0, \tag{32}$$

with the state vector $\mathbf{w} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ E \end{pmatrix}$. The flux function $\mathbf{F}^{inv}$ is given by

$$\mathbf{F}^{inv}(\mathbf{w}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + p \cdot \mathbf{Id} \\ \mathbf{v}(E + p) \end{pmatrix}, \tag{33}$$

where ρ is density, $\mathbf{v} = (v_1, v_2)$ is velocity, E is total energy and p is pressure. Pressure is evaluated using the equation of state of a perfect gas,

$$p = (\gamma - 1)\Big(E - \frac{\rho}{2}\|\mathbf{v}\|_2^2\Big)$$

with the isentropic coefficient $\gamma = 1.4$. For the numerical validation of the Euler equations (32), we consider an extension of the linear advection equation (30) with a constant velocity $\mathbf{v} \equiv \mathbf{a} = (0.3, 0.3)^T$ and pressure $p = 1$, and the initial condition

$$\rho(x, 0) = 1 + 0.3 \sin\Big(\sum_{j=1}^{2} x_j\Big), \tag{34}$$

which gives the exact solution,

$$\rho(x,t) = 1 + 0.3\sin\Big(\sum_{j=1}^{2}(x_j - a_j t)\Big),\ \mathbf{v} = \mathbf{a},\ p = 1. \tag{35}$$

The λ value for the global Lax-Friedrichs numerical flux (see [19, Eq. (13) and Eq. (17)]) is chosen to be $\lambda = \big(1,1,1,1\big)$, as per the values given in [27].

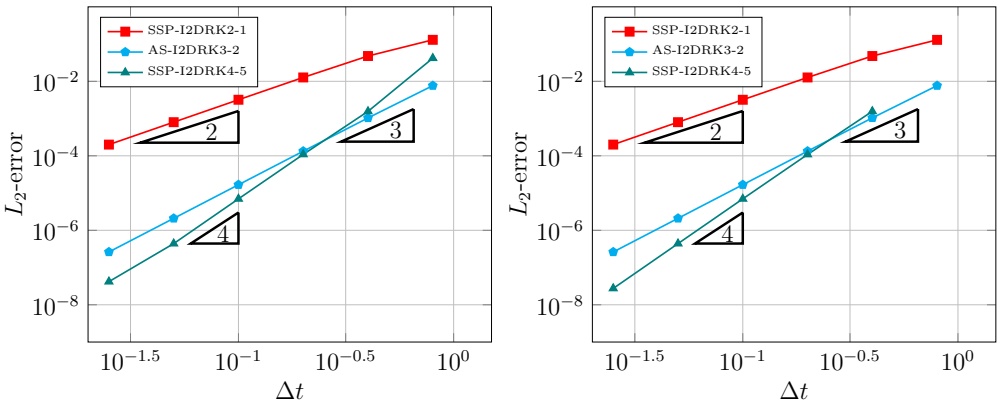


Figure 5: L_2-error for Euler equations with advection of density sine wave with $T_{end} = 0.8$ for the initial condition (34) using the second and fourth order SSP timestepping schemes, and the third order A-stable scheme. L_2-error is plotted for different timesteps Δt. The Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-8}$, $\varepsilon_{GMRES} = 10^{-3}$ (left) and the Tab. 2 case-2 with $\varepsilon_{Newton} = 10^{-12}$, $\varepsilon_{GMRES} = 10^{-5}$ (right) setups are chosen here with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration. The missing points are the ones that could not hit the given (non-)linear tolerances within the given GMRES iterations limit.

The exact solution (35) is used as the reference solution for analyzing the L_2-error. The convergence results for the second (*SSP-I2DRK2-1*) and fourth order (*SSP-I2DRK4-5*) schemes, and for a third order A-stable Runge-Kutta scheme (*AS-I2DRK3-2*) are visualized in Fig. 5. For the schemes *SSP-I2DRK2-1* and *AS-I2DRK3-2* the errors decreases as the time step decreases with the desired order of convergence. However, the fourth order SSP scheme exhibits its actual convergence order for $\Delta t \leq 0.4$ and the *AS-I2DRK3-2* scheme slightly outperforms *SSP-I2DRK4-5* for the larger timesteps.

As the implicit stages increase in the time stepping, the Newton and the GMRES iterations increase accordingly, see Fig. 6. However for the schemes *SSP-I2DRK2-1* and *SSP-I2DRK4-5*, the average iteration per stage per timestep remains more or less the same, see Fig. 6.

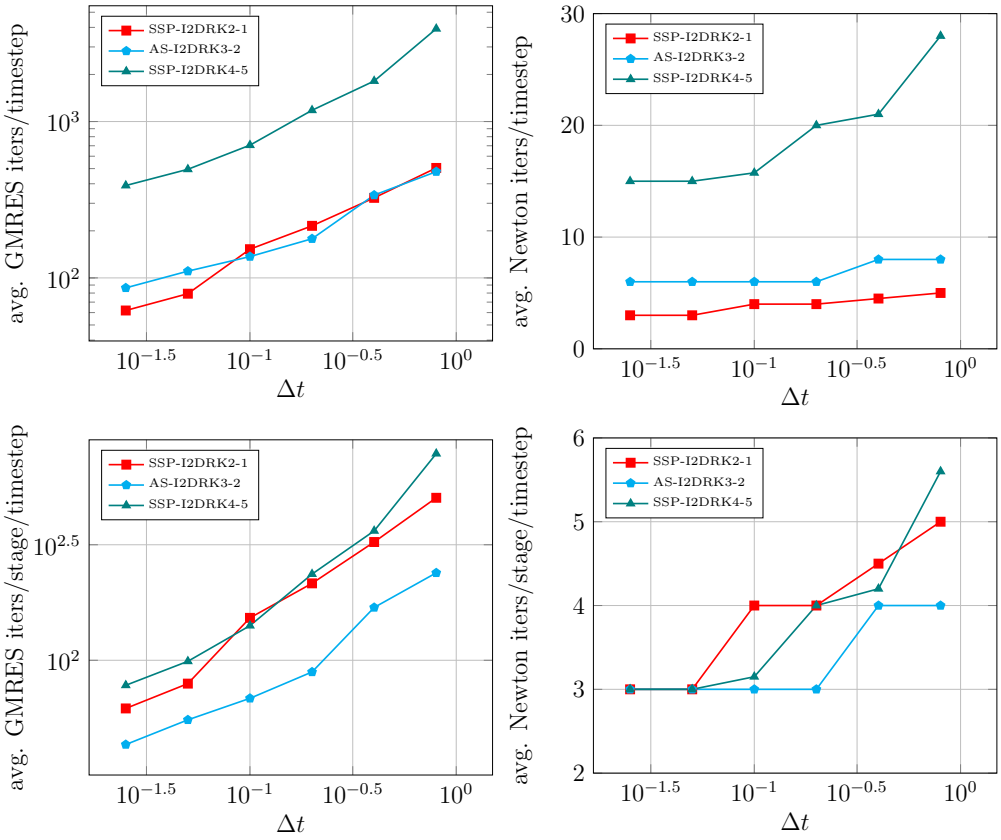


Figure 6: Euler equations with advection of density sine wave with the initial condition (34): the average iterations required for the linear solver (top left) and the Newton's method (top right) per timestep, and the average iterations required for the linear solver (bottom left) and the Newton's method (bottom right) per stage per timestep, for varying timestep sizes Δt for different timestepping schemes using Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-8}$, $\varepsilon_{GMRES} = 10^{-3}$ setup with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

4.3. Navier-Stokes equations

Next, we consider the two dimensional Navier-Stokes equations,

$$\mathbf{w}_t + \nabla \cdot \Big(\mathbf{F}^{inv}(\mathbf{w}) - \mathbf{F}^{\nu}(\mathbf{w}, \nabla \mathbf{w}) \Big) = 0, \tag{36}$$

with the state variables $\mathbf{w}$, the inviscid Euler flux $\mathbf{F}^{inv}$; and the viscous flux $\mathbf{F}^{\nu}$ given by

$$\mathbf{F}^{\nu}(\mathbf{w}, \nabla\mathbf{w}) = \begin{pmatrix} 0 \\ \tau \\ \tau.\mathbf{v} + \mathbf{q} \end{pmatrix} \tag{37}$$

where τ is the viscous tensor and $\mathbf{q}$ is the heat flux, given by

$$\tau := \mu\big(\nabla\mathbf{v} + (\nabla\mathbf{v})^T - \frac{2}{3}(\nabla \cdot \mathbf{v})\mathbf{Id}\big), \text{ and } \mathbf{q} := \lambda_T \nabla T \tag{38}$$

respectively. The corresponding other parameters and constants used in the above equations are, dynamic viscosity μ, temperature T given by the ideal gas equation, thermal conductivity $\lambda_T = \frac{c_p \mu}{Pr}$, $Pr = 0.72$ is the fluid specific Prandtl number, specific heat capacity $c_p = \frac{R\gamma}{\gamma - 1}$ and the specific gas constant $R = \frac{1}{\gamma}$.

As the viscous flux $\mathbf{F}^{\nu}$ in the Navier-Stokes equations (36) depends on the state vector $\mathbf{w}$ as well as its gradient $\nabla\mathbf{w}$, it results into a second order PDE system. So it is required to use an extended first order form for the equation (36) so as to utilize the fully discrete forms mentioned in the previous sections. Here, we use the BR2 lifting operator (see [28]) for the descretization of the second order equations. See [19, Sec. 5.1.1 - Sec. 5.1.3] for a detailed derivation.

For the numerical validation of the Navier-Stokes equation, we use the same set up as for the Euler equations. The viscosity is chosen to be $\mu = 10^{-3}$. In order to compute the L_2-error, a reference solution is computed via a fourth order explicit scheme [29] with a very small timestep $\Delta t = 10^{-6}$. The convergence results for the schemes are plotted in Fig. 7. The second order SSP scheme (*SSP-I2DRK2-1*) and the third order RK scheme (*AS-I2DRK3-2*) exhibit their desired order of convergence almost for every timesteps. However, the fourth order SSP scheme (*SSP-I2DRK4-5*) attains its actual order of convergence only for timesteps $\Delta t \leq 0.0125$. The *AS-I2DRK3-2* scheme outperforms the *SSP-I2DRK4-5* for a wide range of timesteps ($0.0125 < \Delta t \leq 0.8$). Even the second order schemes performs slightly better than the fourth order scheme for a couple of timesteps.

Comparison of the linear and non-linear iterations for the Navier-Stokes equations for the three schemes in Fig. 8 shows a similar behavioral pattern as that of the Euler equations.

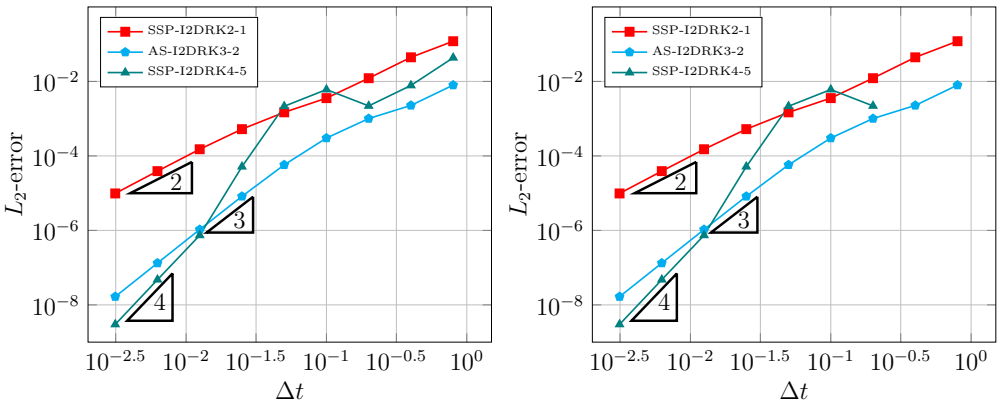


Figure 7: L_2-error for Navier-Stokes equations with advection and diffusion of density sine wave with $T_{end} = 0.8$ for the initial condition (34) using the second and fourth order SSP timestepping schemes, and the third order A-stable scheme. L_2-error is plotted for different timesteps Δt. The Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ (left) and the Tab. 2 case-2 with $\varepsilon_{Newton} = 10^{-12}$, $\varepsilon_{GMRES} = 10^{-5}$ (right) setups are chosen here with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration. The missing points are the ones that could not hit the given (non-) linear tolerances within the given GMRES iterations limit.

4.4. Third order SSP in detail

We use the two-derivative two-stage third order Runge-Kutta schemes *AS-I2DRK3-2*, γ-*RK3-2* and *RK3-2* (scheme with same stability angle as that of *SSP-I2DRK3-2*) to investigate the difficulty of using the third order SSP scheme. The construction of these schemes is shortly described in Appendix A and their (non-)linear stability properties are given in Tab. 3. The

Scheme	**A(α)-Stability**	**SSP**	**Reference**
AS-I2DRK3-2	90.00^o	No	Eq. (13)
0.5-RK3-2	89.80^o	No	Eq. (A.1) with $\gamma = 0.5$
0.1-RK3-2	84.05^o	No	Eq. (A.1) with $\gamma = 0.1$
0.004-RK3-2	80.12^o	No	Eq. (A.1) with $\gamma = 0.004$
0.00016-RK3-2	79.95^o	No	Eq. (A.1) with $\gamma = 0.00016$
0-RK3-2	79.94^o	Yes	[10, Sec. 2.3]
RK3-2	79.94^o	No	Eq. (A.2)

Table 3: A(α)-stable two-derivative two-stage third order Runge-Kutta schemes used for the investigation of the *SSP-I2DRK3-2* scheme.

Butcher tables for the schemes *RK3-2* and γ-*RK3-2* can be found in A.1 and

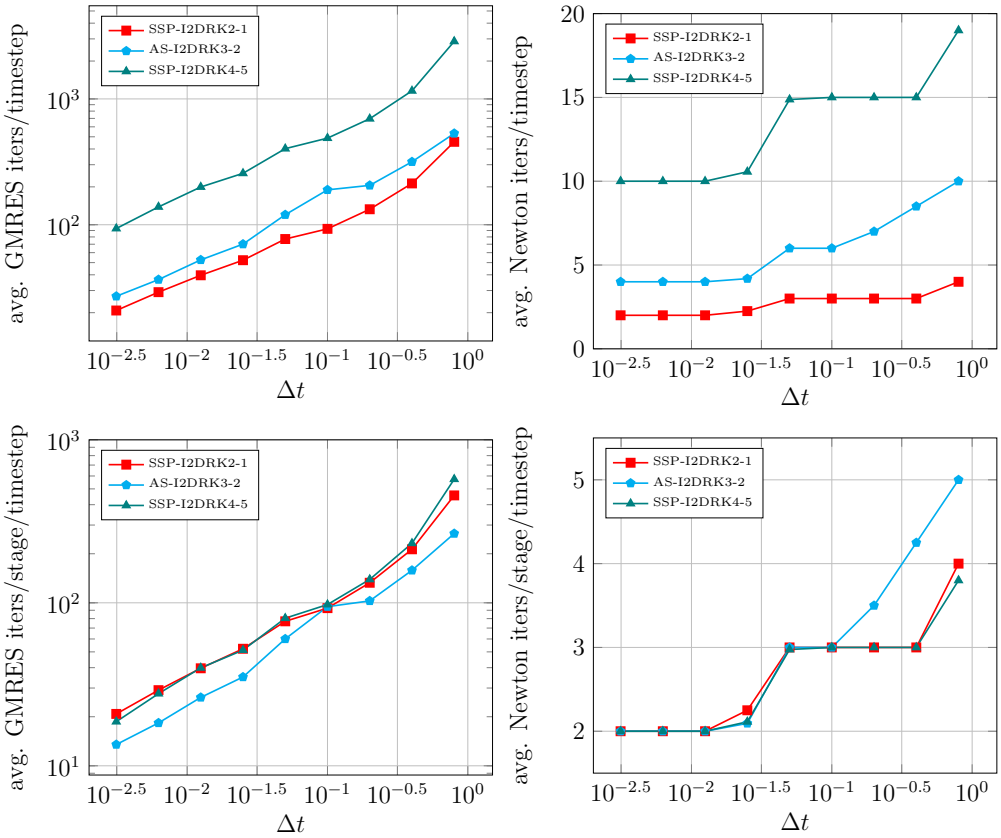


Figure 8: Navier-Stokes equations with advection and diffusion of density sine wave with the initial condition (34): the average iterations required for the linear solver (top left) and Newton's method (top right) per timestep, and the average iterations required for the linear solver (bottom left) and Newton's method (bottom right) per stage per timestep, for varying timestep sizes Δt for different timestepping schemes using Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

A.2. As $\gamma \to 0$ the scheme γ-*RK3-2* becomes *SSP-I2DRK3-2* ("0-*RK3-2*").

In Fig. 9, it can be seen that the third-order SSP scheme encounters convergence issues for some timesteps for which the other third order schemes (not SSP) perform better, when applied on simple linear advection equations. For Euler equations, the requirement of linear iterations for each Newton step for a single timestep is given in Fig. 10 for different third order RK schemes. It can be seen in Fig. 10 that the third order SSP requires a huge number of

GMRES iterations for a wide range of timesteps.

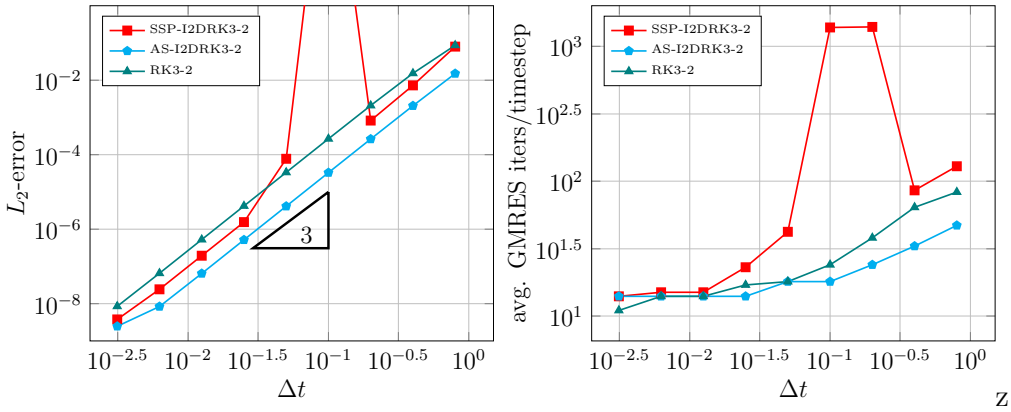


Figure 9: L_2-error (left) and average GMRES iterations per timestep (right) for linear advection of sine wave with $T_{end} = 0.8$ with the initial condition (31) simulated using the third order schemes for different timesteps Δt. The Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup is chosen here with in a limit of 10 Newton iterations per implicit solve and 2000 GMRES iterations per Newton iteration.

As γ decreases, the GMRES iterations for the γ-*RK3-2* schemes are increasing (see Fig. 10) towards the GMRES iterations of the *SSP-I2DRK3-2* scheme. The common properties of the *SSP-I2DRK3-2* and the γ-*RK3-2* with low γ values are;

- the implicit coefficient of the first derivative (a_{11}) for the first stage is either very small or zero
- the stability angle of the γ-*RK3-2* scheme is tending towards the stability angle of *SSP-I2DRK3-2* (79.94^o) as $\gamma \to 0$

So the behavior of having high GMRES iterations for *SSP-I2DRK3-2* can be either because of *the low stability angle or zero first derivative coefficient in the first stage*. However for the $A(79.94^o)$ stable *RK3-2* scheme with a relatively high first derivative coefficient in the first stage, the GMRES iterations are comparatively low (see Fig. 10). Analyzing Fig. 11, it can be found that the first stage requires a huge number of GMRES iterations compared to the second stage. So it must be the zero first derivative coefficient in the first stage that creates the issue of high GMRES requirement, which makes it impractical to use for a wide range of timesteps. Unfortunately there is no two-derivative two-stage third order SSP scheme with a non-zero first derivative coefficient in the first stage, as we have shown in Sec. 2.

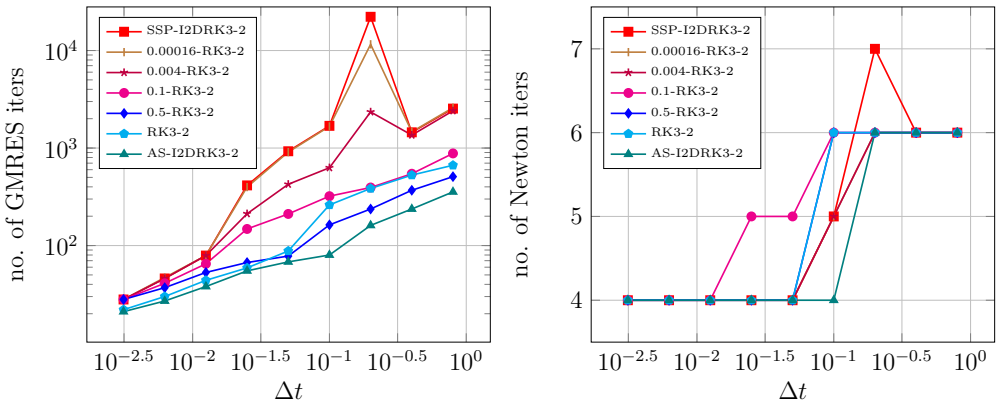


Figure 10: Euler equations with advection of density sine wave with initial condition (34): the cumulated iterations required for linear solver (left) and Newton's method (right) to complete one timestep for varying timestep sizes by the third order SSP and the other schemes in Tab. 3 with Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup within a limit of 5000 GMRES iterations per Newton step.

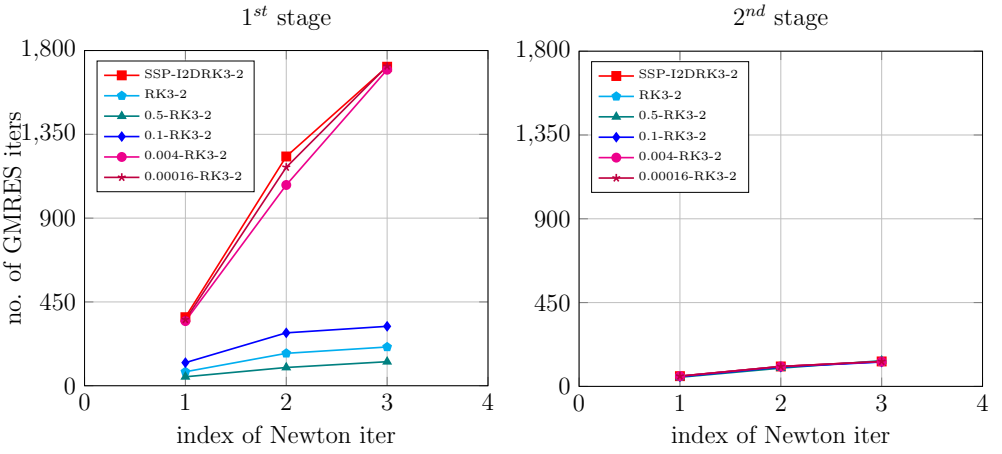


Figure 11: Euler equations with advection of sine wave with initial condition (34): the GMRES iterations (first and second stage separately) for a single timestep $\Delta t = 0.4$ for third order SSP and the other schemes in Tab. 3 with Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup within a limit of 5000 GMRES iterations per Newton step.

4.5. Comparison with HBPC scheme.

In this section, a comparison between the SSP scheme and HPBC scheme has been done on the basis of L_2-error, number of linear and non-linear iterations, and the wall clock time. The *HBPC(4,0)* is the HBPC scheme

with only the second order predictor step and *HBPC(4,2)* is the fourth order HBPC scheme with a predictor and two corrector steps, see [7, 8]. Same setup (case 1, Tab. 2) has been used for SSP and HBPC for the comparison. In particular for HBPC, predictor and corrector steps use the same GMRES and Newtons tolerances as given in Tab. 2.

The *SSP-I2DRK2-1* and the *HBPC(4,0)* gives the same results on Euler and Navier-Stokes equations (see Figs. 12 and 14) as they are exactly the same scheme. However the *HBPC(4,2)* gives better results on L_2-error for Euler and Navier-Stokes equations than *SSP-I2DRK4-5*. It can be seen from Figs. 13 and 15 that the *SSP-I2DRK4-5* scheme requires a lot more linear and non-linear iterations than the *HBPC(4,2)*. As the wall-clock time is directly proportional to the linear and non-linear iterations, a huge difference in relative wall-clock time can be visible between *HBPC(4,2)* and *SSP-I2DRK4-5*, and it is obvious to have such a difference as the *SSP-I2DRK4-5* has five implicit stages whereas the *HBPC(4,2)* has only three implicit stages. For the Navier-Stokes equations the *HBPC(4,2)* achieves its desired order of convergence for relatively large timesteps for which the *SSP-I2DRK4-5* could not (see Fig. 14).

5. Conclusion and outlook

In this work, we have analyzed the stability properties of the two-derivative SSP schemes [10], and have shown that the diagonally implicit two-derivative two-stage third order SSP can never be A-stable. SSP timestepping schemes [10] were implemented for Navier-Stokes equations in a DGSEM spatial framework. The complexity of the implementation of two-derivative Runge-Kutta scheme was outstripped by the introduction of the additional variable $\boldsymbol{\sigma}$ for the first order derivative as done in [16]. The stage values of non-linear equations were solved using Newton's method by employing GMRES method with a matrix-free approach on underlying preconditioned linear systems. The second and fourth order SSP schemes gave good convergence results on Euler and Navier-Stokes equations within a considerable number of Newton and GMRES iterations. The domination of temporal error over the spatial error for the chosen simulation setups was seen from the convergence plots. However, the third order SSP suffered from severe convergence issues due to the requirement of a huge number of non-linear and linear iterations and it was studied relative to the results on A-stable two-derivative two-stage third order Runge-Kutta scheme. Using a series of two-derivative third or-

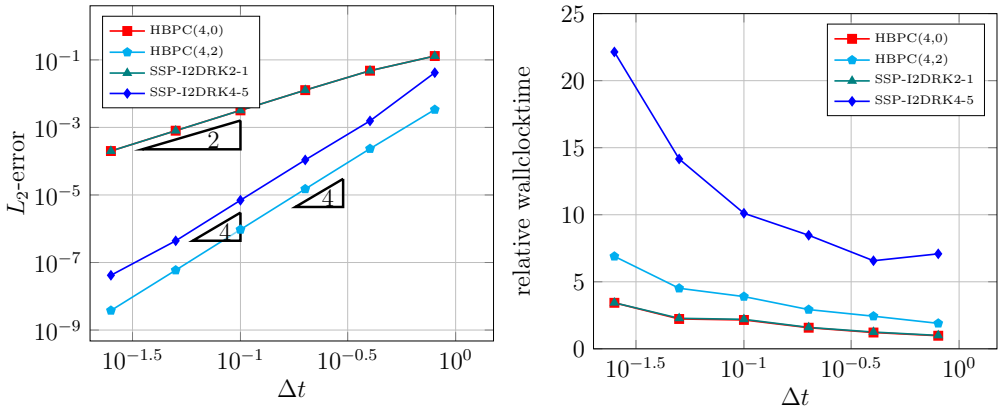


Figure 12: L_2-error (left) and the relative wall-clock time (right) for the SSP and HBPC schemes for the simulation of Euler equations with advection of density sine wave at $T_{end} = 0.8$ with the initial condition (34). The relative wall-clocktime is calculated with respect to simulation wall-clock time taken by *SSP-I2DRK2-1* for $\Delta t = 0.8$. L_2-error is plotted for different timesteps Δt. The Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-8}$, $\varepsilon_{GMRES} = 10^{-3}$ setup is chosen here with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

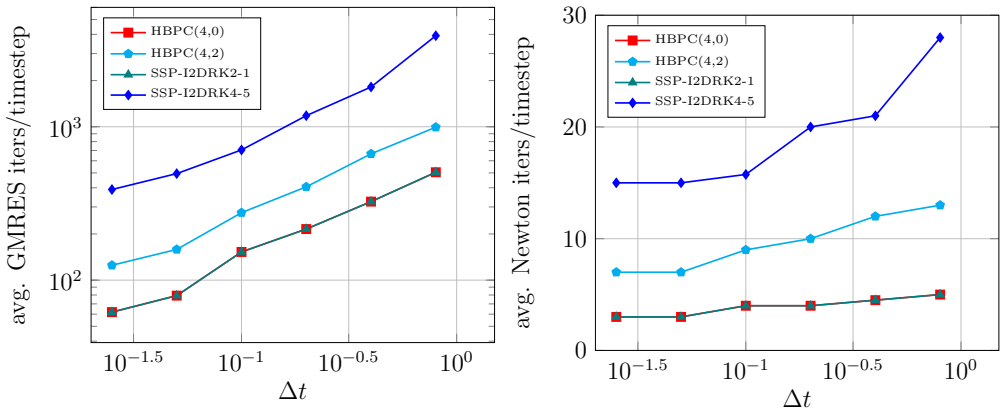


Figure 13: Euler equations with advection of density sine wave with the initial condition (34): the average iterations required for the linear solver (left) and Newton's method (right) per stage per timestep for varying timestep sizes Δt for the SSP and HBPC schemes using Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-8}$, $\varepsilon_{GMRES} = 10^{-3}$ setup with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

der schemes which is converging to the third order SSP scheme, it is shown that the zero first derivative coefficient in the first stage is responsible for the bad performance. Hence the third order SSP scheme was tested only on

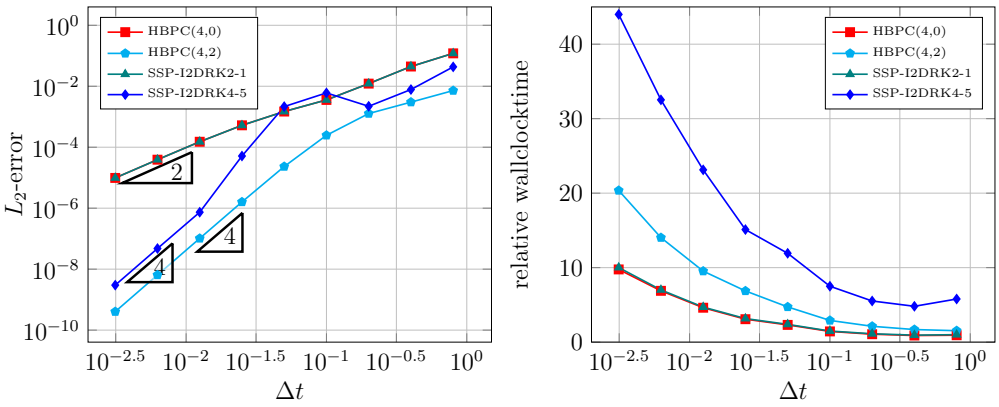


Figure 14: L_2-error (left) and the relative wall-clock time (right) for the SSP and HBPC schemes for the simulation of Navier-Stokes equations with advection and diffusion of density sine wave at $T_{end} = 0.8$ with the initial condition (34). The relative wall-clocktime is calculated with respect to simulation wall-clock time taken by *SSP-I2DRK2-1* for $\Delta t = 0.8$. L_2-error is plotted for different timesteps Δt. The Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup is chosen here with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

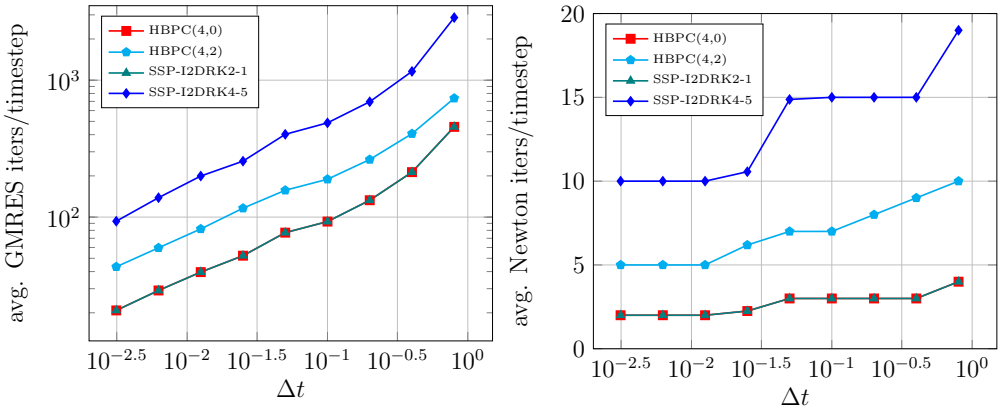


Figure 15: Navier-Stokes equations with advection and diffusion of density sine wave with the initial condition (34): the average iterations required for the linear solver (left) and Newton's method (right) per stage per timestep for varying timestep sizes Δt for the SSP and HBPC schemes using Tab. 2 case-1 with $\varepsilon_{Newton} = 10^{-7}$, $\varepsilon_{GMRES} = 10^{-3}$ setup with in a limit of 10 Newton iterations per implicit solve and 5000 GMRES iterations per Newton iteration.

linear advection equation. The fourth order HBPC scheme with predictor and corrector steps [19] performed better than SSP schemes when compared on the basis of convergence on relatively large timesteps and wall-clock time.

The Newton adaptive parallel-in-time HBPC [21] can perform even better, whereas the SSP schemes lack the property of being parallelized.

There are three possible directions for future investigations. Firstly, we are interested in implementing the asymptotic preserving IMEX timestepping schemes for low-Mach problems combined with DGSEM spatial discretization. Secondly, incorporation of the higher derivatives (order greater than two) using the Jacobian-free methods [6] and hence more flexibility can be achieved over the coefficients. Lastly, as there are higher order strong stability preserving GLMs in literature [15, 30, 31], the implementation of these schemes into the DGSEM framework is subject to numerical investigations.

Acknowledgments

Arjun Thenery Manikantan was funded by the "Bijzonder Onderzoeksfonds" (BOF) from UHasselt - project no. BOF21KP12. Jonas Zeifang was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project no. 457811052. We acknowledge the VSC (Flemish Supercomputer Center) for providing computing resources. The VSC is funded by the Research Foundation - Flanders (FWO) and the Flemish Government.

Appendix A. Construction of γ-*RK3-2* and *RK3-2* schemes

The γ-*RK3-2* schemes are constructed by keeping the implicit term of the first derivative (a_{11}) in the first stage as a free variable γ. Using the order conditions [10, Sec. 2.1, p=1, 2, 3], the remaining coefficients were written as a function of γ. Hence we have the Butcher coefficients for the two-derivative two-stage third order Runge-Kutta scheme γ-*RK3-2*

$$\mathbf{A} = \begin{bmatrix} \gamma & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} -\frac{1}{6} & 0 \\ -\frac{1}{6(1-\gamma)} & -\frac{1}{2} + \frac{1}{6(1-\gamma)} \end{bmatrix}, \tag{A.1}$$

with $\gamma \neq 1$. The γ-*RK3-2* are $A(\alpha)$-stable schemes, where the stability angles (see Tab. 3) are determined using the algorithm from [9, Sec. 3]. We use only γ-*RK3-2* schemes with $\gamma \leq 0.5$ for comparison results in Sec. 4.4.

The *RK3-2* scheme is constructed to have the same stability angle as that of an *SSP-I2DRK3-2* scheme. The scheme is obtained via slightly tuning the $\frac{1}{60}$-*RK3-2* scheme with a different coefficient $\dot{a}_{11} = -\frac{100}{6307}$. Hence the Butcher

coefficients for two-derivative two-stage third order $A(79.94^o)$-stable *RK3-2* scheme is given by

$$\mathbf{A} = \begin{bmatrix} \frac{1}{60} & 0 \\ 0 & 1 \end{bmatrix} \text{ and } \dot{\mathbf{A}} = \begin{bmatrix} -\frac{100}{6307} & 0 \\ -\frac{10}{59} & -\frac{39}{118} \end{bmatrix}. \tag{A.2}$$

References

[1] R. Hartmann, F. Bassi, I. Bosnyakov, L. Botti, A. Colombo, A. Crivellini, M. Franciolini, T. Leicht, E. Martin, F. Massa, et al., Implicit methods, in: TILDA: Towards Industrial LES/DNS in Aeronautics, Springer, 2021, pp. 11–59.

[2] F. Bassi, L. Botti, A. Colombo, A. Ghidoni, F. Massa, Linearly implicit Rosenbrock-type Runge-Kutta schemes applied to the discontinuous Galerkin solution of compressible and incompressible unsteady flows, Computers and Fluids 118 (2015) 305–320.

[3] P. Turán, On the theory of the mechanical quadrature, Acta Universitatis Szegediensis Acta Scientiarum Mathematicarum 12 (1950) 30–37.

[4] A. H. Stroud, D. D. Stancu, Quadrature formulas with multiple Gaussian nodes, SIAM Journal on Numerical Analysis 2 (1965) 129–143.

[5] E. Hairer, G. Wanner, Multistep-multistage-multiderivative methods for ordinary differential equations, Computing (Arch. Elektron. Rechnen) 11 (1973) 287–303.

[6] J. Chouchoulis, J. Schütz, J. Zeifang, Jacobian-free explicit multiderivative Runge–Kutta methods for hyperbolic conservation laws, Journal of Scientific Computing 90 (2022) 96.

[7] J. Schütz, D. Seal, An asymptotic preserving semi-implicit multiderivative solver, Applied Numerical Mathematics 160 (2021) 84–101.

[8] J. Schütz, D. C. Seal, J. Zeifang, Parallel-in-time high-order multiderivative IMEX solvers, Journal of Scientific Computing 90 (2022) 1–33.

[9] J. Zeifang, J. Schütz, D. Seal, Stability of implicit multiderivative deferred correction methods, BIT Numerical Mathematics (2022).

[10] S. Gottlieb, Z. J. Grant, J. Hu, R. Shu, High order strong stability preserving multiderivative implicit and IMEX Runge–Kutta methods with asymptotic preserving properties, SIAM Journal on Numerical Analysis 60 (2022) 423–449.

[11] R. Spiteri, S. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods, SIAM Journal on Numerical Analysis 40 (2002) 469–491.

[12] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Review 43 (2001) 89–112.

[13] A. J. Christlieb, S. Gottlieb, Z. J. Grant, D. C. Seal, Explicit strong stability preserving multistage two-derivative time-stepping schemes, Journal of Scientific Computing 68 (2016) 914–942.

[14] Z. Grant, S. Gottlieb, D. C. Seal, A strong stability preserving analysis for explicit multistage two-derivative time-stepping schemes based on Taylor series conditions, Communications on Applied Mathematics and Computation 1 (2019) 21–59.

[15] A. Moradi, J. Farzi, A. Abdi, Strong stability preserving second derivative general linear methods, Journal of Scientific Computing 81 (2019) 392–435.

[16] J. Schütz, D. Seal, A. Jaust, Implicit multiderivative collocation solvers for linear partial differential equations with discontinuous Galerkin spatial discretizations, Journal of Scientific Computing 73 (2017) 1145–1163.

[17] A. Jaust, J. Schütz, D. C. Seal, Implicit multistage two-derivative discontinuous Galerkin schemes for viscous conservation laws, Journal of Scientific Computing 69 (2016) 866–891.

[18] N. C. Nguyen, J. Peraire, B. Cockburn, High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics, Journal of Computational Physics 230 (2011) 3695–3718.

[19] J. Zeifang, J. Schütz, Implicit two-derivative deferred correction time discretization for the discontinuous Galerkin method, Journal of Computational Physics 464 (2022) 111353.

[20] D. A. Kopriva, Implementing spectral methods for partial differential equations: Algorithms for scientists and engineers, Springer Science & Business Media, 2009.

[21] J. Zeifang, A. Thenery Manikantan, J. Schütz, Time parallelism and Newton-adaptivity of the two-derivative deferred correction discontinuous Galerkin method, UHasselt Preprint UP2201 (2022). URL: `https://www.uhasselt.be/media/3snkyq1s/up2201.pdf`.

[22] E. Hairer, G. Wanner, Solving ordinary differential equations II, Springer Series in Computational Mathematics, 1991.

[23] F. Hindenlang, G. Gassner, C. Altmann, A. Beck, M. Staudenmaier, C.-D. Munz, Explicit discontinuous Galerkin methods for unsteady problems, Computers & Fluids 61 (2012) 86–93.

[24] N. Krais, A. Beck, T. Bolemann, H. Frank, D. Flad, G. Gassner, F. Hindenlang, M. Hoffmann, T. Kuhn, M. Sonntag, et al., FLEXI: A high order discontinuous Galerkin framework for hyperbolic–parabolic conservation laws, Computers & Mathematics with Applications 81 (2021) 186–219.

[25] M. Franciolini, A. Crivellini, A. Nigro, On the efficiency of a matrix-free linearly implicit time integration strategy for high-order discontinuous Galerkin solutions of incompressible turbulent flows, Computers & Fluids 159 (2017) 276–294.

[26] D. A. Knoll, D. E. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, Journal of Computational Physics 193 (2004) 357–397.

[27] K. Kaiser, J. Schütz, A high-order method for weakly compressible flows, Communications in Computational Physics 22 (2017) 1150–1174.

[28] F. Bassi, S. Rebay, G. Mariotti, S. Pedinotti, M. Savini, A high-order accurate discontinuous Finite Element method for inviscid and viscous turbomachinery flows, Proceedings of 2nd European Conference on Turbomachinery, Fluid Dynamics and Thermodynamics (1997) 99–108.

[29] M. Carpenter, C. Kennedy, Fourth-order $2N$-storage Runge-Kutta schemes, Technical Report, NASA Langley Research Center, 1994.

[30] A. Moradi, A. Abdi, J. Farzi, Strong stability preserving second derivative general linear methods with Runge–Kutta stability, Journal of Scientific Computing 85 (2020) 1.

[31] A. Moradi, A. Abdi, G. Hojjati, Strong stability preserving implicit and implicit–explicit second derivative general linear methods with RK stability, Computational and Applied Mathematics 41 (2022) 135.

UHasselt Computational Mathematics Preprint Series

www.uhasselt.be/cmat