Optimal time integration methods for computational aeroacoustics

CES-Seminar

Björn Peeters

Supervised by: Michael Schlottke-Lakemper

Chair of Fluid Mechanics and
Institute of Aerodynamics Aachen

Aachen, 2016

Ich versichere, dass ich die Arbeit selbstständig mit der Unterstützung der Betreuer verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

I hereby declare that I have written this work independently with the support by my supervisors and without contributions from any sources other than cited in the text and the acknowledgements.

Aachen, September 2016

Contents

1 Introduction

Flow-induced noise is attracting increasing attention in various industries for diverse motives. In aerospace engineering, for instance, noise arises from the flow around the sharp edges of airfoils or the blades of turbines. Reducing this flow-induced noise is the basis for airplanes that not only comply with standards and statutory regulations but have a reduced noise level inside the cabin.
Such cases can be simulated using computational aeroacoustics (CAA) in order to determine the generation and propagation of acoustic waves in a flow field. There are in general two approaches to CAA, given by the direct numerical simulation (DNS) and the hybrid methods. The DNS directly calculates the flow field and the acoustic wave propagation from the Navier-Stokes equations. Thus it resolves the small length-scales of the flow as well as the large acoustic scales and thereby becomes unsuitable for larger problems due to general limitations in memory and runtime. In contrast to this, in the hybrid approach the flow field is obtained from a computational fluid dynamics (CFD) simulation from which acoustic source terms are extracted and used e.g. in the acoustic perturbation equations (APE)[1] to determine the acoustic field. The coupling of the solvers in the hybrid case requires the exchange of data at each time step. Normally, the time step size of the CFD simulation is larger than the one from the CAA thus intermediate evaluations of the CFD solver are required (Fig. 1). This is expensive in terms of computational effort and is sought to be avoided, i.e. with larger time step sizes of the CAA simulation.

In this work different time integration schemes are compared in order to obtain a more optimal, with regard to runtime and time step size, time integration of the acoustics simulation. The numerical methods of computational aeroacoustics are introduced in Sec. 2. Following this, the low-storage Runge-Kutta schemes for temporal discretisation are described in Sec. 3. Afterwards their simulation results are evaluated in Sec. 4 with regard to their potential for fewer intermediate time steps.

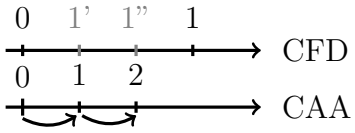


Figure 1: Temporal coupling of CFD and CAA solver

2 Numerical methods

In the hyrid method the CFD simulation yields the turbulent flow field. To further obtain the acoustic field by means of the APE a discretisation of the equations in space and time is required. The specific numerical methods of the acoustic solver, which is part of the multiphysics framework ZFS, used in this work are outlined hereafter.

2.1 Spatial discretisation

The spatial discretisation is performed with a nodal discontinuous Galerkin (DG) method [2], which utilizes polynomials for the approximation of the exact solution that are allowed to be discontinuous at cell boundary. It is used in favour of a classical finite volume discretisation as its order of accuracy solely depends on the polynomial degree. Further details on the DG method and its implementation can be found in [3], [4] and [5].

2.2 Temporal discretisation

For the discretisation in time a Runge-Kutta approach is used, which is an explicit multi-stage single-step method utilized for solving ordinary differential equations (ODE). A Runge-Kutta scheme consists of m stages, for which the semi-discrete DG operator F^n, corresponding to the spatial discretisation, is evaluated. The numerical solution and intermediate values are denoted by u^n and y_i respectively. The parameter Δt is the time step size.

$$\begin{aligned} y_i &= u^n + \Delta t \sum_{j=1}^{i-1} a_{ij} F^{n+1}(y_j) \quad \forall i \in 1, ..., m \\ u^{n+1} &= u^n + \Delta t \sum_{j=1}^{m} b_j F^{n+1}(y_j) \end{aligned} \tag{1}$$

The coefficients a_{ij} and b_j are method-specific values, usually provided in form of a Butcher tableau, that are obtained with regard to certain objectives, such as the resulting order of accuracy.
In general, increasing the number of stages leads to a higher order of accuracy and better stability properties with the drawback of additionally required memory.

3 Runge-Kutta methods

3.1 Low-storage Runge-Kutta methods

A straightforward implementation of the Runge-Kutta scheme (Eq. 1) requires a memory storage of size $(m+1) \times (degrees\ of\ freedom \cdot no\ of\ variables)$ for the solution of all stages. Therefore its utilization becomes infeasible for large scale simulations due to memory restrictions.

This motivates so-called low-storage Runge-Kutta methods that only require 2N memory registers, with N being the number of equations of the differential operator times the degrees of freedom, while preserving the order of accuracy and stability of the original form. Two different low-storage algorithms were derived by Williamson [6] and van der Houwen [7], denoted as 2N- and 2R-methods, following the notation in [8]. The Williamson scheme (Algo. 1) resembles the classical Runge-Kutta approach [9] with a focus on the reduction of computational cost for the evaluation of each stage. It stores the intermediate variables S_1, S_2 and the value of the function $F(S_1)$ for each stage. The implementation of van der Houwen (Algo. 2) makes a more efficient use of the two registers by alternating between them at each stage and thereby reducing the necessary memory storage. Compared to the Williamson 2N-method the previous solution of a stage is not preserved, which is a drawback for methods that rely upon error estimation. To make this feature available an additional register can be added thus increasing the required memory.

$S_1 := u^n \qquad \rightarrow y_1$
for $i = 2$ to $m+1$ **do**
  $S_2 := a_i S_2 + \Delta t F(S_1)$
  $S_1 := S_1 + b_i S_2 \qquad \rightarrow y_i$
end for
$u^{n+1} = S_1$

Algorithm 1: Williamson (2N)

$S_2 := u^n \qquad \rightarrow y_1$
for $i = 1$ to m **do**
  $S_1 := S_2 + (a_{i,i-1} - b_{i-1})\Delta t S_1 \qquad \rightarrow y_i$
  $S_1 := F(S_1)$
  $S_2 := S_2 + b_i \Delta t S_1$
end for
$u^{n+1} = S_2$

Algorithm 2: Van der Houwen (2R)

3.2 Low-dissipation and low-dispersion Runge-Kutta methods

The classical fourth-order Runge-Kutta method features desired large stability limits but it exhibits deficiencies when applied to aeroacoustic problems. Acoustic waves are nondissipative, nondispersive and propagate in each direction, though the Runge-Kutta

methods induce dissipation and dispersion errors. Resolving this typically requires a higher-order scheme and small time step sizes, which leads to the issue of increasing requirements in time and memory. In order to diminish those effect Hu, Hussaini and Manthey [10] developed the so called low-dissipation and low-dispersion Runge-Kutta (LDDRK) schemes. They applied the Runge-Kutta method to a convective wave equation and obtained the exact and numerical amplification factor. Those results where related to the generated dissipation and dispersion errors and yielded optimized coefficients.
Based on the LDDRK scheme, Stanescu and Habashi [11] derived low-storage implementations for different numbers of stages.

3.3 Runge-Kutta schemes

For the evaluation of time integration with regard to runtime and number of time steps, several schemes were utilized (Table 1). The low-storage Carpenter method [12], which is currently the default time integration scheme in the acoustics solver of ZFS, was used as a reference for all simulations throughout this work.
The LDD456 two-step method [11] is a LDDRK scheme which alternates between 5 and 6 stages for each evaluation, based on the observation that this technique yields a higher accuracy and larger possible time step size. The method of Calvo [13] was constructed with respect to similar objectives but uses the van der Houwen scheme. Tselios and Simos [14] derived a general function that can be reformulated into the other fourth-order six-stage methods. Their new method is deduced from this function by searching for coefficients that ensure minimal dissipation and dispersion while obtaining a maximal order of accuracy and a large stability interval.

Name	Abb.	Steps	Stages	Order	Algorithm
Carpenter	RK45	1	5	4	2N
Low-dissipation/low-dispersion	LDD25	1	5	2	2N
Low-dissipation/low-dispersion	LDD46	1	6	4	2N
Two-step method	LDD456	2	5/6	4	2N
Calvo	LDD46*	1	6	4	2R
Tselios, Simos	LDD47	1	7	4	2R

Table 1: Low-storage Runge-Kutta schemes

4 Results and evaluation

In order to compare the behaviour of the different Runge-Kutta schemes with regard to accuracy and time step size, each algorithm given in Table 1 was implemented in ZFS. The time step size Δt of each time step is calculated depending on the CFL number and the polynomial degree of the discretisation.
All results are obtained for the respective stability limit of each method, which was derived by varying the CFL number and finding its maximal permitted value that yields a convergent solution of the test case.

4.1 Linear advection

The one-dimensional linear advection equation (Eq. 2) is simulated with advection velocity $a = 1$ for the initial condition $u_0 = u(t = 0) = 0.5e^{-(x/3)^2}$. The analytical solution is given by $u(t, x) = 0.5e^{-((x-at)/3)^2}$, which resembles a shift of the initial condition by at.

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{2}$$

A polynomial degree of $p = 3$ with a reference spatial discretisation R1 is used in the test case and the solution at time $t = 50s$ is sought. The results (Fig. 2) indicate that all schemes yield similar outcomes though none meets the exact solution. At the maximum, both 2R-methods are slightly more accurate than the reference method however in the rest of the domain the result of the Carpenter scheme resembles the exact solution more accurately than the other ones. Clearly visible are oscillations ahead and behind the peak, which corresponds to the dispersion error.

A distinct difference can be observed regarding the CFL numbers (Table 2) with the stability limit for $LDD46*$ and $LDD47$ being larger than the one for $RK45$, which confirms the conclusions drawn in [13] and [14]. This indicates possible larger time step sizes for the 2R-methods. For investigating the behaviour of the schemes on a finer grid

Poly. degree and discr.	RK45	LDD25	LDD46	LDD456	LDD46*	LDD47
p = 3, R1	1.811	1.567	1.767	1.660	2.108	2.606
p = 3, R2	1.806	1.556	1.708	1.647	2.028	2.537
p = 8, R1	1.586	1.416	1.585	1.517	1.897	2.406

Table 2: Linear advection: CFL numbers at stability limit

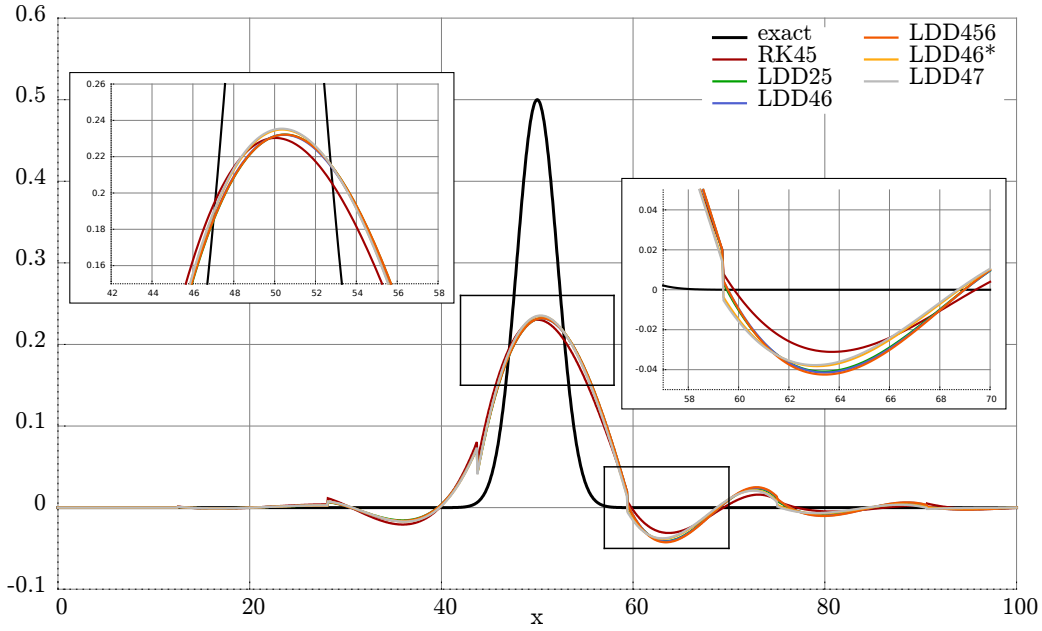


Figure 2: Linear advection at $t = 50s$, $p = 3$ and R1

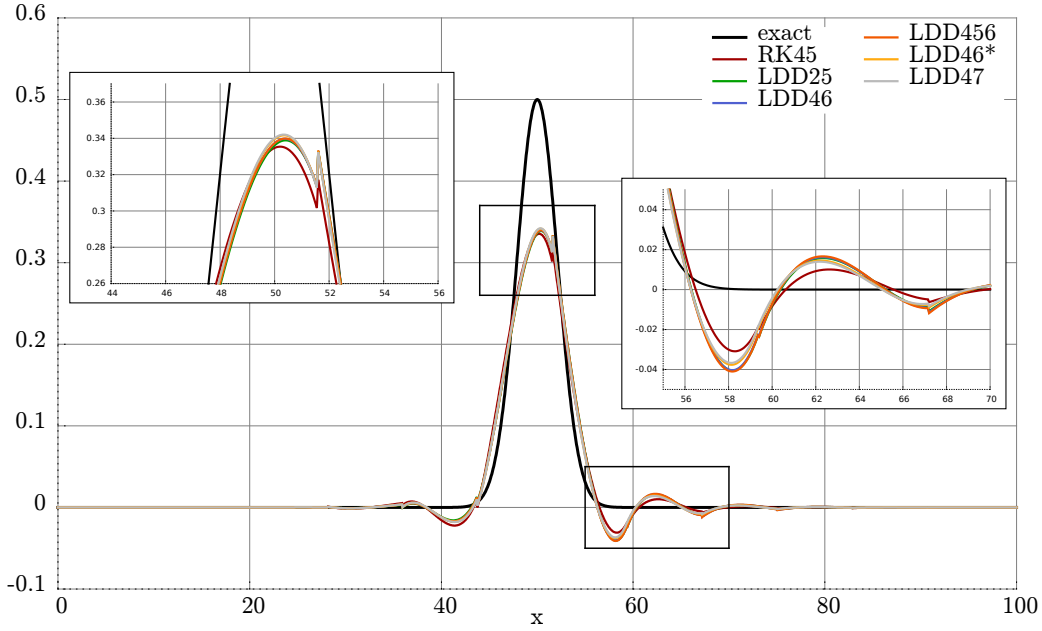
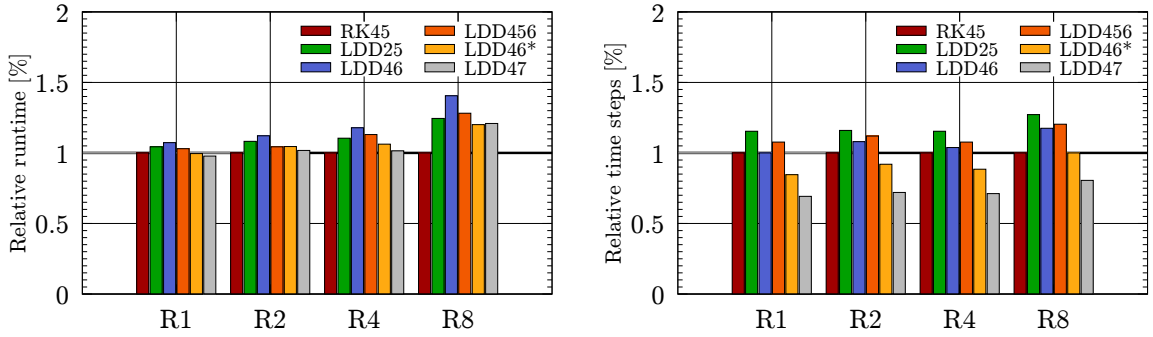


Figure 3: Linear advection at $t = 50s$, $p = 3$ and $R2$

the spatial discretisation was doubled for the next case, denoted by R2. Compared to the previous case the results (Fig. 3) resemble the analytical solution noticeable better, with a more accurate slope in the centre. In contrast to the last configuration, the oscillations occur in a smaller range on both sides of the peak. The increased spatial resolution seems also to be conjoined with the maximum stable time step sizes, which are varyingly pronounced for the different Runge-Kutta schemes.
Besides accuracy, the runtime and the number of required time steps are important criteria for evaluating the usefulness of a scheme. For reasons of comparability, the values in Fig. 4a-4b are normalized with the results from the reference. The Runge-Kutta methods were used in a test case with $p = 3$ and a spatial resolution that is doubled in each subsequent setup, denoted by $R1$, $R2$, $R4$ and $R8$. As can be seen in Fig. 4b the *LDD46** and *LDD47* schemes need significantly less evaluations than the *RK45* model, which can be explained by their larger stability interval thus larger time step size. A comparison of their runtimes (Fig. 4a) shows just a small advantage for the Carpenter implementation, especially for the coarser discretisations thus making the 2R-methods good candidates for optimal time integration methods. Moreover, both plots illustrate that the other three methods, namely *LDD25*, *LDD46* and *LDD456*, seem to be irrelevant for further consideration.

To determine whether the polynomial degree has an effect on the accuracy of the numerical solution, it was changed from $p = 3$ to $p = 8$ while retaining the original grid. As Fig. 5 illustrates, the solutions in the centre approximate the exact solution of the linear advection much better than the previous settings. Furthermore, the dispersion error is less pronounced for all schemes. Combining $p = 8$ with an increased spatial dis-



(a) Runtime

(b) Time steps

Figure 4: Linear advection: Relative runtime and relative time steps

cretisation leads to solutions that are no longer distinguishable, but still differ in their runtime and their number of time steps.
The conclusions drawn up earlier for runtime and time steps also hold for $p = 8$, only differing in their absolute values, and are therefore not shown.

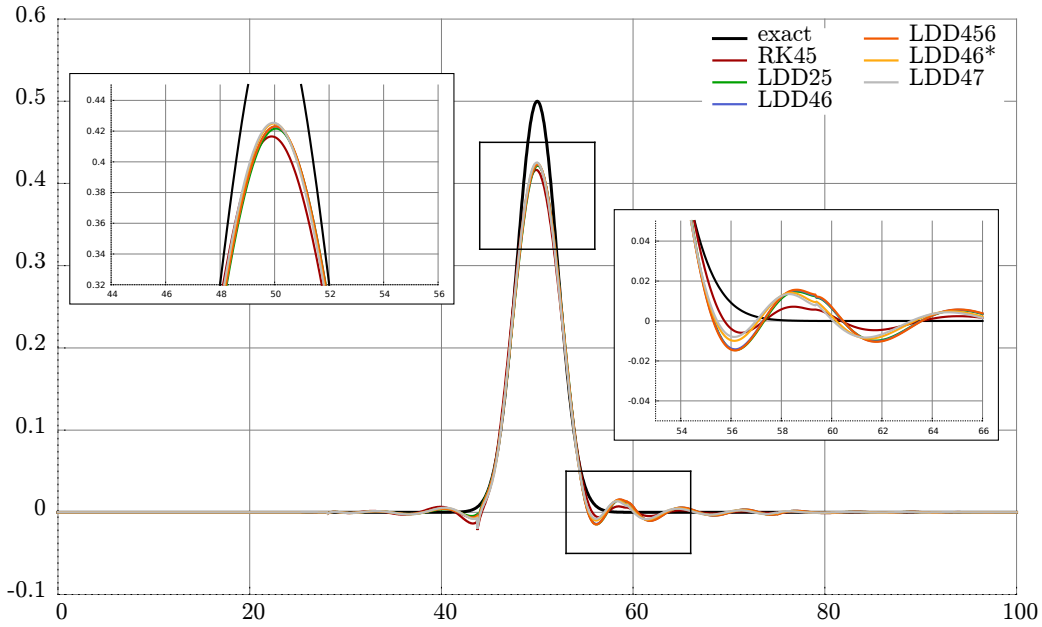


Figure 5: Linear advection at $t = 50s$, $p = 8$ and R1

4.2 Acoustic perturbation equations

Now, the sound propagation of an Gaussian pressure pulse in two dimensions with the APE is considered. A mean velocity of $\bar{u} = 0.5$ in positive x-direction is used. The case is simulated until $t = 30s$ with polynomial degree $p = 3$ and a reference spatial discretisation R1. Simulation results along the x-axis are shown.
As Fig. 6 indicates, all schemes yield reasonable results with some inaccuracies following from element-to-element oscillations. The method with the largest amplitude is $LDD25$ followed by $LDD46*$ and $LDD456$, whereas the reference solution and $LDD47$ show almost no oscillations. Table 3 additionally indicates a larger stable CFL number for

Poly. degree and discr.	RK45	LDD25	LDD46	LDD456	LDD46*	LDD47
p = 3, R1	1.736	1.204	1.297	1.279	1.553	1.891

Table 3: Gaussian pulse: CFL numbers at stability limit

the scheme of Tselios in comparison to all other approaches. Looking at the runtime (Fig. 7a) it can be seen that the simulation with this method takes longer than the reference one. However, similar to the last test cases the $LDD47$ approach needs fewer time steps, possibly leading to a reduction of intermediate time steps and thereby to a reduced runtime of the hybrid simulation.

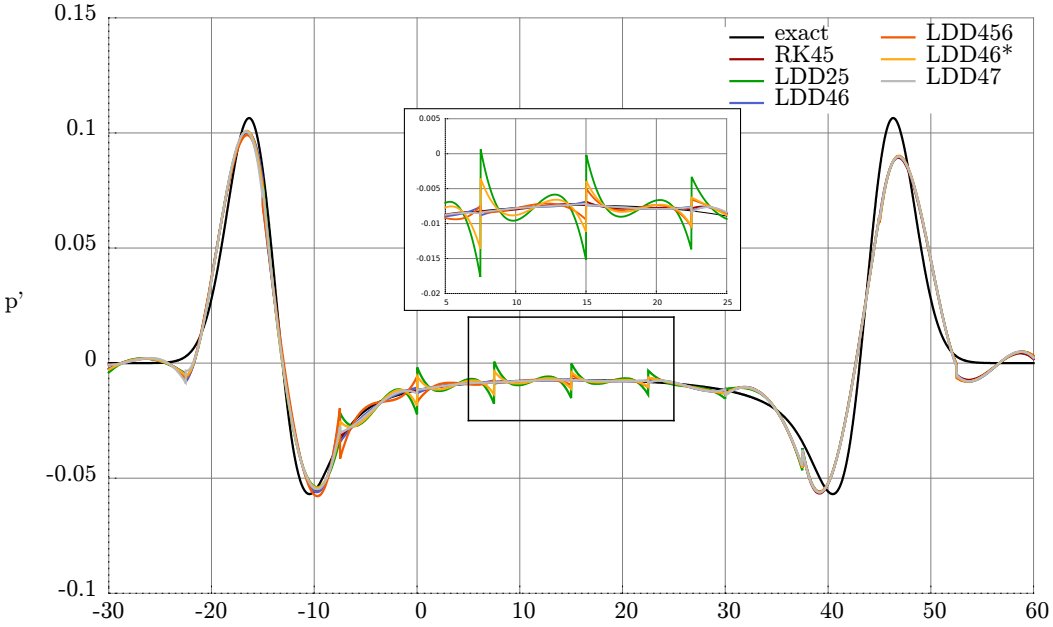
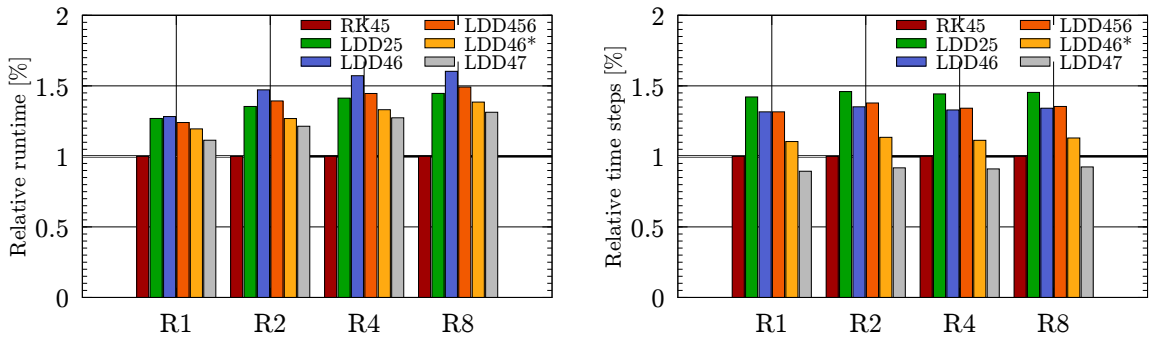


Figure 6: Gaussian pulse at $t = 30s$ with $\bar{u} = 0.5$, $p = 3$ and R1



(a) Runtime

(b) Time steps

Figure 7: Gaussian pulse: Relative runtime and relative time steps

5 Conclusions and outlook

In this work different Runge-Kutta schemes for time integration were evaluated regarding their suitability in computational aeroacoustics. They were implemented in ZFS and used in several test cases. The results were compared to the currently implemented approach with regard to their accuracy, runtime and number of time steps for the maximum stable setting.

The evaluation of the low-storage Runge-Kutta schemes indicates that the Carpenter implementation is still the best approach if a fast runtime is desired. This is due to its low number of stages combined with a high order of accuracy. If the objective is a low number of time steps the method from Tselios et al. is an optimal choice as it allows larger time step sizes. This could potentially reduce the number of intermediate time steps for the exchange between the CFD and CAA solver for the hybrid approach thus reducing the total runtime.

In order to analyse whether this also holds for other settings, like 3D test cases, further research needs to be conducted. Possible other problems are for example hybrid simulations for the simulation of wideband noise.

Another option to find approaches that need less evaluations in time might be the low-storage strong stability preserving (SSP) methods.

6 Bibliography

[1] R. Ewert and W. Schröder. Acoustic perturbation equations based on flow decomposition via source filtering. *Journal of Computational Physics*, 188(2):365 – 398, 2003.

[2] Michael Schlottke-Lakemper, Matthias Meinke, and Wolfgang Schröder. *A Hybrid Discontinuous Galerkin-Finite Volume Method for Computational Aeroacoustics*, pages 743–753. Springer International Publishing, 2016.

[3] J.S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer, 2007.

[4] D. Kopriva. *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Scientific Computation. Springer, 2009.

[5] G. Gassner. Discontinuous-Galerkin-Verfahren, *Institute of Aerodynamics and Gas Dynamics, Universität Stuttgart*, 2013.

[6] J. H. Williamson. Low-storage Runge–Kutta schemes. 35(1):48–56, March 1980.

[7] P.J. van der Houwen. *Construction of integration formulas for initial value problems*. North-Holland, 1977.

[8] David I. Ketcheson. Runge-Kutta methods with minimum storage implementations. *Journal of Computational Physics*, 229(5):1763 – 1773, 2010.

[9] J. C. Butcher. *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta and General Linear Methods*. Wiley-Interscience, 1987.

[10] F.Q. Hu, M.Y. Hussaini, and J.L. Manthey. Low-dissipation and low-dispersion Runge-Kutta schemes for computational acoustics. *Journal of Computational Physics*, 124(1):177 – 191, 1996.

[11] D. Stanescu and W.G. Habashi. 2N-storage low dissipation and dispersion Runge-Kutta schemes for computational acoustics. *Journal of Computational Physics*, 143(2):674 – 681, 1998.

[12] Mark H Carpenter and Christopher A Kennedy. Fourth-order 2N-storage Runge-Kutta schemes. 1994.

[13] M. Calvo, J.M. Franco, and L. Rández. A new minimum storage Runge-Kutta scheme for computational acoustics. *Journal of Computational Physics*, 201(1):1 – 12, 2004.

[14] K. Tselios and T.E. Simos. Optimized Runge-Kutta methods with minimal dispersion and dissipation for problems arising from computational acoustics. *Physics Letters A*, 363(1-2):38 – 47, 2007.