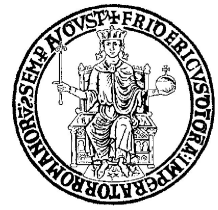
Development of high-fidelity numerical methods for turbulent flows simulation

FRANCESCO CAPUANO



PhD Thesis
University of Naples Federico II

UNIVERSITÀ DI NAPOLI FEDERICO II

DOTTORATO DI RICERCA IN
INGEGNERIA AEROSPAZIALE, NAVALE
E DELLA QUALITA' – XXVII CICLO

DEVELOPMENT OF HIGH-FIDELITY NUMERICAL METHODS FOR TURBULENT FLOWS SIMULATION

FRANCESCO CAPUANO

Supervisors (UniNa):
Prof. L. DE LUCA
Prof. G. COPPOLA

Supervisors (CIRA):
Dr. A. SCHETTINO
Dr. S. BORRELLI

Coordinatore:
Prof. **L. DE LUCA**
Department of Industrial Engineering – Aerospace Sector

March 2015

Development of high-fidelity numerical methods for turbulent flows simulation –

Francesco Capuano

Typesetting: LaTeX

Preface

This work has been supported and promoted by the Italian Aerospace Research Center (CIRA), where I work since 2011.

The main objective of the research activity has been to develop advanced numerical methods and efficient computational tools for high-fidelity simulations of turbulent flows, particularly with Large-Eddy Simulation (LES) and Direct Numerical Simulation (DNS) techniques. The exponential rise in computing power over the last decades has enabled unsteady calculations of turbulent flows in complex configurations. The continued developments in numerical and physical modeling have allowed LES and DNS to establish as predictive tools for both engineering analyses and scientific investigations. However, additional research efforts are required for the construction of efficient and accurate numerical methods, that are regarded as the key to the success of high-fidelity computation of turbulent flows.

The research activity was articulated into two main subjects.

The first one has been concerned with the development of a computational LES code aimed at the simulation of compressible reacting flows with high-accuracy and high-performance requirements. The work has been carried out in the framework of the HYPROB program, funded by the Italian Ministry of Research. The developed software, named SPARK-LES, is a finite-volume code that solves the compressible, multi-species, reacting Navier-Stokes equations on structured multi-block grids. The bulk of the work has focused on the construction of the computational core: centered high-order explicit and compact schemes for space integration have been implemented, verified and validated on basic test-cases. SPARK-LES features an MPI parallel implementation and makes full use of Fortran 90 capabilities for CPU efficiency. The code has demonstrated excellent performances and high accuracy on classical turbulence benchmarks, such as the Taylor-Green vortex, the channel flow and the temporal mixing layer. Future work includes reacting test-cases as well as implementation of advanced physical and numerical modeling.

The second research subject has focused on the construction of efficient numerical schemes with discrete energy-conservation properties for the incompressible Navier-Stokes equations. Energy-conserving discretizations are widely regarded as a fundamental requirement for high-fidelity simulations of turbulent flows. A well-known approach to obtain semi-discrete conservation of energy in the inviscid limit is to employ the skew-symmetric splitting of the non-linear term. However, this approach has the drawback of being roughly twice as expensive as the computation of the divergence

or advective forms alone. A novel time-advancement strategy that retains the conservation properties of skew-symmetric-based schemes at a reduced computational cost has been developed. This method is based on properly constructed Runge-Kutta schemes in which a different form (advective or divergence) for the convective term is adopted at each stage. A general theoretical framework has been developed to derive new schemes with prescribed accuracy on both solution and energy conservation. The technique has been first developed and fine-tuned on the Burgers' equation, and then applied to the incompressible Navier-Stokes equations. Simulations of homogeneous isotropic turbulence performed at the Center for Turbulence Research (CTR) in Stanford have demonstrated that the novel procedure provides the same robustness of the skew-symmetric form while halving the computational cost for the non-linear term.

The following publications have arisen from the research activities:

[i] F. CAPUANO, G. COPPOLA, AND L. DE LUCA. **Quasi-symplectic Runge-Kutta methods for incompressible Navier-Stokes equations**. *J. Comput. Phys.*, in preparation.

[ii] F. CAPUANO, G. COPPOLA, G. BALARAC AND L. DE LUCA. **Energy-preserving turbulent simulations at a reduced computational cost**. *J. Comput. Phys.*, 19 pp., under review.

[iii] F. CAPUANO, G. COPPOLA AND L. DE LUCA. **An efficient time advancing strategy for energy-preserving simulations**. *J. Comput. Phys.*, 30 pp., in press.

[iv] F. CAPUANO AND A. MASTELLONE. **Parallelization of finite-volume compact schemes for turbulent compressible flow**. Parallel CFD 2015, Montreal, May 17-20, 2015.

[v] A. SCHETTINO, L. CUTRONE, S. DI BENEDETTO, F. CAPUANO, A. MASTELLONE, G. SACCONE AND G. DI LORENZO. **Development and validation of numerical solvers for the analysis of high-pressure combustion in liquid rocket engines**. 8th European Symposium on Aerothermodynamics for Space Vehicles", Lisbon, March 2-6, 2015, 8 pp.

[vi] F. CAPUANO, G. COPPOLA, L. DE LUCA AND G. BALARAC. **A low-cost RK time advancing strategy for energy-preserving turbulent simulations**. *Bullettin of the American Physical Society*, 11/2014, **59**(20), p.193.

[vii] F. CAPUANO, G. COPPOLA, G. BALARAC AND L. DE LUCA. **A low-cost time-advancing strategy for energy-preserving turbulent simulations**. *Proceedings of the 2014 Summer Program*, Center for Turbulence Research, Stanford, pp. 377–386.

[viii] F. CAPUANO, A. MASTELLONE, S. DI BENEDETTO, L. CUTRONE AND A. SCHETTINO. **Preliminary developments towards a high-order and efficient LES code for propulsion applications**. *Proceedings of the jointly organized WCCM XI - ECCM V - ECFD VI*, Barcelona, July 20-25, 2014.

[ix] F. CAPUANO, G. COPPOLA AND L. DE LUCA. **Low-cost energy-preserving RK schemes for turbulent simulations**. *Progress in Turbulence VI*, Proceedings iTi Conference on Turbulence, Bertinoro, September 21-24, 2014, 4pp.

[x] A. MASTELLONE, F. CAPUANO, S. DI BENEDETTO AND L. CUTRONE. **Direct numerical simulation of the Taylor-Green Vortex at Re = 1600**. 3rd International Workshop on High-Order CFD Methods, Kissimmee, January 3-4, 2015.

[xi] F. CAPUANO, G. COPPOLA AND L. DE LUCA. **Effects of numerical errors in Large-Eddy Simulation of transitional wall-bounded flows**. European Turbulence Conference 14, Lyon, September 1-4, 2013.

In the following list, the major original accomplishments are summarized, along with bibliographic information to which the interested reader is referred.

- Design, development and validation of the computational code SPARK-LES ([v], [viii], [x]);

- Development of new approaches for parallelization of finite-volume compact schemes ([iv]);
- Development of a theoretical framework for efficient energy-preserving Runge-Kutta time-advancement schemes (Chapter 3);
- Derivation of energy-preserving Runge-Kutta Butcher arrays for Burgers' and Navier-Stokes equations (Chapter 4);
- Derivation of quasi-symplectic Runge-Kutta schemes for incompressible Navier-Stokes equations ([i]);
- Development of a cost metric for the Runge-Kutta integration of the convective term (Appendix A);
- Development of an efficient approximate projection method for the incompressible Navier-Stokes equations (Appendix C).

The content of this thesis is focused on the second of the above-mentioned subjects, and is mainly based on publications [ii] and [iii]. Excerpts from these papers are used with permission from the publisher.

Many people contributed to this work.

I would like to sincerely and gratefully acknowledge my academic supervisors, Prof. Luigi de Luca and Prof. Gennaro Coppola, for the encouragement they gave me, their generous support and polished manners, and for their commitment to excellence in research. Their guidance throughout the last three years has indefinitely contributed to my personal and scientific growth.

Many thanks also to my tutors at CIRA, Dr. Antonio Schettino and Dr. Salvatore Borrelli, for their support and interest, as well as to all my colleagues with whom I had profitable interactions. I would like to especially thank Andrea Mastellone, without whom the development of SPARK-LES could certainly not have been accomplished.

I would also like to thank Prof. Parviz Moin for the unique opporunity he gave me to work at the Center for Turbulence Research in Stanford during the 2014 Summer Program, as well as for fruitful scientific discussions. I would also like to acknowledge all the professors and post-graduate students I cooperated with at Stanford; among them: Jane Bae, Guillaume Balarac, Sanjeeb Bose, Roel Verstappen, Wybe Rozema, Curtis Hamman, Armin Zare, Mihailo Jovanovic, Stefan Domino.

Finally, I would like to thank my family, Elena and my friends. To these three — in no particular order — I owe most of my personal achievements.

Napoli, March 2015 Francesco Capuano

Ai miei nonni

Contents

Nomenclature

Roman Symbols

$\mathbf{1}$	Column vector of ones
$\mathbf{U}$	Diagonal matrix of the discrete velocity vector
$\mathbf{u}$	Discrete velocity vector
$\hat{a}$	Number of non-zero a_{ij} Runge-Kutta coefficients
$\hat{b}$	Number of non-zero b_i Runge-Kutta coefficients
$\mathcal{N}$	Continuous convective term
a_{ij}	Runge-Kutta coefficients
b_i	Runge-Kutta coefficients
c_i	Runge-Kutta coefficients
e	Global energy
L	Length of the stencil for a finite-difference scheme
M	Discrete global momentum
m	Global momentum
N	Number of mesh points
O	Number of operations
p	Pressure or order of accuracy
R	Linear stability function
s	Number of stages of the Runge-Kutta scheme
T	Computational time
t	Time
u	Continuous velocity field
x, ymz	Spatial coordinates
E	Discrete global energy
Re	Reynolds number

Greek Symbols

ΔE	Discrete energy variation
Δt	Time step
Δx	Mesh size
μ	Four-stage Runge-Kutta parameter
ν	Kinematic viscosity
∂_t	Temporal partial derivative

∂_x	Spatial partial derivative
θ	Parameter
Θ_1	First-order term of the total energy error
Θ_2	Second-order term of the total energy error
Θ_s	Spatial energy error
Θ_t	Temporal energy error
ε	Error measure of energy conservation

Superscripts

n	Time level
sp	Spatial
T	Transpose
tot	Total

Subscripts

0	Initial
a, adv	Advective form
d, div	Divergence form
i, j	Runge-Kutta stages
skw	Skew-symmetric form
x, y, z	Spatial directions

Discrete operators

$\mathbf{C}$	Discrete convective operator
$\boldsymbol{\mathcal{A}}$	Discrete convective operator in advective form
$\boldsymbol{\mathcal{D}}$	Discrete convective operator in divergence form
$\boldsymbol{\mathcal{S}}$	Discrete convective operator in skew-symmetric form
$\boldsymbol{\mathcal{C}}$	Discrete 1D convective operator
$\boldsymbol{\mathcal{L}}$	Discrete 1D Laplacian operator
$\mathbf{D}$	Discrete derivative operator
$\mathbf{G}$	Discrete gradient operator
$\mathbf{L}$	Discrete 3D Laplacian operator
$\mathbf{M}$	Discrete divergence operator
$\mathbf{P}$	Projection operator
$\tilde{\cdot}$	Projected operator

Acronyms

1D	One-dimensional
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
LES	Large-Eddy Simulation
NS	Navier-Stokes
RANS	Reynolds-Averaged Navier-Stokes
RK	Runge-Kutta

List of Figures

List of Tables

1. Introduction

Turbulence is one of the most interesting and challenging phenomena in modern physics. Turbulent motions are outstandingly common in almost every aspect of everyday life [1; 2], and a broad class of engineering devices is based on fluids that work in the *turbulent regime* [3; 4]. The advent of supercomputers has pushed the computational fluid dynamics (CFD) towards a reliable numerical resolution of the *Navier-Stokes equations*, that represent a remarkably accurate mathematical model for both transitional and turbulent flow regimes [5].

1.1 Numerical simulation of turbulent flows

Turbulent flows possess a broad range of time and length scales, whose energy is spectrally distributed according to the well-known energy cascade paradigm [6]. A fully time- and space-resolved simulation of the entire mathematical model, equipped with proper initial and boundary conditions, requires that all relevant scales, from integral ones down to the dissipative *Kolmogorov scales*, are properly represented on the discrete level. This approach, called *direct numerical simulation* (DNS), is in general extremely expensive and is nowadays affordable only for low to moderate Reynolds numbers Re and in relatively simple geometries [7]. The required CPU time for a DNS can be roughly estimated as proportional to Re^3. It is estimated that with today's most powerful supercomputers, flows with $\mathrm{Re} = \mathcal{O}(10^4)$ can be directly simulated. Higher Reynolds numbers require an unaffordable computational effort, that will not be mitigated even by the passage from petascale to exascale supercomputing [8].

For high-Reynolds number flows, one must necessarily resort to some sort of modeling. The historical approach has been to apply an averaging process to the governing equations, and solve for the mean value of flow variables. The resulting *Reynolds-averaged Navier-Stokes* (RANS) technique consists in modeling the unclosed terms that arise from application of the averaging operator [9]. Although the RANS approach remains the standard practice in CFD for industrial applications, it suffers from fundamental uncertainties due to the fact that the unclosed terms are modeled phenomenologically. Therefore, RANS is not predictive when applied outside the domain of model calibration, and has reached a plateau that does not benefit from the increased computer power.

An alternative approach emerged in the early 70's, when Deardorff — inspired by Kolmogorov laws — proposed to compute only the largest, energy-carrying spatial scales of turbulence, while modeling the effects of the smaller, universal scales [10].

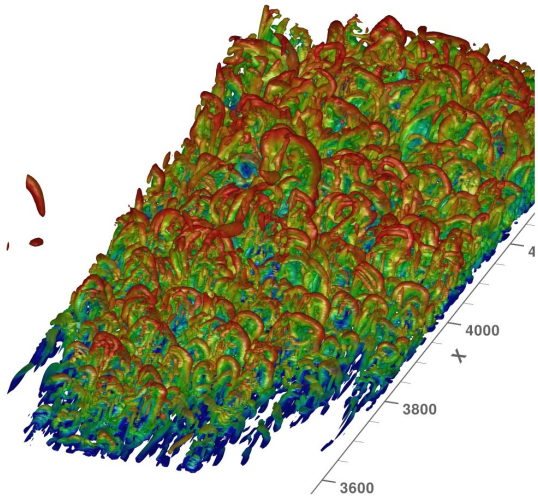


Figure 1.1: Direct numerical simulation of zero-pressure-gradient flat-plate boundary layer. Visualization of the iso-surfaces of the second invariant of the velocity gradient tensor. From [18].

The self-similarity of the smaller scales is supposed to allow an easier and generally applicable modeling. From a mathematical point of view, LES can be obtained by applying a low-pass filtering to the governing equations [11]; the resulting unclosed terms represent the effect of the filtered scales on the resolved ones and are substituted by so-called *subgrid-scale models*. Practically, the filtering operators is usually represented by the discrete derivative operators and the computational grid itself [12]. Unlike RANS models, subgrid-scale models are derived on the basis of consolidated and universal principles, and are therefore much less susceptible to uncertainties. Popular subgrid-scale models are of the eddy-viscosity type and are based on the resolved scales, via either a constant coefficient [13] or a dynamically adjusted parameter [14].

Nowadays, DNS and LES are regarded as the primary scientific tools for turbulence simulation. While DNS is able to provide physical insight but the application to complex configurations is rather limited, LES can be also applied to flows of practical interest and is rapidly expanding in all fields of engineering and science.

1.1.1 Applications and challenges of DNS and LES

DNS has been applied historically to prototype flows; landmark direct numerical simulations include homogeneous isotropic turbulence [15], plane channel flow [16], the backward facing step [17] and the turbulent boundary layer [18], see Fig. 1.1. These simulations shed light on fundamental physical processes and laid the foundations towards more complex situations. More recently, a number of direct numerical simulations of low-Reynolds number flows in slightly more realistic configurations has also been successfully attempted [19].

Large-eddy simulation is applied to a broad class of problems of scientific and practical interest, ranging from aerodynamics to combustion, from aeroacoustics to meterology, as well as many others [20–22]. A particularly interesting application of LES is to turbulent reacting flows [23; 24]. Combustion systems, especially for non-premixed devices, are mainly based on a mixing process between fuel and oxidizer. Combustion takes place when the reactants are mixed at a molecular level, a process that occurs at the smallest turbulent scales. However, the largest scales play a fundamental role as they stir the reactants and increase the scalar gradients. LES is particularly successful in combustion simulations as it directly resolves the largest scales, thus catching the scalar mixing process with remarkable accuracy [25]. As an indirect consequence, the chemical conversion process is also represented much more accurately than in simplified approaches, although reactions occur at the subgrid level and have to be modeled entirely [3].

Many research groups develop LES/DNS codes for multiple applications. A non exhaustive list is reported as follows. The Center for Turbulence Research (CTR) at Stanford performs large-eddy simulations of complex systems with the unstructured code Charles [26]; the Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique (CERFACS, France) is active in the field of numerical simulation of combustion with the code AVBP [27]; the COmplex de Recherche Interprofessionnel en Aérothermochimie (CORIA, France) performs DNS and LES of turbulent combustion by means of the SiTCom-B flow solver [28]. The Italian Aerospace Research Center (CIRA) is also developing qualified technologies to support the design of propulsion systems in the framework of the HYPROB Program [29]. To this aim, a LES code, named SPARK-LES, is being developed with the final aim of simulating turbulent reacting flows within liquid rocket thrust chambers [30].

Despite the extraordinary achievements already accomplished by DNS and LES, several issues have still to be addressed. Being free of any models, the current challenges for DNS are mainly concerned with the efficiency of solution algorithms on highly parallel systems and with the accuracy of numerical methods. On the other hand, large-eddy simulation has reached satisfactory predictive capabilities, but additional research efforts are still required in terms of physical and numerical modeling. Since in LES the energy cascade process is truncated within the inertial range, a significant amount of energy is present at the high-end of the wavenumber spectrum supported by the grid. As a consequence, numerical discretization methods have to ensure the subtle balances among non-linear interactions, molecular dissipation and subgrid-scale models without altering the underlying physics. This is a very challenging task for numerical methods and is discussed in more detail in Section 1.1.2. On the physical side, construction of parameter-free subgrid-scale models is also an active field of research, as well as the development of accurate wall-layer models for simulation of high-Reynolds number external flows [31; 32]. Finally, LES also shares with DNS the need for efficient algorithms, a need that will not be mitigated by the presumed passage to exascale computing. As a consequence, development of modeling and numerical methods will be the key to the success of high-fidelity computation

of turbulent flows. There is a widespread feeling that the solution will eventually come (once again) from fundamental research.

1.1.2 Numerical methods

Turbulence is a multi-scale phenomenon that involves delicate energy transfer mechanisms and subtle balances between non-linear inertial forces and dissipative fluxes. As a result, numerical methods used to discretize the Navier-Stokes equations need to be particularly accurate to reproduce the underlying physics without spuriously interacting with the phenomena at play. In particular, the discretization has to provide a good resolution of the scales involved in the simulation and must not introduce spurious dissipation (or production) of energy to ensure that the turbulent cascade is properly reproduced. This is particularly crucial for large-eddy simulation, in which a significant amount of energy is present at the high-end of the wavenumber spectrum supported by the grid. Moreover, in LES the subgrid-scale model must not be overwhelmed by the truncation error of the discretization method.

A straightforward way to meet these requirements is to use high-order approximations of spatial derivatives. In DNS, the solution is often represented by spectral schemes [33], that are allowed by the usually simple geometry and periodic boundary conditions. However, spectral methods have to be abandoned when modeling more complex systems, and one must resort to finite-difference or finite-volume methods [34]. These schemes introduce a lower-order truncation error and support a finite range of wavenumbers [35]. As a remedy, it is widely agreed that LES has to be performed with schemes that are at least second-order accurate in space, so that the truncation error does not overshadow the subgrid-scale model [36]. In order to improve resolution, compact schemes are often preferred, as they provide superior resolution properties on equal order of accuracy if compared to their explicit counterpart [37].

Connected to this is the matter of numerical dissipation. A (linear) modified wavenumber analysis shows that symmetric (or centered) schemes are non-dissipative for all wavenumbers on uniform meshes [38]. On the contrary, upwind schemes add numerical dissipation to the approximate solution at high wavenumbers. Even very high-order upwind methods can quickly dissipate a large fraction of the resolvable turbulent eddies, which is not desirable for simulations of turbulence. Therefore, upwind schemes are generally considered not suitable for large-eddy and direct numerical simulations [39]. Their use is typically limited to the so-called MILES (Monotone-Integrated Large-Eddy Simulation) approach, in which the inherent artificial dissipation provided by the numerical scheme is carefully tuned to reproduce the physical energy cascade from large to small scales (i.e., it works as a built-in subgrid-scale model) [21]. However, this technique is not consolidated and its use is not recommended for high-fidelity simulations.

A problem in using centered high-order schemes is non-linear stability. The amplification of aliasing errors that occur in computing convective terms can in fact accumulate and result in a *pile-up* of the energy at high wavenumbers, eventually causing

the solution to blow up [40]. A successful scheme for large-eddy simulation should be able to suppress these sources of instability while keeping high accuracy and reasonable computational cost. A number of possible approaches include

- addition of proper artificial dissipation;
- low-pass filtering of computed solution at each time step, to prevent the development of high-frequency instabilities [41];
- enforcement of secondary conservation properties (e.g., kinetic energy).

Both the first two methods introduce a certain amount of numerical dissipation, that has to be avoided in numerical simulation of turbulent flows. On the other hand, the enforcement of secondary conservation properties, e.g., the construction of discretely energy-conserving numerical methods, is a cutting-edge research theme that will be discussed in detail in the following section.

In contrast to space discretization algorithms, time-advancement schemes are usually regarded to be more important for the overall efficiency of the LES model than for the accuracy, provided that at least second-order methods are chosen [31]. The choice of the time step is also usually limited by the turbulent time scale [42], and therefore explicit time-stepping algorithms are often adopted.

1.1.3 Discrete conservation principles

In Section 1.1.2, it has been outlined how numerical methods for DNS and LES have to be nondissipative, accurate and stable at the same time. A possible solution to these counteracting requirements lies in the employment of nondissipative discretizations able to mimic, on a discrete level, the conservation of important invariants of the continuous equations [43]. These so-called *physics-compatible* discretizations have gained increasing popularity over the years, and have found an ideal context of application in numerical simulation of turbulent flows. In particular, discrete conservation of kinetic energy in the inviscid limit is highly desirable in order to obtain physically relevant solutions [44]. Energy-conserving methods are capable of enforcing a non-linear stability bound to the discrete solution, allowing for stable long-time integration. Moreover, being free of numerical diffusion, they ensure that the energy cascade is not artificially contaminated, in fully-resolved computations, and that the contribution of subgrid-scale motions is entirely modeled, in under-resolved cases [39]. While in incompressible flows the global kinetic energy is an invariant of the Navier-Stokes equations, for compressible flows there can be an isentropic exchange between kinetic and internal energy. However, in both cases the discretization of the convective term must not spuriously contribute to the discrete kinetic energy balance. An illustrative example of the importance of energy conservation is given in Fig. 1.2.

In the context of incompressible flow, the first energy-conserving finite-difference method dates back to 1965, when Harlow & Welch developed a second-order scheme in a staggered arrangement that discretely conserves mass, momentum and kinetic

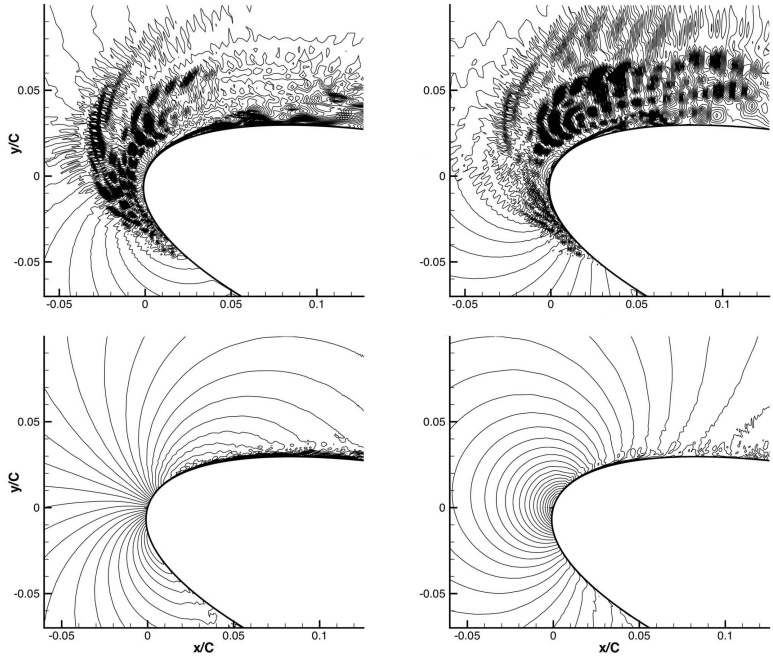


Figure 1.2: Contour plots of the instantaneous velocity components u (left) and v (right) over an airfoil at angle of attack of 16.6. The top figures are relative to a non-dissipative scheme without discrete kinetic energy conservation; the bottom figures to a scheme which conserves discrete energy. From [45].

energy on uniform meshes [46]. The scheme has been later extended to fourth-order by Morinishi [47] and to unstructured grids by Mahesh *et al*. [48]. An attempt to extend the fully-conservative, high-order scheme of Morinishi was made by Vasilyev, although simultaneous conservation of momentum and kinetic energy could not be achieved [49]. From a mathematical point of view, conservation of kinetic energy in the inviscid limit can be shown to be strongly related to the preservation of some of the fundamental symmetries of the continuous differential operators on the discrete level. This fact was exploited by Verstappen & Veldman to obtain a fourth-order finite-volume scheme that is fully conservative even on non-uniform meshes [50].

The formulation used for the non-linear convective term is also crucial in terms of energy-conservation properties. The so-called skew-symmetric (or splitting) form, defined as a proper average of divergence and advective forms, has been shown to guarantee *a priori* semi-discrete conservation of energy for several high-order centered schemes, over both regular and staggered grid systems [47]. The skew-symmetric form has also been shown to cause a reduction of aliasing errors with respect to the other formulations [51; 52]. In this regard, spectral simulations of turbulent flows with divergence or advective forms proved to be unstable without dealiasing, while the rotational form gave inaccurate results [53]. For these reasons, the splitting form of the nonlinear convective term has been successfully adopted to yield stable simulations of turbulent flows in the incompressible framework [54; 55]. In the context of compressible flow, a variable-density version of the skew-symmetric form was first introduced in [56], and has recently been discussed and extended in [57]. The underlying philosophy has been used to obtain robust methods in both finite-difference [58; 59] and finite-volume formulations [60; 61]. The splitting form is nowadays regularly applied in several highly accurate finite-difference and spectral numerical codes for DNS and LES of turbulent flows [62–64].

However, the major drawback of skew-symmetric-based methods is the increased computational cost with respect to classical divergence or advective forms [65], since evaluation of two derivatives (for incompressible flow) is required for each nonlinear component of the convective term [52]. Any attempt to reduce the number of derivative evaluations in computing the convective term requires the employment of divergence or advective forms, which are not guaranteed to preserve energy, at least on non-staggered or non-uniform grids. In this context, the demand for efficient methods naturally suggests the exploration of cheaper alternatives to the skew-symmetric splitting, which are able to retain its favourable conservation and aliasing properties.

In addition to the above-mentioned issues related to spatial discretization, time integration also plays a role. Typically, time advancement is performed in a semi-discretized approach, and the conservation properties of the temporal scheme are not usually investigated in detail. Full energy conservation (i.e., in space and time) can only be obtained by means of implicit methods [66], but the strong demand for efficient procedures usually dictates the use of explicit time-stepping.

In summary, the overall discrete energy conservation error typically results in the sum of a spatial and a temporal contribution: the former can be nullified by the use of

the spatially conservative methods (e.g., skew-symmetric or staggered methods), the latter by implicit time integration.

1.2 Objectives and outline of the work

The objective of the present work is to develop a time-advancement strategy that retains the conservation properties of skew-symmetric-based schemes at a reduced computational cost. This is achieved by properly constructed Runge-Kutta schemes in which a different form (advective or divergence) for the convective term is adopted at each stage. The final goal is to reproduce the effects of the skew-symmetric form on energy conservation up to a specific order of accuracy. The main advantage is that, since the method is based only on advective and divergence forms, it halves the cost of the computation of the non-linear term at the end of a time step.

The thesis is organized as follows. In Chapter 2, the conservation properties of a class of numerical methods used for the discretization of the non-linear convective term is reviewed and discussed, for both the Burgers' equation and the incompressible Navier-Stokes equations. Temporal energy errors for the class of Runge-Kutta schemes are also analyzed. The theoretical development of the novel *alternating* strategy is presented in Chapter 3, while the coefficients of the new Runge-Kutta schemes are derived in Chapter 4. In Chapter 5, several numerical tests are reported to demonstrate that the novel schemes are able to reproduce the energy-conservation properties of skew-symmetric methods. Concluding remarks are given in Chapter 6.

2. Energy-conservation properties of numerical methods

In this chapter, the energy-conservation properties of a class of numerical schemes used for the discretization of the non-linear term are analyzed and discussed. Semi-discretized algorithms are considered, i.e., methods in which the continuous equation is first discretized in space and then advanced in time.

Two physical models are considered here: the one-dimensional Burgers' equation and the incompressible Navier-Stokes equations. The former is a well-known prototype equation of considerable physical and mathematical interest [67]. It can be regarded as the simplest partial differential equation reproducing some of the peculiar features of non-linear convective transport terms present in more realistic models (e.g., the Navier-Stokes equations). When subject to stochastic forcing or random initial conditions, it also exhibits a chaotic behaviour that has been labeled as *Burgulence* (Burgers' turbulence) and is widely studied as a simplified model of real turbulence [68; 69]. Hence, the Burgers' equation represents an ideal framework to lay the foundations towards the application to more complex systems.

2.1 Spatial discretization

2.1.1 Burgers' equation

The simplest model to analyze energy-conservation properties of spatial discretizations of non-linear convective terms is the inviscid Burgers' equation, which can be formally written as

$$\partial_t u + \mathcal{N}(u) = 0, \tag{2.1}$$

where the non-linear convective term $\mathcal{N}(u)$ can be expressed in one of the continuously equivalent forms

$$(Div.) \equiv \partial_x u^2/2, \tag{2.2}$$

$$(Adv.) \equiv u\partial_x u, \tag{2.3}$$

$$(Skew.) \equiv (\partial_x u^2 + u\partial_x u)/3, \tag{2.4}$$

that are referred to as *divergence*, *advective* and *skew-symmetric*, respectively. The skew-symmetric form is a properly weighted average of the divergence and advective

forms. When posed on an interval $[a,b]$ with periodic boundary conditions, Eq. (2.1) has solutions for which all the moments are conserved:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_a^b \frac{u^n}{n}\,dx = \int_a^b u^{n-1}\frac{\partial u}{\partial t}\,dx = -\int_a^b u^n \frac{\partial u}{\partial x}\,dx = -\int_{u(a)}^{u(b)} u^n\,du = 0. \tag{2.5}$$

Specifically, the total momentum and the total energy of the solution (which are obtained in the cases $n = 1$ and $n = 2$, respectively) remain fixed to their initial values during time evolution.

Although in the continuous case the different forms of the convective term are mathematically equivalent, their spatially discretized counterparts can behave very differently, especially in terms of energy-preserving features and aliasing errors. This is due to the fact that discrete operators generally are not guaranteed to correctly reproduce the numerical equivalents of integration by parts and differentiation chain rule (cfr. [53]). The beneficial conservation and aliasing properties of the skew-symmetric form have long been recognized by many authors ([47; 52; 53]), while there is much more debate about the other two formulations. The present analysis examines the energy-conserving behavior of the spatially discretized version of Eq. (2.1) when the different forms are employed.

To pursue the scope, the semi-discretized version of the Burgers' equation is considered, which can be expressed as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{C}(\mathbf{u})\mathbf{u} = 0. \tag{2.6}$$

In Eq. (2.6), $\mathbf{u}$ is the vector of the nodal values of u, i.e. $u_i(t) = u(x_i,t)$. The linear convective operator $\mathbf{C}(\mathbf{u})$ has also been introduced. Depending on which among the three forms divergence, advective or skew-symmetric has been discretized, $\mathbf{C}(\mathbf{u})$ can be equal to one of the following operators

$$\mathbf{C}(\mathbf{u}) = \begin{cases} \mathcal{D} \equiv \mathbf{D}\mathbf{U}/2 & \textit{divergence,} \quad (2.7a) \\ \mathcal{A} \equiv \mathbf{U}\mathbf{D} & \textit{advective,} \quad (2.7b) \\ \mathcal{S} \equiv \dfrac{1}{3}(2\mathcal{D}+\mathcal{A}) & \textit{skew-symmetric,} \quad (2.7c) \end{cases}$$

where $\mathbf{U} = \mathrm{diag}(\mathbf{u})$ and $\mathbf{D}$ is the derivative matrix associated with the spatial discretization. Hereinafter, the analysis will be conducted by considering uniform meshes and periodic boundary conditions. For spectral methods and centered finite difference schemes, both explicit and compact, the derivative matrix turns out to be skew-symmetric:

$$\mathbf{D}^T = -\mathbf{D}. \tag{2.8}$$

The conservation properties of the discretized Eq. (2.6) can be inferred by considering the induced equations for the evolution of the discrete counterpart of integral, i.e. the scalar product. By premultiplying Eq. (2.6) by $\mathbf{1}^T$, where $\mathbf{1}$ is the column vector of all ones (discrete integrator on uniform mesh), it can be shown that, for skew-symmetric

derivative matrices, the total discrete momentum M is conserved, for both $\mathcal{A}$ and $\mathcal{D}$. From this result one easily concludes that every discretization employing a linear convex combination of the advective and divergence forms (e.g. the skew-symmetric form) conserves total momentum.

As regards global energy conservation, the relevant scalar product is $\mathbf{u}^T\mathbf{u} = \|\mathbf{u}\|^2$, for which the evolution equation reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{u}\|^2 = -2\mathbf{u}^T\mathbf{C}(\mathbf{u})\mathbf{u}. \tag{2.9}$$

Since the time derivative of the energy of any solution of the semi-discretized Eq. (2.6) can be expressed as a quadratic form associated to the matrix $\mathbf{C}$, a sufficient condition for the semi-discretized equation to be energy-conserving is that the matrix $\mathbf{C}$ is skew-symmetric for every $\mathbf{u}$. Similar considerations have been recently employed for the construction of difference operators which are optimized by forcing the reproduction of crucial symmetry properties of the underlying differential operator, in spite of minimizing local truncation error [50; 70].

From this analysis one immediately concludes that the semi-discretized Eq. (2.6) obtained by employing divergence or advective forms does not conserve global energy, since the associated matrices $\mathcal{D}$ and $\mathcal{A}$ are in general not skew-symmetric. The averaged skew-symmetric form, on the other hand, is naturally energy-conserving in cases in which the derivative matrix $\mathbf{D}$ is skew-symmetric. This property is also readily seen to be equivalent to the numerical analogue of integration by parts.

The analysis presented above can be completed by characterizing the different errors on energy conservation introduced by the divergence and advective forms. By evaluating Eq. (2.9) in the two cases one obtains:

$$\frac{\mathrm{d}E_{\mathrm{adv}}}{\mathrm{d}t} = -\mathbf{u}^T\mathbf{U}\mathbf{D}\mathbf{u} \tag{2.10}$$

$$\frac{\mathrm{d}E_{\mathrm{div}}}{\mathrm{d}t} = -\frac{1}{2}\mathbf{u}^T\mathbf{D}\mathbf{U}\mathbf{u} = \frac{1}{2}\mathbf{u}^T\mathbf{U}\mathbf{D}\mathbf{u} \tag{2.11}$$

where $E = \|\mathbf{u}^2\|/2$. From Eqs. (2.10) and (2.11) it follows that:

$$\frac{\mathrm{d}E_{\mathrm{div}}}{\mathrm{d}t} = -\frac{1}{2}\frac{\mathrm{d}E_{\mathrm{adv}}}{\mathrm{d}t}. \tag{2.12}$$

Equation (2.12) shows that the energy derivatives for the divergence and convective forms have opposite signs and that their relation is such that an average of the two forms with weights 2 and 1 (which defines the skew-symmetric form) furnishes exact conservation of energy:

$$\frac{\mathrm{d}E_{\mathrm{skw}}}{\mathrm{d}t} = \frac{2\frac{\mathrm{d}E_{\mathrm{adv}}}{\mathrm{d}t} + \frac{\mathrm{d}E_{\mathrm{div}}}{\mathrm{d}t}}{3} = 0. \tag{2.13}$$

Equations (2.12) (2.13) can be also rewritten in terms of the following relation between the divergence and advective discrete operators:

$$\mathcal{A}^T = -2\mathcal{D}, \tag{2.14}$$

that holds regardless of $\mathbf{u}$.

2.1.2 Navier-Stokes equations

The analysis of the previous section is here extended to the incompressible Navier-Stokes equations,

$$\frac{\partial u_i}{\partial t} + \mathcal{N}_i(u) = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}}\frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{2.15}$$

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{2.16}$$

where $\mathcal{N}_i(u)$ is the non-linear convective term and Re is the Reynolds number. The convective term can be cast in several analytically equivalent forms, for instance

$$(Div.)_i \equiv \frac{\partial u_j u_i}{\partial x_j}, \tag{2.17}$$

$$(Adv.)_i \equiv u_j \frac{\partial u_i}{\partial x_j}, \tag{2.18}$$

$$(Skew.)_i \equiv \frac{1}{2}\frac{\partial u_j u_i}{\partial x_j} + \frac{1}{2} u_j \frac{\partial u_i}{\partial x_j}. \tag{2.19}$$

These are referred to as *divergence*, *advective* and *skew-symmetric* forms, respectively, and are analogous to Eqs. (2.2–2.4). Other forms can also be constructed (e.g., rotational), but will not be considered in this study. When posed on a domain Ω with periodic boundary conditions, Eqs. (2.15–2.16) possess a number of invariants, most notably the momentum $m = \int_\Omega u_i dV$ and, for $\mathrm{Re} \to \infty$, the kinetic energy $e = \int_\Omega u_i^2/2\, dV$, which represent the *linear* and *quadratic* invariants, respectively. In a continuous setting, these properties hold for any expression adopted for the non-linear term. When discretized, Eqs. (2.17–2.19) have been shown to behave differently, both in terms of conservation properties [47] and aliasing issues [53].

A semi-discretized version of Eqs. (2.15–2.16) can be expressed as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} + \mathbf{C}(\mathbf{u})\mathbf{u} = -\mathbf{G}\mathbf{p} + \frac{1}{\mathrm{Re}}\mathbf{L}\mathbf{u}, \tag{2.20}$$

$$\mathbf{M}\mathbf{u} = \mathbf{0}, \tag{2.21}$$

where $\mathbf{u}$ is the discrete velocity vector containing the three components on the three-dimensional mesh, $\mathbf{u} = [\mathbf{u}_x\, \mathbf{u}_y\, \mathbf{u}_z]^T$, the matrices $\mathbf{G} \in R^{N_\mathbf{u} \times N_p}$ and $\mathbf{M} \in R^{N_p \times N_\mathbf{u}}$ are the discrete gradient and divergence operators, respectively, while $\mathbf{L} \in R^{N_\mathbf{u} \times N_\mathbf{u}}$ is the

block-diagonal Laplacian $\mathrm{diag}(\mathcal{L},\mathcal{L},\mathcal{L})$ with $\mathcal{L} \in R^{N_p \times N_p}$. In what follows, it will be assumed that the gradient, divergence and Laplacian operators are discretized consistently, in such a way that the relations $\mathbf{G}^T = -\mathbf{M}$ and $\mathcal{L} = \mathbf{MG}$ hold. The convective term is expressed as the product of a linear block-diagonal convective operator $\mathbf{C}(\mathbf{u})$ and $\mathbf{u}$:

$$\mathbf{C}(\mathbf{u})\mathbf{u} = \begin{bmatrix} \mathcal{C}(\mathbf{u}) & & \\ & \mathcal{C}(\mathbf{u}) & \\ & & \mathcal{C}(\mathbf{u}) \end{bmatrix} \begin{bmatrix} \mathbf{u}_x \\ \mathbf{u}_y \\ \mathbf{u}_z \end{bmatrix}. \tag{2.22}$$

By extending the one-dimensional case to three dimensions, the operator $\mathbf{C}(\mathbf{u})$ can assume one of the following forms:

$$\mathbf{C}(\mathbf{u}) = \begin{cases} \mathcal{D} \equiv \mathrm{diag}\left(\mathbf{D}_x\mathbf{U}_x + \mathbf{D}_y\mathbf{U}_y + \mathbf{D}_z\mathbf{U}_z\right) & \textit{divergence,} \quad (2.23a) \\ \mathcal{A} \equiv \mathrm{diag}\left(\mathbf{U}_x\mathbf{D}_x + \mathbf{U}_y\mathbf{D}_y + \mathbf{U}_z\mathbf{D}_z\right) & \textit{advective,} \quad (2.23b) \\ \mathcal{S} \equiv \dfrac{1}{2}\left(\mathcal{A} + \mathcal{D}\right) & \textit{skew-symmetric,} \quad (2.23c) \end{cases}$$

In Eq. (2.23), the matrices $\mathbf{D}_{(\cdot)}$ and $\mathbf{U}_{(\cdot)}$ represent the discrete derivative operators and the diagonal matrices of the discretized velocity components along the three directions (e.g., $\mathbf{U}_x = \mathrm{diag}(\mathbf{u}_x)$).

A *regular* arrangement of the variables on the grid is employed [47], for which the velocity components u_i and pressure p are stored at the same points. Centered finite-difference and spectral discretizations on a Cartesian, periodic and equally spaced grid will be considered here. Under such hypotheses, the derivative operators satisfy a discrete summation-by-parts rule; from an algebraic point of view, this implies that the derivative matrices are all skew-symmetric, $\mathbf{D}^T = -\mathbf{D}$. As a consequence, it can easily be shown that the convective operator $\mathcal{S}$ also inherits this property for all $\mathbf{u}$, unlike the other two forms.

The discrete conservation properties can be analyzed by deriving the evolution equation of the discrete kinetic energy $E = \mathbf{u}^T\mathbf{u}/2$, which reads

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -\mathbf{u}^T\mathbf{C}(\mathbf{u})\mathbf{u} - \mathbf{u}^T\mathbf{G}\mathbf{p} + \frac{1}{\mathrm{Re}}\mathbf{u}^T\mathbf{L}\mathbf{u}. \tag{2.24}$$

In Eq. (2.24), the pressure term is conservative if $\mathbf{G}^T = -\mathbf{M}$ and $\mathbf{Mu} = 0$. The convective and diffusive terms appear as quadratic forms with associated matrices $\mathbf{C}(\mathbf{u})$ and $\mathbf{L}$, respectively. The diffusive term is clearly not conservative (if properly discretized, the operator $\mathbf{L}$ is a negative-definite matrix), while the convective term is conservative if the skew-symmetric operator of Eq. (2.23c) is adopted. In the other two cases, the convection matrices $\mathbf{C}(\mathbf{u})$ are in general not skew-symmetric and hence conservation of energy is not guaranteed.

An analysis similar to the one carried out at the end of Section 2.1.1 reveals that errors coming from the divergence and advective forms have opposite signs. The key observation is that, for skew-symmetric discrete derivative operators, the following relation holds

$$\mathcal{A}^T = -\mathcal{D}, \tag{2.25}$$

which immediately implies $-\mathbf{u}^T\mathcal{A}\mathbf{u} = \mathbf{u}^T\mathcal{D}\mathbf{u}$ and hence that the energy variation due to the divergence and advective forms are equal and of opposite sign.

2.2 Time-advancement

In contrast to spatial discretizations, energy conservation properties of time-integration algorithms are much less discussed in literature. It is generally believed that the errors due to the spatial discretization are much larger than those coming from time-advancement, especially in numerical simulation of turbulent flows. In such cases, the time-step is usually dictated by accuracy and not by stability restrictions [42], leading to particularly small energy errors. Common choices for time-integration methods are the multi-step Adams-Bashforth or the multi-stage Runge-Kutta schemes. In the present work, the attention will be focused on the latter.

Application of a generic Runge-Kutta scheme to a convective term $\mathbf{C}(\mathbf{u})\mathbf{u}$ reads

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \sum_{i=1}^{s} b_i \widetilde{\mathbf{C}}(\mathbf{u}_i)\mathbf{u}_i, \tag{2.26}$$

$$\mathbf{u}_i = \mathbf{u}^n - \Delta t \sum_{j=1}^{s} a_{ij} \widetilde{\mathbf{C}}(\mathbf{u}_j)\mathbf{u}_j, \tag{2.27}$$

where s is the number of stages and a_{ij} and b_i are the coefficients of the scheme. Equations (2.26)-(2.27) apply to either the one-dimensional convective operator (2.7) or the Navier-Stokes operator (2.23). However, the tilde $\widetilde{\cdot}$ indicates that the discrete convection matrix $\widetilde{\mathbf{C}}$ is projected by means of an operator $\mathbf{P}$:

$$\widetilde{\mathbf{C}} = \mathbf{P}\mathbf{C}. \tag{2.28}$$

The projection operator $\mathbf{P}$ can assume different values depending whether the Burgers' equation or the Navier-Stokes equations are under consideration:

$$\mathbf{P} = \begin{cases} \mathbf{I} & \text{Burgers' equation,} \quad (2.29a) \\ \mathbf{I} - \mathbf{G}\mathcal{L}^{-1}\mathbf{M} & \text{Navier-Stokes equations,} \quad (2.29b) \end{cases}$$

where $\mathbf{I}$ is an identity matrix of proper dimensions. In the case of the Navier-Stokes equations the convective operator has to be projected to yield a divergence-free velocity field. This is usually accomplished by means of fractional-step methods [71; 72]. Similarly, $\widetilde{\mathcal{D}}$, $\widetilde{\mathcal{A}}$ and $\widetilde{\mathcal{S}}$ are the projected divergence, advective and skew-symmetric operators. It is worth to note that the projection operator has a number of interesting properties, which are valid for both Burgers' and Navier-Stokes equations:

$$\mathbf{P}^T = \mathbf{P}, \tag{2.30}$$

$$\mathbf{P}^k = \mathbf{P}, \tag{2.31}$$

$$\mathbf{P}\mathbf{u} = \mathbf{u},\ \mathbf{u}^T\mathbf{P} = \mathbf{u}^T \qquad \forall \mathbf{u} \,|\, \mathbf{M}\mathbf{u} = 0, \tag{2.32}$$

i.e., the projection operator is symmetric, idempotent and leaves unaffected divergence-free velocity fields (this is true for all vectors **u** in the case of the Burgers' equation). For the Navier-Stokes equations, the operator is also singular.

The coefficients a_{ij} and b_i are often arranged into the so-called *Butcher array* [73]:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \dots & \dots & a_{ss} \\ \hline & b_1 & \dots & \dots & b_s \end{array} ,$$

where $c_i = \sum_j a_{ij}$. Since the semi-discretized equations under study constitute a system of *autonomous* ordinary differential equations, the c_i coefficients will not be displayed hereinafter. Nonetheless, these coefficients might still come into play whenever time-dependent source terms or boundary conditions are considered. The coefficients a_{ij} and b_i are usually determined in order to maximize the formal order of accuracy of the solution. The order conditions are a set of non-linear constraints on the coefficients. Their number increases as the desired order of time-accuracy on the computed solution increases. The relations needed to obtain up to fourth-order accuracy are reported in Table 2.1. In particular, Eq. (2.33) is the first-order relation, Eq. (2.34) is needed for second-order accuracy, Eqs. (2.35–2.36) have to be added to achieve third order and finally the four Eqs. (2.37–2.40) are the fourth-order conditions. For Runge-Kutta schemes up to four stages s, one has $p \leq s$, where p is the order of accuracy. Notable examples of Runge-Kutta schemes are summarized in Table 2.2.

Order	Relation	
1	$\sum_i b_i = 1$	(2.33)
2	$\sum_{ij} b_i a_{ij} = 1/2$	(2.34)
3	$\sum_{ijk} b_i a_{ij} a_{ik} = 1/3$	(2.35)
	$\sum_{ijk} b_i a_{ij} a_{jk} = 1/6$	(2.36)
4	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{il} = 1/4$	(2.37)
	$\sum_{ijkl} b_i a_{ij} a_{ik} a_{kl} = 1/8$	(2.38)
	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{jl} = 1/12$	(2.39)
	$\sum_{ijkl} b_i a_{ij} a_{jk} a_{kl} = 1/24$	(2.40)

Table 2.1: Conditions for order of accuracy up to four for Runge-Kutta schemes

Two-stage, second-order		
0		
$\frac{1}{2\theta}$	0	
$1-\theta$	θ	

Kutta scheme		
0		
$\frac{1}{2}$	0	
-1	2	0
$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

RK4			
0			
$\frac{1}{2}$	0		
0	$\frac{1}{2}$	0	
0	0	1	0
$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Table 2.2: Some common Runge-Kutta schemes.

The discrete energy balance for the entire space-time algorithm can eventually be derived. Eqs. (2.26–2.27) can be manipulated to give an expression for the evolution of energy in a single time step of the Runge-Kutta procedure. By defining $\Delta E = E^{n+1} - E^n$, Eqs. (2.26–2.27) easily lead to the relation [66]:

$$\frac{\Delta E}{\Delta t} = \underbrace{-\sum_{i=1}^{s} b_i \mathbf{u}_i^T \widetilde{\mathbf{C}}(\mathbf{u}_i)\,\mathbf{u}_i}_{I} \underbrace{-\frac{\Delta t}{2}\sum_{i,j=1}^{s}\left(b_i a_{ij} + b_j a_{ji} - b_i b_j\right)\mathbf{u}_i^T \widetilde{\mathbf{C}}^T(\mathbf{u}_i)\,\widetilde{\mathbf{C}}(\mathbf{u}_j)\,\mathbf{u}_j}_{II} \quad (2.41)$$

The two terms on the right-hand side of Eq. (2.41) can be defined as

I) *Spatial* error: $\sum_{i=1}^{s} b_i \mathbf{u}_i^T \widetilde{\mathbf{C}}(\mathbf{u}_i)\,\mathbf{u}_i$

II) *Temporal* error: $\frac{\Delta t}{2}\sum_{i,j=1}^{s}\left(b_i a_{ij} + b_j a_{ji} - b_i b_j\right)\mathbf{u}_i^T \widetilde{\mathbf{C}}^T(\mathbf{u}_i)\,\widetilde{\mathbf{C}}(\mathbf{u}_j)\,\mathbf{u}_j$

The first one is composed by a linear combination of s different terms, each having the usual structure of a quadratic form for the convection matrix $\widetilde{\mathbf{C}}$. Each term is identically zero if a skew-symmetric form is adopted for $\widetilde{\mathbf{C}}$, hence the name *spatial* error appears to be appropriate.

The second quantity causing a variation of energy has a more complex structure and does not vanish in general, even in the case of skew-symmetric operators $\widetilde{\mathbf{C}}$. It usually inherits the same order of the scheme and can be nullified only for suitably chosen Runge-Kutta integrators, and this justifies the term *temporal* for this part of the error. It is well known that for the temporal error to be zero one has to employ so-called *symplectic* methods, for which $b_i a_{ij} + b_j a_{ji} - b_i b_j = 0$, a constraint which cannot be satisfied by explicit methods [74]. It is worth noting that the presence of the projection operator does not influence the discrete energy, as long as $\mathbf{M}\mathbf{u}_i = 0$ [66].

2.3 Computational cost

The above considerations can be summarized as follows. If one employes a spatial discretization in which the skew-symmetric form for the convective operator is adopted, the spatial error on energy conservation is absent. The temporal error due to Runge-Kutta integration is however still present and it can be shown to vanish, as Δt is reduced, at least with the same order of accuracy p of the Runge-Kutta scheme employed. If one wants to completely nullify the temporal error, implicit (symplectic) methods have to be chosen. However, these can not be used for many cases of interest due to the unaffordable computational effort that would require. The employment of a non energy-conserving spatial discretization (e.g. in divergence or advective forms), on the other hand, produces a zeroth-order error on energy conservation (due to the spatial part of the error) independently of the accuracy of the Runge-Kutta procedure. This situation cannot be alleviated by the time integration procedure, since the error is completely due to the spatial discretization and would still be present in an exact integration of the system of ODEs.

The number of floating point operations required to perform a complete time-advancement step depends heavily on the form adopted for the non-linear term. The practical implementation of the skew-symmetric form in a finite-difference code is easily shown to be roughly twice as expensive as standard divergence or advective forms alone. In the framework of incompressible Navier-Stokes equations, for instance, the splitting form requires 18 derivatives evaluations, while the advective or divergence forms take 9 derivatives [7]. In 1D, both numbers are reduced by a factor of 9. In spectral algorithms, the computation of the convective term in skew-symmetric form requires twice the number of discrete Fourier transforms (forward and backward) with respect to the other formulations. Although the other modules of the overall solution algorithm can take a certain part of the total CPU time (e.g., solution of pressure equation, computation of viscous terms, etc.), the net cost increase due to the use of the skew-symmetric splitting can be noteworthy, especially for explicit time-marching algorithms or in spectral codes. In the latter case, the Poisson equation for pressure is practically costless, while the viscous term is often treated analytically. By using the cost metric presented in Appendix A, it can be shown that in order to advance in time the convective term with a standard RK4 scheme and a second-order centred scheme, 52 operations per node are required for the skew-symmetric form, while 28 operations are needed for divergence or advective forms. The difference increases as the spatial order of accuracy is increased. The objective of the present work is to investigate time-advancing strategies that are able to retain the beneficial properties of the skew-symmetric form at a reduced computational cost.

3. The alternating RK strategy

The rationale underlying this method stems from the fact that the global errors on energy conservation associated with discretized divergence and advective forms have opposite signs. The basic idea is to take advantage of the time-advancement scheme to cancel the errors of these two forms up to a certain order of accuracy. By using only divergence or advective forms, the resulting scheme can be more cost-effective than a skew-symmetric-based one. In the following analysis, only inviscid flow is considered ($\mathrm{Re} \to \infty$), since interest is focused exclusively on the conservation properties of the convective term.

3.1 Core of the method

The core of the method consists in advancing the non-linear term by means of a modified, explicit Runge-Kutta algorithm, which can be expressed as

$$\mathbf{u}^{n+1} = \mathbf{u}^n - \Delta t \sum_{i=1}^{s} b_i \widetilde{\mathbf{C}}_i(\mathbf{u}_i)\mathbf{u}_i \tag{3.1}$$

$$\mathbf{u}_i = \mathbf{u}^n - \Delta t \sum_{j=1}^{i-1} a_{ij} \widetilde{\mathbf{C}}_j(\mathbf{u}_j)\mathbf{u}_j \tag{3.2}$$

where s is the number of stages, a_{ij} and b_i are the Runge-Kutta coefficients. The novel procedure is valid for projected, quadratic non-linear operators of the type of Eq. 2.28. It can hence be applied to both the convective operator of the Burgers' equation (2.7) and that of the Navier-Stokes equations (2.23). Equations (3.1-3.2) differ from the standard Runge-Kutta procedure (2.26)-(2.27) in that they can accommodate a different formulation for the non-linear term within the stages. In fact, the operator $\widetilde{\mathbf{C}}$ is indexed by a suffix, meaning that it can be expressed in either divergence or advective form at each stage.

The overall aim of this method is to find a set of coefficients, along with a sequence of divergence and advective forms, that maximizes the formal order of accuracy on both the discrete solution $\mathbf{u}$ and the discrete energy conservation. A prescribed order of accuracy for the discrete solution $\mathbf{u}$ can be attained by imposing a proper set of order conditions (cfr. Table 2.1). Hereinafter, the term *order conditions* will be used to refer to accuracy on the discrete solution. The resulting nonlinear system is usually underdetermined: the remaining degrees of freedom can be exploited to impose an additional set of equations for energy conservation.

The order conditions will be derived in the next subsection. Then, the derivation of the energy-conservation equations, which is the central part of the novel procedure, will be described in detail.

3.1.1 Order conditions

In this section, the order conditions for the modified Runge-Kutta schemes presented in Eqs. (3.1–3.2) are derived. It will be shown that the use of different operators at each sub-stage does not alter the convergence towards the exact solution of the partial differential equation. Indeed, since the discrete operators $\mathbf{C}_i(\mathbf{u}_i)\mathbf{u}_i$ tend all to the continuous term $\mathcal{N}(u)$ as $\Delta x \to 0$, the convergence properties of the proposed procedure turns out to have the same characteristics of those of classical semi-discretized algorithms.

For the sake of simplicity, a three-stage, third-order case will be taken into consideration; however, the analysis can easily be extended to a more general case. By following the standard path, the right-hand side of Eq. (3.1) is expanded in Taylor series to yield:

$$\begin{aligned}\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i \mathbf{f}_i + \Delta t^2 \left[\sum_{i=1}^{s} \left(b_i \dot{\mathbf{f}}_i \sum_{j=1}^{s} a_{ij} \mathbf{f}_j \right) \right] + \\ + \Delta t^3 \left[\sum_{i=1}^{s} b_i \dot{\mathbf{f}}_i \left(\sum_{j=1}^{s} a_{ij} \dot{\mathbf{f}}_j \sum_{k=1}^{s} a_{jk} \mathbf{f}_k \right) + \frac{1}{2} \sum_{i=1}^{s} b_i \ddot{\mathbf{f}}_i \left(\sum_{j=1}^{s} a_{ij} \mathbf{f}_j \right)^2 \right] + \mathcal{O}(\Delta t^4)\end{aligned} \tag{3.3}$$

where dots denote differentiation and $\mathbf{f}_i = \mathbf{C}_i(\mathbf{u}^n)\mathbf{u}^n$. In Eq. (3.3), the powers and products between vectors have to be intended as element-wise operations. The various terms $\mathbf{f}_i$ and $\dot{\mathbf{f}}_i$ can be collected out by invoking the accuracy of the spatial discretization. In fact, assuming that both the discrete divergence and advective operators have been constructed by employing a derivative matrix of order of accuracy p, the difference between them is an order-p infinitesimal function, as the spatial mesh size $\Delta x \to 0$. In practice, one has:

$$\mathbf{f}_i = \mathbf{f}_j + O(\Delta x^p) = \mathbf{f} + \mathcal{O}(\Delta x^p), \tag{3.4}$$

where $\mathbf{f}$ represents the evaluation of the exact nonlinear convective term on the spatial mesh; an analogue relation holds for $\dot{\mathbf{f}}_i$.

By using Eq. (3.4), Eq. (3.3) can be rewritten as:

$$\begin{aligned}\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{f} \sum_{i=1}^{s} b_i + \Delta t^2 \mathbf{f}\dot{\mathbf{f}} \left[\sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} \right] + \\ + \Delta t^3 \left[\dot{\mathbf{f}}^2 \mathbf{f} \sum_{i=1}^{s} b_i \sum_{j=1}^{s} a_{ij} \sum_{k=1}^{s} a_{jk} + \frac{1}{2} \ddot{\mathbf{f}} \mathbf{f}^2 \sum_{i=1}^{s} b_i \left(\sum_{j=1}^{s} a_{ij} \right)^2 \right] + \\ + e_1 + e_2 + e_3 + e_{\Delta t},\end{aligned} \tag{3.5}$$

where $e_k = \Delta t^k \mathcal{O}(\Delta x^p)$ and $e_{\Delta t} = \mathcal{O}(\Delta t^4)$.

The Taylor expansion of the exact solution reads:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \mathbf{f} + \frac{1}{2}\Delta t^2 \mathbf{f}\dot{\mathbf{f}} + \frac{1}{6}\Delta t^3 \left[\dot{\mathbf{f}}^2 \mathbf{f} + \ddot{\mathbf{f}}\mathbf{f}^2\right] + e_{\Delta t}. \tag{3.6}$$

A term-by-term comparison between Eq. (3.5) and Eq. (3.6) shows that the order conditions that have to be enforced on the coefficients a_{ij} and b_j to satisfy third-order accuracy coincide with the classical conditions (2.33)-(2.36):

$$\sum_i b_i = 1, \tag{3.7}$$

$$\sum_i b_i \sum_j a_{ij} = \frac{1}{2}, \tag{3.8}$$

$$\sum_i b_i \left(\sum_j a_{ij}\right)^2 = \frac{1}{3}, \tag{3.9}$$

$$\sum_i b_i \sum_j a_{ij} \sum_k a_{jk} = \frac{1}{6}. \tag{3.10}$$

However, in contrast to the standard procedure, the difference between computed and exact solutions is composed by additional errors, besides the third-order term $e_{\Delta t}$. The other terms are constituted by the product of two infinitesimal functions of time- and space-step sizes. The situation illustrated above can be easily generalized to order r Runge-Kutta schemes, for which the satisfaction of order conditions will produce an error made up of the term $e_{\Delta t} = \mathcal{O}(\Delta t^{r+1})$ as well as of a sum of additional terms e_k, with $k = 1, 2, \ldots, r$.

A similar situation is also present in the classical case in which the same form is adopted at each stage of the Runge-Kutta method. Although a convergence study of the procedure towards the semi-discretized system of ordinary differential equations possible in that case, the analysis of the convergence of the discretization with respect to the original partial differential equation conduct to an equation similar to Eq. (3.5). In this case, the e_i terms stem from the substitution of the spatially discretized right hand side of the system of ODE with the evaluation of the continuous nonlinear term on the spatial mesh.

The behavior of the error components e_i (whose leading term is e_1) can be analyzed by considering different convergence processes, involving time, space or both time and space refinements. As an example, if one considers a limit process in which both space and time increments are reduced in such a way that the ratio $\lambda = \Delta t/\Delta x$ is taken fixed, the generic error e_k asymptotically behaves as $K\Delta t^{p+k}/\lambda^p$, which means that for this limit process the temporal order of accuracy is governed by both spatial and temporal discretization accuracy. The leading order term e_1 will scale at the same rate of $e_{\Delta t}$, as the time step is reduced, in the particular case in which $p = r$. Hence, the method will be still of order r (with respect to temporal accuracy) if the space discretization has order $p \geq r$.

A more interesting limit process is the one in which the space discretization step is taken fixed and time increments are reduced. In this case the procedure will be

formally not consistent, since asymptotically the global error, whose leading term is again proportional to e_1, will be a constant. However, since this constant can be made arbitrarily small by refining the spatial step size Δx, for practical purposes a sufficiently accurate space discretization will reduce the terms e_k below the quantity $e_{\Delta t}$ for the time steps of interest. Put in another way, given any region of variation of the time step Δt, there exists a sufficiently accurate space discretization for which the errors e_k assume a magnitude which is smaller than the temporal error $e_{\Delta t}$. For such space discretizations the modified Runge-Kutta procedure behaves as the classical procedure, for what concerns the temporal accuracy of the solution.

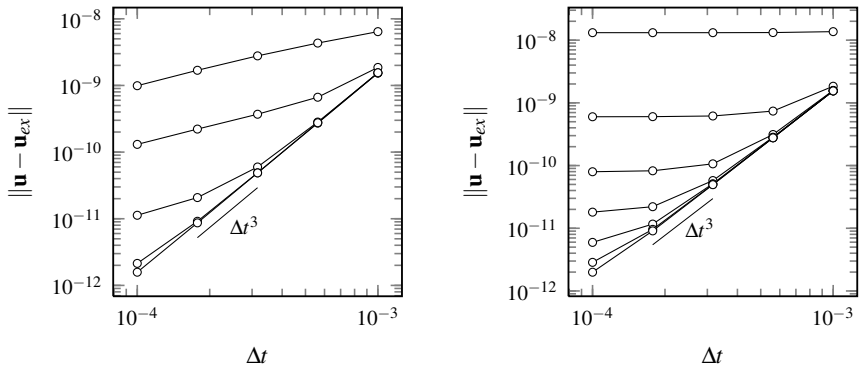


Figure 3.1: Convergence of the error on the solution. *(Left)* Sequence CDC, for a number of points N from 20 to 40. *(Right)* Sequence CCD, for a number of points N from 60 to 300. Additional details are available in the text.

Figure 3.1 confirms the analysis. Results are obtained by integrating the inviscid Burgers' equation with periodic boundary conditions on the domain $[-1,1]$, with initial condition $u_0(x) = \sin(\pi x)$. Several values of N (number of points) and Δt are examined, and numerical solution is compared to the exact solution by calculating an L_2 norm, $\|\mathbf{u}-\mathbf{u}_{ex}\|$. A sixth-order compact scheme is used for spatial integration [37], while for time-integration a 3-stage, third-order Runge-Kutta scheme is adopted. As the number of points is increased, the contribution of e_i is progressively diminished, until the correct third-order slope of the temporal method is correctly recovered. It is interesting to note that the sequence CDC converges much more quickly with respect to the sequence CCD. This behaviour is due to the inherent energy-conservation properties of the CDC sequence, that will be revealed in the subsequent sections.

3.2 Energy-conservation conditions

In this section, the energy-conservation conditions for the alternating Runge-Kutta methods are derived and discussed. This is the peculiar part of the novel procedure, as the derivation of such conditions is not trivial and requires particular care.

3.2.1 Energy analysis

The total discrete energy error introduced over a single time-step advancement by Eqs. (3.1–3.2) can be obtained by taking the inner product between $\mathbf{u}^{n+1}$ and itself. By defining $\Delta E = E^{n+1} - E^n$, Eqs. (3.1–3.2) can be manipulated to yield

$$\frac{\Delta E}{\Delta t} = \underbrace{-\sum_{i=1}^{s} b_i \, \mathbf{u}_i^T \widetilde{\mathbf{C}}_i(\mathbf{u}_i)\,\mathbf{u}_i}_{I} \underbrace{-\frac{\Delta t}{2}\sum_{i,j=1}^{s} g_{ij}\, \mathbf{u}_i^T \widetilde{\mathbf{C}}_i^T(\mathbf{u}_i)\,\widetilde{\mathbf{C}}_j(\mathbf{u}_j)\,\mathbf{u}_j}_{II}, \tag{3.11}$$

where $g_{ij} = b_i a_{ij} + b_j a_{ji} - b_i b_j$. Equation (3.11) is very similar to Eq. (2.41), but the introduction of alternating forms for the convetive term yields significant differences. Again, the two terms on the right-hand side of Eq. (3.11) can be defined as

I) *Spatial* error: $\Theta_s = -\sum_{i=1}^{s} b_i \mathbf{u}_i^T \widetilde{\mathbf{C}}_i(\mathbf{u}_i)\,\mathbf{u}_i$,

II) *Temporal* error: $\Theta_t = -\dfrac{\Delta t}{2}\sum_{i,j=1}^{s} g_{ij}\, \mathbf{u}_i^T \widetilde{\mathbf{C}}_i^T(\mathbf{u}_i)\,\widetilde{\mathbf{C}}_j(\mathbf{u}_j)\,\mathbf{u}_j$.

The spatial error vanishes for standard skew-symmetric methods ($\widetilde{\mathbf{C}}_i \equiv \widetilde{\mathbf{C}} = \widetilde{\mathbf{S}}$), while the temporal error can still be nullified by symplectic methods. However, the use of different discretized forms inside the stages of the Runge-Kutta procedure gives rise to new possibilities for obtaining cost-effective energy-preserving algorithms. An appropriate choice of Runge-Kutta coefficients can lead to methods in which the mixed spatial and temporal errors are nullified up to a certain order of accuracy.

The starting point of the analysis is the evaluation of the energy error as a Taylor series expansion in the time step increment Δt. This expression can be obtained by recursively substituting Eq. (3.2) into the right hand side of Eq. (3.2) itself and in Eq. (3.11). The key ingredient of the procedure will be, as shown below, the linearity of $\mathbf{C}_i(\mathbf{u}_i)$ as a function of $\mathbf{u}_i$. By expressing the terms $\mathbf{u}_j$ at the right hand side of Eq. (3.2) through Eq. (3.2) itself and by employing the exact relation

$$\widetilde{\mathbf{C}}_j(\mathbf{u}_j) = \widetilde{\mathbf{C}}_j(\mathbf{u}^n) - \Delta t \sum_k a_{jk} \widetilde{\mathbf{C}}_j\left(\widetilde{\mathbf{C}}_k(\mathbf{u}_k)\,\mathbf{u}_k\right), \tag{3.12}$$

one easily obtains

$$\mathbf{u}_i = \mathbf{u} - \Delta t\left(\sum_j a_{ij}\widetilde{\mathbf{C}}_j\mathbf{u}\right) + \Delta t^2\left[\sum_{j,k} a_{ij}a_{jk}\left(\widetilde{\mathbf{C}}_j\widetilde{\mathbf{C}}_k + \mathbf{C}_{jk}\right)\mathbf{u}\right] + \mathcal{O}\left(\Delta t^3\right), \tag{3.13}$$

where the summations are extended to the number of stages s. The following conventions have been adopted: $\mathbf{u} = \mathbf{u}^n$; $\widetilde{\mathbf{C}}_j = \widetilde{\mathbf{C}}_j(\mathbf{u}^n)$ and

$$\mathbf{C}_{jk} \equiv \widetilde{\mathbf{C}}_j\left(\widetilde{\mathbf{C}}_k(\mathbf{u}^n)\,\mathbf{u}^n\right). \tag{3.14}$$

In these last two relations, and in the following ones involving the matrices $\widetilde{\mathbf{C}}_i$, parentheses denote functional dependence. In all the other cases in which the functional

dependence of the matrices $\widetilde{\mathbf{C}}_j$ is not specified, it is assumed that the quantities are evaluated at $\mathbf{u}^n$.

Equation (3.13) can be employed to express $\widetilde{\mathbf{C}}_i(\mathbf{u}_i)$ as a Taylor series in Δt through Eq. (3.12):

$$\widetilde{\mathbf{C}}_i(\mathbf{u}_i) = \widetilde{\mathbf{C}}_i - \Delta t \sum_j a_{ij}\mathbf{C}_{ij} + \Delta t^2 \sum_{i,j} a_{ij}a_{jk}\left(\mathbf{C}_{i,jk} + \mathbf{C}_{ijk}\right) + \mathcal{O}\left(\Delta t^3\right), \tag{3.15}$$

where the further definitions have been introduced:

$$\mathbf{C}_{i,jk} \equiv \widetilde{\mathbf{C}}_i\left(\widetilde{\mathbf{C}}_j\widetilde{\mathbf{C}}_k\mathbf{u}\right), \tag{3.16}$$

$$\mathbf{C}_{ijk} \equiv \widetilde{\mathbf{C}}_i\left(\mathbf{C}_{jk}\mathbf{u}\right). \tag{3.17}$$

By substituting Eqs. (3.13–3.15) into Eq. (3.11) one obtains the expression for the energy variation ΔE as a Taylor series. The spatial and temporal parts of the error are reported in Eq. (3.19) and Eq. (3.20), respectively. The sum of the two contributions $\Theta_s + \Theta_t$ can be eventually rearranged to yield Eq. (3.21), where $g'_{ij} = b_i a_{ij} - b_j a_{ji} + b_i b_j$ and

$$\mathbf{F}^i_{jk} \equiv \widetilde{\mathbf{C}}_i^T\mathbf{C}_{jk} + \widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j\widetilde{\mathbf{C}}_k \tag{3.18}$$

have also been defined.

It is interesting to note that, unlike standard methods, in the novel procedure the temporal error Θ_t does not inherit the original order of the Runge-Kutta scheme. Instead, it has a non-vanishing term of $\mathcal{O}(\Delta t)$ that mixes with the spatial contribution.

Eq. (3.21) constitutes the basic relation on which the optimized Runge-Kutta schemes can be constructed. The procedure is based on the principle that suitable choices for the coefficients b_i and a_{ij} can nullify the successive terms in the series expansion of ΔE, thus identifying *optimal* schemes for what concerns conservation of energy.

3.2.2 Determination of energy-conservation equations

The determination of the coefficients is obtained by explicitly imposing the conditions that the various terms at the right hand side of Eq. (3.21) vanish, independently on $\mathbf{u}$. These constraints conduct to the determination of algebraic nonlinear equations involving the coefficients b_i and a_{ij} which can be coupled to classical order conditions to give a global system for the determination of new Runge-Kutta schemes. The number and the structure of the nonlinear equations related to energy conservation depend on several parameters, such as the number of stages, the chosen sequence of $\widetilde{\mathbf{C}}_i$'s and the details of the spatial discretization. In general, each of the operators involving the product or the composition of two or more matrices $\widetilde{\mathbf{C}}_i$ (i.e. each of the terms $\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j$, $\mathbf{C}_{ij}$, $\widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j$, ...) has to be considered as an independent function of the state vector $\mathbf{u}$. In these general cases the number of constraint to be imposed to the coefficients dramatically grows with both the number of stages and the order

$$\begin{aligned}\Theta_s = &-\sum_i b_i \mathbf{u}_i^T \widetilde{\mathbf{C}}_i(\mathbf{u}_i)\,\mathbf{u}_i = \\ &-\mathbf{u}^T\left(\sum_i b_i \widetilde{\mathbf{C}}_i\right)\mathbf{u} + \Delta t\,\mathbf{u}^T\left[\sum_{i,j} b_i a_{ij}\left(\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j + \mathbf{C}_{ij} + \widetilde{\mathbf{C}}_j^T\widetilde{\mathbf{C}}_i\right)\right]\mathbf{u} + \\ &-\Delta t^2\,\mathbf{u}^T\sum_{i,j,k} b_i a_{ij}\left[a_{jk}\left(\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j\widetilde{\mathbf{C}}_k + \widetilde{\mathbf{C}}_i\mathbf{C}_{jk} + \mathbf{C}_{i,jk} + \mathbf{C}_{ijk} + \mathbf{F}^i_{jk}\right) + a_{ik}\left(\mathbf{C}_{ij}\widetilde{\mathbf{C}}_k + \mathbf{F}^j_{ik}\right)\right]\mathbf{u} + O(\Delta t^3),\end{aligned} \tag{3.19}$$

$$\begin{aligned}\Theta_t = &-\frac{\Delta t}{2}\sum_{ij} g_{ij}\mathbf{u}_i^T\widetilde{\mathbf{C}}_i^T(\mathbf{u}_i)\,\widetilde{\mathbf{C}}_j(\mathbf{u}_j)\,\mathbf{u}_j = \\ &-\frac{\Delta t}{2}\mathbf{u}^T\left[\sum_{i,j} g_{ij}\widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j\right]\mathbf{u} + \frac{\Delta t^2}{2}\mathbf{u}^T\left[\sum_{i,j,k} g_{ij}a_{jk}\mathbf{F}^i_{jk} + g_{ij}a_{ik}\mathbf{F}^j_{ik}\right]\mathbf{u} + O\left(\Delta t^3\right),\end{aligned} \tag{3.20}$$

$$\begin{aligned}\frac{\Delta E}{\Delta t} = \Theta_t + \Theta_s = \\ &-\mathbf{u}^T\left[\sum_i b_i\widetilde{\mathbf{C}}_i\right]\mathbf{u} + \frac{\Delta t}{2}\mathbf{u}^T\left[\sum_{i,j} 2b_i a_{ij}\left(\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j + \mathbf{C}_{ij}\right) + g_{ij}\widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j\right]\mathbf{u} + \\ &-\frac{\Delta t^2}{2}\mathbf{u}^T\left\{\sum_{i,j,k} 2b_i a_{ij}\left[a_{jk}\left(\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j\widetilde{\mathbf{C}}_k + \widetilde{\mathbf{C}}_i\mathbf{C}_{jk} + \mathbf{C}_{i,jk} + \mathbf{C}_{ijk}\right) + a_{ik}\left(\mathbf{C}_{ij}\widetilde{\mathbf{C}}_k\right)\right] + g'_{ij}\left(\mathbf{F}^i_{jk}a_{jk} + \mathbf{F}^j_{ik}a_{ik}\right)\right\}\mathbf{u} + O\left(\Delta t^3\right).\end{aligned} \tag{3.21}$$

(a)	fix the number of stages of the method;
(b)	choose a sequence of advective and divergence forms;
(c)	group the terms of Eq. (3.22) and Eq. (3.23) into combinations of few independent terms;
(d)	impose conditions on a_{ij} and b_i to nullify the terms.

Table 3.1: Summary of steps needed to determine the energy-conservation conditions for alternating schemes

of accuracy, since the number of independent groups to be nullified grows linearly with the number of stages and more than linearly with the order of accuracy. In such situations the global system of equations quickly becomes overdetermined, especially in the more appealing case of explicit schemes. In what follows, it will be shown that the structure of Eq. (3.21) can be highly simplified to lead to a tractable set of conditions.

The analysis will be limited to the requirement of a maximum of second-order on energy conservation. Higher-order energy conservation is possible, but would require involved calculations which add little to the exposition of the main idea, and are possible subjects of future work. The two terms that have to be nullified are reported here again from Eq. (3.21) for convenience:

$$\Theta_1 = \mathbf{u}^T \left[\sum_i b_i \widetilde{\mathbf{C}}_i \right] \mathbf{u}, \tag{3.22}$$

$$\Theta_2 = \frac{\Delta t}{2} \mathbf{u}^T \left[\sum_{i,j} 2 b_i a_{ij} \left(\widetilde{\mathbf{C}}_i \widetilde{\mathbf{C}}_j + \mathbf{C}_{ij} \right) + g_{ij} \widetilde{\mathbf{C}}_i^T \widetilde{\mathbf{C}}_j \right] \mathbf{u}. \tag{3.23}$$

The practical determination of a suitable set of conditions is then obtained in the four steps depicted in Table 3.1. The steps (a) and (b) indicate that the resulting equations are clearly dependent upon the number of stages and the particular sequence employed. Therefore, different equations are expected for each sequence that can be possibly adopted for a fixed $s-$stage Runge-Kutta method. The core of the method lies in the treatment of step (c). The key observation that strongly simplifies the structure of the required constraints is that in several circumstances the *physically compatible* relation $\mathbf{D}^T = -\mathbf{D}$ for the discrete derivative operator can be assumed. In these cases the advective and divergence discrete forms are related by relations already derived in Section 2.1 and reported here for convenience:

$$\mathcal{A}^T = -2\mathcal{D} \tag{3.24}$$

for the 1D Burgers' equation and

$$\mathcal{A}^T = -\mathcal{D} \tag{3.25}$$

for the Navier-Stokes equations. Equations (3.24–3.25) do not hold for the corresponding projected forms. However, the following *equivalence operator* $\sim$ can be introduced. The operator applies to matrices and holds when the corresponding quadratic forms associated with a vector $\mathbf{u}$ are equal for any value of $\mathbf{u}$, e.g.

$$\mathbf{A} \sim \mathbf{B} \qquad \Longleftrightarrow \qquad \mathbf{u}^T\mathbf{A}\mathbf{u} = \mathbf{u}^T\mathbf{B}\mathbf{u} \qquad \forall\, \mathbf{u} \in R^{N_\mathbf{u}}.$$

By using this notation, the following relations can be derived:

$$\widetilde{\mathcal{A}} \sim -2\widetilde{\mathcal{D}} \tag{3.26}$$

for the Burgers' equation and

$$\widetilde{\mathcal{A}} \sim -\widetilde{\mathcal{D}} \tag{3.27}$$

for the Navier-Stokes equations; in the derivation, the properties of the projection operator (2.30)–(2.32) have also been used. Equations (3.26–3.27) will be hereinafter referred to as *transformation relations* since they allow to transform the various products or compositions of matrices $\widetilde{\mathbf{C}}_i$ into a set of independent functions of $\mathbf{u}$.

In the following, the first- and second-order terms will be analyzed individually to apply the transformation relations and obtain a set of constraints.

First-order term

The first-order term Θ_1 reported in Eq. (3.22) is a linear combination of convective operators. By using the transformation relations, a single term proportional to a quadratic form in which only one of the two operators $\widetilde{\mathcal{D}}$ or $\widetilde{\mathcal{A}}$ is present, can be collected out. Therefore, a final equation of the type

$$\sum_i \alpha_i b_i = 0 \tag{3.28}$$

is obtained for first-order conservation of energy, where the coefficients α_i depend on the sequence adopted and on the particular relation between the convective operators. The projection operator is found to be irrelevant in this term, provided that $\mathbf{M}\mathbf{u} = 0$.

Second-order term

The treatment of the second-order term Θ_2 needs some additional care. In fact, Eq. (3.23) shows that it is composed by the sum of several terms, each being proportional to a quadratic form involving one of the operators $\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j$, $\widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j$, $\mathbf{C}_{ij}$. The first two operators, constituted by the product of convective matrices, are equivalent in all cases to one of the operators $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}$,$\widetilde{\mathcal{D}}\widetilde{\mathcal{D}}$,$\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}$ multiplied by a scalar. In fact, one has $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}} \sim 4\widetilde{\mathcal{D}}\widetilde{\mathcal{D}}$ for the Burgers equation and $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}} \sim \widetilde{\mathcal{D}}\widetilde{\mathcal{D}}$ for the Navier-Stokes operators. As a consequence, the number of independent forms reduces to three, as each of the products between two convective matrices $\widetilde{\mathbf{C}}_i\widetilde{\mathbf{C}}_j$ or $\widetilde{\mathbf{C}}_i^T\widetilde{\mathbf{C}}_j$ is equivalent to one of the basic operators $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}$,$\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}$, multiplied by a proper scalar.

It can be also shown, although not completely evident at first, that also the composite terms $\mathbf{C}_{ij}$ are equivalent to one of the three forms $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}$,$\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}$. This fact can be shown by observing that for every vector $\mathbf{u}$ and for every matrix $\mathbf{B}$ the following relation holds:

$$\mathrm{diag}\left(\mathbf{Bu}\right)\mathbf{u} = \mathbf{UBu}, \tag{3.29}$$

where, as usual, $\mathbf{U} = \mathrm{diag}\left(\mathbf{u}\right)$. This relation can be employed in order to express the forms $\mathbf{C}_{ij}$ into one of the basic forms $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}$. As an example, for the terms $\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)$ and $\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)$ (parentheses denoting functional dependence) one has, for the Burgers' equation:

$$\mathbf{u}^T\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)\mathbf{u} = \tfrac{1}{2}\mathbf{u}^T\mathbf{PD}\mathrm{diag}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)\mathbf{u} \overset{(3.29)}{=} \tfrac{1}{2}\mathbf{u}^T\mathbf{PDU}\widetilde{\mathcal{A}}\mathbf{u} = \mathbf{u}^T\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}\mathbf{u}, \tag{3.30}$$

$$\begin{aligned}
\mathbf{u}^T\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)\mathbf{u} &= \mathbf{u}^T\mathbf{P}\mathrm{diag}\left(\widetilde{\mathcal{A}}\mathbf{u}\right)\mathbf{Du} = \\
&= \mathbf{u}^T\mathbf{P}\mathrm{diag}\left(\mathbf{PUDu}\right)\mathbf{Du} = \\
&\overset{(2.32)}{=} \mathbf{u}^T\mathrm{diag}\left(\mathbf{PUDu}\right)\mathbf{Du} = \\
&\overset{\text{transp.}}{=} -\mathbf{u}^T\mathbf{D}\mathrm{diag}\left(\mathbf{PUDu}\right)\mathbf{u} = \\
&\overset{(3.29)}{=} -\mathbf{u}^T\mathbf{DUPUDu} = \\
&\overset{(2.32)}{=} -\mathbf{u}^T\mathbf{PDUPUDu} = \\
&= -2\mathbf{u}^T\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}\mathbf{u}.
\end{aligned} \tag{3.31}$$

By acting in a similar way on the other possible terms $\mathbf{C}_{ij}$ one finally obtains the following equivalences:

$$\begin{aligned}
\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right) &\sim \widetilde{\mathcal{D}}\widetilde{\mathcal{A}}, \\
\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right) &\sim -2\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}, \\
\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{D}}\mathbf{u}\right) &\sim -\frac{1}{2}\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}, \\
\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{D}}\mathbf{u}\right) &\sim \widetilde{\mathcal{D}}\widetilde{\mathcal{D}} \sim \frac{1}{4}\widetilde{\mathcal{A}}\widetilde{\mathcal{A}},
\end{aligned} \tag{3.32}$$

for the Burgers' equation and

$$\begin{aligned}
\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right) &\sim -\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}, \\
\widetilde{\mathcal{A}}\left(\widetilde{\mathcal{D}}\mathbf{u}\right) &\sim -\widetilde{\mathcal{D}}\widetilde{\mathcal{D}}, \\
\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{A}}\mathbf{u}\right) &\sim \widetilde{\mathcal{D}}\widetilde{\mathcal{A}}, \\
\widetilde{\mathcal{D}}\left(\widetilde{\mathcal{D}}\mathbf{u}\right) &\sim \widetilde{\mathcal{D}}\widetilde{\mathcal{D}},
\end{aligned} \tag{3.33}$$

for the incompressible Navier-Stokes equations. These relations can be employed in the evaluation of the second-order term in Eq. (3.21), and allow to group the various

terms into three independent ones, associated with the forms $\widetilde{\mathcal{A}}\widetilde{\mathcal{A}}$, $\widetilde{\mathcal{A}}\widetilde{\mathcal{D}}$ and $\widetilde{\mathcal{D}}\widetilde{\mathcal{A}}$, independentely on the number of stages. Hence, the number of equations associated with the fulfillment of second-order accuracy on energy conservation is three. The form of these equations depends, of course, on the particular alternation sequence.

Summary

The application of step (d) of Table 3.1 leads to the conclusion that one additional linear constraint on the coefficients b_i has to be added to enforce first-order conservation of energy, whereas three nonlinear equations on the coefficients b_i and a_{ij} are needed for second-order conservation. The constraints on energy can be coupled to classical order conditions to give a global system for determining new Runge-Kutta schemes.

The energy-conservation conditions for both 1D and incompressible Navier-Stokes equations are reported in Appendix B.

4. Novel Runge-Kutta schemes

In this chapter, the energy-conservation relations derived in Chapter 3 will be coupled to classical order conditions to derive novel low-cost Runge-Kutta schemes with a prescribed order of accuracy on both solution and energy-conservation. The procedure will be carried out at first for the Burgers' equation and then for the incompressible Navier-Stokes equations.

A summary of the steps needed to derive the RK coefficients for the novel schemes is given in Table 4.1. Regarding step (a), two-, three- and four-stage schemes will be considered, as they have been applied historically to numerical simulations of turbulent flows and allow for a relatively large number of degrees of freedom. However, the analysis is general and can be in principle applied to methods of an arbitrary number of stages. An s-stage RK method gives $s(s+1)/2$ degrees of freedom. Second-, third- and fourth-order conditions on the solution take 2, 4 and 8 equations, respectively, whereas the first- and second-order conditions on energy conservation for alternating schemes take 1 and 3 equations. Depending on the number of available parameters provided by the method, order conditions and energy-preserving properties can be combined in many different ways, as indicated in step (b). The execution of steps (c), (d) and eventually (e) leads to several families of schemes. In order to cope with the large number of possible methods, a proper nomenclature has been established, which is reported in the following section.

(a)	fix the number of stages of the method;
(b)	choose a desired accuracy on solution and energy, compatibly with the number of degrees of freedom available;
(c)	choose a sequence of divergence and advective forms, among the possible sequences permitted by the stages of the method;
(d)	construct the non-linear system by using the relevant order conditions from Table 2.2 and the energy-conservation equations from Appendix B;
(e)	solve the system.

Table 4.1: Summary of steps needed to derive the Runge-Kutta coefficients.

4.1 Nomenclature

The novel Runge-Kutta schemes will be labeled by the following synthetic notation. Each method is first designated by a symbol that indicates whether it applies to 1D Burgers' equation (1D) or to the Navier-Stokes equations (NS). Then, an acronym reports the sequence of divergence (D) and advective (A) forms within the stages of the Runge-Kutta method for which the scheme has been built. Then, the theoretical orders of accuracy on solution (S) and on energy conservation (E) are specified. Finally, a cost coefficient proportional to the number of derivatives required per time-step is reported in brackets, to serve as a simple cost metric for comparing the performances of the various schemes. In general, the number of matrix-vector products is reasonably approximated by sd^2 for advective and divergence forms and $2sd^2$ for the skew-symmetric form, where d is the number of dimensions. For simplicity, $d = 1$ will be assumed hereinafter for the estimation of the cost-coefficients for both the Burgers' and Navier-Stokes schemes. In this simplified metric, the cost of the skew-symmetric form is doubled with respect to divergence and advective forms. The results of a more detailed cost analysis, reported in Appendix A, show that this is in many cases a very good approximation, at least for explicit finite-difference schemes. As an example, the label

$$1\mathrm{D}/\underbrace{\mathrm{ADDA}}_{\text{Sequence}}-\overbrace{3\mathrm{S}2\mathrm{E}}^{\text{Order}}\ \underbrace{(4)}_{\text{Cost}}$$

indicates a four-stage scheme for the 1D Burgers' equation which has to be used in conjunction with an ADDA sequence and is third-order accurate on solution and second-order accurate on energy conservation. When multiple methods are available, a subscript is used next to the sequence, e.g. $\mathrm{ADDA}_{1,2}$. It can also happen that a method applies to a sequence as well as to its dual counterpart (e.g., ADA and DAD). In this case, a circle arrow is reported next to the sequence, e.g. $\mathrm{ADA}_{\circlearrowleft}$.

4.2 1D schemes

In this section, alternating schemes for the Burgers' equation are derived and discussed. In order to keep the presentation as clear as possible, the description in this section will be particularly detailed, especially for what concerns the application of the theoretical developments described in Section 3.2.2. At the end of the section, the predicted order of accuracy will also be verified numerically.

4.2.1 Two-stage methods

The analysis is firstly conducted for the simple case of two-stage explicit Runge-Kutta methods. The free parameters for these schemes are b_1, b_2 and a_{12} and the maximum order conditions which can be satisfied are that of a second order scheme: $b_1 + b_2 = 1$

1D/AD–2S1E(2)		1D/DA–2S1E(2)	
0		0	
$\frac{3}{4}$	0	$\frac{3}{2}$	0
$\frac{1}{3}$	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{1}{3}$

Table 4.2: Two-stage optimized Runge-Kutta methods for Burgers' equation.

and $b_2 a_{21} = 1/2$. By imposing these two constraints one obtains the well known one-parameter family $b_1 = 1-\theta, b_2 = \theta$ and $a_{12} = 1/2\theta$, with $\theta \neq 0$. The free parameter θ can be fixed by requiring that also the first order condition on energy conservation is satisfied by b_1 and b_2, for each given alternation of forms, leading to

$$\mathbf{u}^T \left(b_1 \widetilde{\mathbf{C}}_1 + b_2 \widetilde{\mathbf{C}}_2 \right) \mathbf{u} = 0. \tag{4.1}$$

For a standard Runge-Kutta method in which $\widetilde{\mathbf{C}}_1 = \widetilde{\mathbf{C}}_2$, this term cannot vanish, except for the case in which the skew-symmetric form is employed, since the condition to be satisfied by b_1 and b_2 would be $b_1 + b_2 = 0$, which is not compatible with the first-order condition $b_1 + b_2 = 1$. In the case in which an alternation of forms is employed, one has different conditions on the b_i's, one for each alternation sequence. For instance, if the first stage is computed by adopting the advective form and the second with the divergence form (sequence AD) one has the condition

$$\mathbf{u}^T \left(b_1 \widetilde{\boldsymbol{\mathcal{A}}} + b_2 \widetilde{\boldsymbol{\mathcal{D}}} \right) \mathbf{u} = 0. \tag{4.2}$$

The transformation relation (3.26) allows to factor out one of the operators in Eq. (4.2), which is hence satisfied, independently on $\mathbf{u}$, if and only if $b_2 = 2b_1$. This last condition fixes the parameters of the Runge-Kutta scheme to the values $b_1 = 1/3, b_2 = 2/3$ and $a_{12} = 3/4$, which has second-order accuracy on the convergence of the solution and first-order accuracy on the conservation of energy. In the established notation, the derived scheme is referred to as 1D/AD–2S1E(2). The dual scheme 1D/DA–2S1E(2) can be easily obtained with the same procedure, which leads to $b_1 = 2b_2$. The coefficients of the schemes are summarized in Table 4.2.

4.2.2 Three-stage methods

Optimized three-stage methods can be derived by applying a similar procedure. The number of free parameters for the case of explicit three stages schemes rises to six $(s(s+1)/2)$, while the number of conditions to be satisfied for the case of a third-order scheme is four. Hence, the family of classical third-order, three-stage schemes

is at best parametrized by two constants, although two one-parameter families of third-order schemes also exist (cfr. Butcher [73]). The degrees of freedom constituted by the free parameters allows to derive either 3S1E(3)–type schemes, or 2S2E(3)–type methods.

3S1E(3) schemes

The number of possible sequences in which the same form is not repeated three times is six, and for each of these sequences a relation between the constants b_i has to be satisfied for first-order conservation of energy. This additional constraint can then be imposed to the coefficients satisfying third-order relations to obtain 3S1E(3) schemes, as it has been done for two-stage schemes. A complete picture of all the possible solutions would require considerable effort. Here, a treatment is presented in which, by renouncing to the complete analysis of all the possible solutions, the full system of order relations (on the solution and on the energy) is directly solved in the general case, without discussing the many exceptional cases. The treatment can be developed at once for all the possible sequences by observing that in all cases the equation related to first order conservation of energy

$$\mathbf{u}^T \left(b_1 \widetilde{\mathbf{C}}_1 + b_2 \widetilde{\mathbf{C}}_2 + b_3 \widetilde{\mathbf{C}}_3 \right) \mathbf{u} = 0, \tag{4.3}$$

leads to the relation

$$\alpha_1 b_1 + \alpha_2 b_2 + \alpha_3 b_3 = 0, \tag{4.4}$$

where $\alpha_i = 1$ or $\alpha_i = -1/2$ in the cases $\widetilde{\mathbf{C}}_i = \widetilde{\boldsymbol{\mathcal{A}}}$ or $\widetilde{\mathbf{C}}_i = \widetilde{\boldsymbol{\mathcal{D}}}$, respectively. For the determination of 3S1E(3) families of schemes, Eq. (4.4) has to be coupled to the classical relations for a third-order Runge-Kutta scheme, i.e., Eqs. (2.33)–(2.36). The general solution for this system can be obtained in two steps: first, derive b_1, b_2 and b_3 from the linear Eqs. (2.33) and (4.4) as a one-parameter family of coefficients. Second, solve for a_{21}, a_{31} and a_{32} from the nonlinear system (2.34–2.36) by considering b_2 and b_3 as parameters. The first step furnishes:

$$\begin{aligned} &b_1 = -\frac{\alpha_2}{\alpha_1 - \alpha_2} + \left(\frac{\alpha_2 - \alpha_3}{\alpha_1 - \alpha_2} \right) \theta; \quad b_2 = \frac{\alpha_1}{\alpha_1 - \alpha_2} - \left(\frac{\alpha_1 - \alpha_3}{\alpha_1 - \alpha_2} \right) \theta; \\ &b_3 = \theta, \end{aligned} \tag{4.5}$$

in the case $\alpha_1 \neq \alpha_2$ and

$$\begin{aligned} &b_1 = -\frac{\alpha_3}{\alpha_1 - \alpha_3} + \left(\frac{\alpha_3 - \alpha_2}{\alpha_1 - \alpha_3} \right) \theta; \quad b_3 = \frac{\alpha_1}{\alpha_1 - \alpha_3} - \left(\frac{\alpha_1 - \alpha_2}{\alpha_1 - \alpha_3} \right) \theta; \\ &b_2 = \theta, \end{aligned} \tag{4.6}$$

in the dual case $\alpha_1 \neq \alpha_3$. The second step furnishes:

$$a_{21} = \frac{1 - 2 b_3 c_\pm}{2 b_2}; \quad a_{32} = \frac{1}{6 b_3 a_{21}}; \quad a_{31} = c_\pm - a_{32}, \tag{4.7}$$

where

$$c_{\pm} = \frac{1}{2\,(b_2+b_3)} \left(1 \pm \sqrt{1 - \frac{(b_2+b_3)\,(3-4b_2)}{2b_3}} \right), \tag{4.8}$$

which is real for $(b_2+b_3)(3-4b_2)/2b_3 \leq 1$. Equations (4.7–4.8) furnish two families of one-parameter schemes for each sequence of alternating forms inside the three stages, provided that b_1, b_2 and $a_{12} \neq 0$ and that the coefficients b_2 and b_3 lie in the range of real values for $c_{\pm}$.

As an example, for the sequence $\widetilde{\mathbf{C}}_1 = \widetilde{\boldsymbol{\mathcal{A}}}$, $\widetilde{\mathbf{C}}_2 = \widetilde{\mathfrak{D}}$, $\widetilde{\mathbf{C}}_3 = \widetilde{\boldsymbol{\mathcal{A}}}$, (i.e. for the sequence ADA), first-order conservation of energy leads to the relation: $b_1 - b_2/2 + b_3 = 0$ which, coupled to Eq. (2.33) has the solution $b_1 = 1/3 - \theta, b_2 = 2/3$ and $b_3 = \theta$. In Fig. 4.1 the coefficients a_{21}, a_{31} and a_{32} of the two families associated with this sequence are reported. The dots on the curves relative to the first family of coefficients are the values of the coefficients of the classical Kutta scheme ($b_1 = b_3 = 1/6$, $b_2 = 2/3$, $a_{21} = 1/2$, $a_{31} = -1$, $a_{32} = 2$), which is a member of this class of schemes and hence has a first order accuracy on energy conservation when employed in conjunction with the ADA alternation of forms. In Fig. 4.1, the range of values of b_3 is such that $b_1 > 0$, as in Runge-Kutta methods the coefficients b_i are normally taken positive for stability reasons [73].

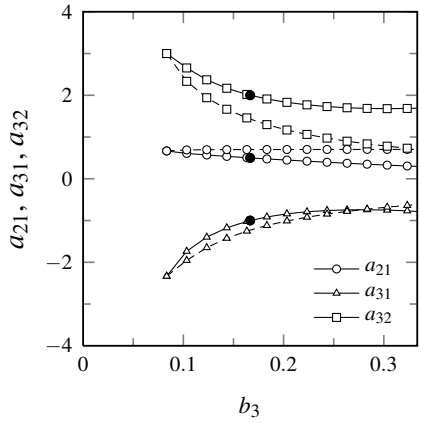


Figure 4.1: Coefficients a_{21}, a_{31} and a_{32} as functions of the parameter b_3 for the two families associated with the 3S1E(3)–type schemes for the sequence ADA. The black-filled circles represent the Kutta scheme.

2S2E(3) schemes

In this case, the non-linear system is constituted, for each sequence, by four equations for second-order conservation of energy (reported in Section B.1.1) and two relations to obtain second-order accuracy for the solution, Eqs. (2.33)–(2.34). The result is a system of six nonlinear equations in the six unknowns b_1, b_2, b_3, a_{21}, a_{31} and a_{32} for

$b_1 + b_2 + b_3 = 1$	1^{st} order on the solution
$b_2 a_{21} + b_3 (a_{31} + a_{32}) = \frac{1}{2}$	2^{nd} order on the solution
$b_1 - \frac{b_2}{2} + b_3 = 0$	1^{nd} order on en. cons.
$b_2^2 - b_3 a_{32} = 0$ $2 - b_3 (a_{32} - 2a_{31}) + b_2 b_3 + b_1 b_2 = 0$ $b_1^2 - 2b_2 a_{21} + 2b_3 a_{31} + 2b_1 b_3 + b_3^2 = 0$	2^{nd} order on en. cons.

Table 4.3: Nonlinear system for 1D/ADA–2S2E(3) schemes

1D/ADA–2S2E(3)

0		
$\frac{1}{3}$	0	0
$\frac{1}{6\theta}$	$\frac{1}{9\theta}$	0
$\frac{1}{3} - \theta$	$\frac{2}{3}$	θ

1D/DAD–2S2E(3)

0		
$\frac{1}{3}$	0	0
$\frac{1}{3\theta}$	$\frac{1}{18\theta}$	0
$\frac{2}{3} - \theta$	$\frac{1}{3}$	θ

Table 4.4: Three-stage, 2S2E(3)–type optimized Runge-Kutta methods for Burgers' equation. Both schemes are valid for $\theta \neq 0$.

the determination of explicit three stage 2S2E(3)–type Runge-Kutta schemes. Due to non-linearity, it is difficult to ascertain if this system has in general a solution and, in that case, if it is unique. Among all the possible sequences of forms for the case of explicit three stage methods, it is found that only in the case of the two sequences ADA and DAD a solution exists. In all the other cases the system has no solution. Moreover, the solution, when available, is not unique.

For ease of exposition, the complete nonlinear system for the sequence ADA is reported here in Table 4.3. Its solution is easily found by successive substitution, and the resulting one-parameter families of schemes for the ADA and DAD sequences are reported in Table 4.4. It can be readily seen that none of these schemes can satisfy the third-order relation Eqs. (2.35)–(2.36) for a particular value of θ, and hence three-stage 3S2E(3) schemes cannot be obtained.

4.2.3 Four-stage methods

Four-stage explicit Runge-Kutta methods are characterized by ten coefficients, which can be determined (at least as families of values) by imposing the required order of accuracy on the solution in the form of non-linear constraints. The standard procedure

is to obtain the families of coefficients by imposing the eight nonlinear conditions necessary to obtain 4^{th} order accuracy. The complete solution to this system is quite cumbersome to reproduce here, and is usually given as one- or two-parameter families of coefficients. An extensive treatment of this problem, together with the steps to be carried out in order to solve the nonlinear systems of conditions, can be found in the classical book by Butcher [73]. The undertaken approach will be again to complement the classical procedure by introducing the possibility of alternating the forms inside the various stages of the Runge-Kutta method and by including the nonlinear constraints on the coefficients associated to conservation of energy at a given order of accuracy. The complete discussion of all the particular cases which can arise in this procedure is out of the scope of the present work. The variety of nonlinear conditions and the number of particular cases exceptionally grows as one introduces all the possible sequences of alternating forms. In this paragraph, it will be simply illustrated how the general program can be studied, and some representative examples will be given. Two cases are investigated: 4S1E(4)–type and 3S2E(4)–type schemes.

4S1E(4) schemes

The analysis starts with 4S1E(4) schemes, which can be obtained by imposing the first-order condition on energy conservation on the families of schemes satisfying fourth-order conditions on the solution. As examples, some of the one-parameter classes of schemes identified by Kutta are analyzed, which are reported in the book by Butcher [73] and labelled as case I–case V. The corresponding Butcher arrays are reported in Table 4.5.

In the first special case identified by Kutta (case I), given a generic sequence of forms, identified as usual by the sequence of α_i, the first-order condition on energy conservation gives: $(\alpha_1+\alpha_4)\mu(\theta)+(\alpha_2+\alpha_3)=0$ which leads to the requirement that the function $\mu(\theta)$ assumes specific values in correspondence of each sequence of α_i,

$$\mu(\theta) = -\frac{(\alpha_2+\alpha_3)}{(\alpha_1+\alpha_4)}. \tag{4.9}$$

The result is that the sequences AAAD and DAAA produce two 4S1E(4) schemes, corresponding to $\theta=(1\pm\sqrt{3})/2$, as well as the sequences AADA and ADAA, which also produce two 4S1E(4) schemes corresponding to $\theta=(2\pm\sqrt{2})/4$. All the other sequences produce zeroth-order schemes on energy conservation.

The case II family of schemes is a 4S1E(4) family, independently on θ, for the two sequences ADDA and AADA. The cases III, IV and V are 4S1E(4) schemes with the sequences ADAA, AADA and ADDA respectively. Note that class V schemes are of particular interest, since the *classical* RK4 scheme belongs to this class for the case $\theta=1/3$. The analysis shows that when implemented with a sequence ADDA this Runge-Kutta scheme has a first order conservation of energy.

case I

0				
$1-\theta$	0			
$\frac{6\theta^2(1-2\theta)}{\gamma(\theta)}$	$\frac{6\theta^2}{\gamma(\theta)}$	0		
$\frac{12\theta^3-24\theta^2+17\theta-4}{2(1-\theta)\mu(\theta)}$	$\frac{\theta(1-2\theta)}{2(1-\theta)\mu(\theta)}$	$\frac{1-\theta}{\mu(\theta)}$	0	
$\frac{\mu(\theta)}{\gamma(\theta)}$	$\frac{1}{\gamma(\theta)}$	$\frac{1}{\gamma(\theta)}$	$\frac{\mu(\theta)}{\gamma(\theta)}$	

case II

0				
θ	0			
$\frac{1}{2}-\frac{1}{8\theta}$	$\frac{1}{8\theta}$	0		
$\frac{1}{2\theta}-1$	$-\frac{1}{2\theta}$	2	0	
$\frac{1}{6}$	0	$\frac{2}{3}$	$\frac{1}{6}$	

case III

0				
$\frac{1}{2}$	0			
$-\frac{1}{12\theta}$	$\frac{1}{12\theta}$	0		
$-\frac{1}{2}-6\theta$	$\frac{3}{2}$	6θ	0	
$\frac{1}{6}-\theta$	$\frac{2}{3}$	θ	$\frac{1}{6}$	

case IV

0				
1	0			
$\frac{3}{8}$	$\frac{1}{8}$	0		
$1-\frac{1}{4\theta}$	$-\frac{1}{12\theta}$	$\frac{1}{3\theta}$	0	
$\frac{1}{6}$	$\frac{1}{6}-\theta$	$\frac{2}{3}$	θ	

case V

0				
$\frac{1}{2}$	0			
$\frac{1}{2}-\frac{1}{6\theta}$	$\frac{1}{6\theta}$	0		
0	$1-3\theta$	3θ	0	
$\frac{1}{6}$	$\frac{2}{3}-\theta$	θ	$\frac{1}{6}$	

Table 4.5: The five cases of one-parameter families of fourth-order Runge-Kutta schemes, as reported in [73]. In the case I family, $\gamma(\theta) = 12\theta(1-\theta)$, $\mu(\theta) = 6\theta - 1 - 6\theta^2$ and $a_{31} + a_{32} = \theta$. This family produces fourth-order schemes provided that $a_{21} \notin \{0, 1/2, 1/2 \pm \sqrt{3}/6, 1\}$.

3S2E(4) schemes

In addition to 4S1E(4) schemes above illustrated, the ten coefficients arising in four-stage Runge-Kutta methods can be fixed by imposing, for each of the possible sequences of alternating forms, a third-order accuracy on the solution (four equations) together with a second-order accuracy on energy conservation (four equations). The solution of the eight-equation nonlinear system (when available) will produce 3S2E(4) schemes. Also, since the system of equation is overdetermined, in general one expects that families of schemes, having one or more parameters, will appear. The nonlinear systems associated to the 14 possible sequences of alternating forms all have four common equations, given by the third-order constraints for a four-stage method

$$\begin{aligned} &b_1 + b_2 + b_3 + b_4 = 1, \\ &b_2 a_{21} + b_3(a_{31} + a_{32}) + b_4(a_{41} + a_{42} + a_{43}) = \frac{1}{2}, \\ &b_2 a_{21}^2 + b_3(a_{31} + a_{32})^2 + b_4(a_{41} + a_{42} + a_{43})^2 = \frac{1}{3}, \\ &a_{21}(b_3 a_{32} + b_4 a_{42}) + b_4 a_{43}(a_{31} + a_{32}) = \frac{1}{6}. \end{aligned} \tag{4.10}$$

In addition to these equations there is the first-order energy-conservation constraint

$$\alpha_1 b_1 + \alpha_2 b_2 + \alpha_3 b_3 + \alpha_4 b_4 = 0, \tag{4.11}$$

where, as usual, the coefficients α_i depend on the particular sequence of forms, and three equations for second-order conservation of energy, whose structure depends on the particular sequence. In Appendix B the four equations associated to second-order conservation of energy are reported for all the possible sequences of alternating forms. Each of these nonlinear systems can be separately studied in order to obtain families of 3S2E(4) schemes for every given sequence. The complete characterization of all the possible solutions for each alternating sequence is, again, out of the scopes of the present study. However, some general considerations, together with some particular solutions, will be presented, as it has been done for previous cases. In many circumstances symbolic nonlinear solvers can also be employed to simplify the task of obtaining solutions.

The structure of the nonlinear system is such that in many cases significative simplifications can be obtained by firstly considering the equations for first-order accuracy on both solution and energy. These two equations involve only the b_i's and immediately furnish the numerical value of one of the unknowns (when three forms are equal within the sequence) or of the sum of two unknowns. In some circumstances this information alone can be sufficient in order to conclude that the non-linear system has no solutions, as in the cases of the sequences AAAD and AADD. In general, it is found that in many cases a next useful step is the derivation of the b_i's as parametric functions of the a_{ij}'s by solving the linear equations in the b_i's. At this point, the remaining fully nonlinear equations can be attacked. It results that, among the possible

1D/ADDA$_1$–3S2E(4)

0			
$\frac{3}{2}$	0		
$\frac{1}{3}$	0	0	
$\frac{14}{25}$	$\frac{28}{75}$	0	0
$\frac{1}{28}$	0	$\frac{2}{3}$	$\frac{25}{84}$

1D/ADDA$_2$–3S2E(4)

0			
$\frac{1}{3}$	0		
$\frac{3}{2}$	0	0	
$\frac{14}{25}$	0	$\frac{28}{75}$	0
$\frac{1}{28}$	$\frac{2}{3}$	0	$\frac{25}{84}$

1D/ADAD–3S2E(4)

0			
$\frac{1}{3}$	0		
$\frac{14}{25}$	δ_1	0	
0	0	$\frac{1}{3}$	0
δ_2	$\frac{1}{4}$	$\frac{25}{84}$	$\frac{5}{12}$

Table 4.6: Optimized 3S2E(4)–type Runge-Kutta schemes. For the 1D/ADAD–3S2E(4) scheme, $\delta_1 = \frac{37+1/3}{100}$ and $\delta_2 = \frac{177+11/7}{5\times 10^3}$.

14 series, the six sequences starting with AA or DD have no solution, together with the two sequences ADDD and DAAA. The remaining six sequences have solutions which are parametrized by one or more constants. Such families of schemes are usually quite difficult to express compactly, but representative examples can be obtained by fixing one or more constants.

As an example, the case of the sequence ADDA is considered, for which the complete nonlinear system for the coefficients is given by Eqs. (4.10), for third-order accuracy on the solution, and by Eqs. (B.20) for second-order accuracy on energy conservation. By making the assumption $a_{32} = 0$ and $b_2 = 0$ or $b_3 = 0$, after some manipulations one obtains the two families:

0			
$\frac{25\theta-42}{75\theta-28}$	0		
$\frac{1}{3}$	0	0	
$\frac{14}{25}$	$\frac{28}{75}-\theta$	θ	0
$\frac{1}{28}$	0	$\frac{2}{3}$	$\frac{25}{84}$

0			
$\frac{1}{3}$	0		
$\frac{75\theta+98}{225\theta}$	0	0	
$\frac{14}{25}$	$\frac{28}{75}-\theta$	θ	0
$\frac{1}{28}$	$\frac{2}{3}$	0	$\frac{25}{84}$

.

The choice of the parameter $\theta = 0$ for the first family and $\theta = 28/75$ for the second is particularly convenient from a computational point of view and leads to the schemes reported in Table 4.6.

For the other sequences, solutions to the corresponding nonlinear system are in general quite difficult to obtain analytically. In all these cases, however, a symbolic solver is usually able to derive one or more families of solutions typically parametrized by two constants. From these expressions one can optimize the scheme by imposing additional requirements. An example of the result of this procedure is the scheme reported in Table 4.6, which has been obtained with the aid of a symbolic manipulator and by requiring a maximum number of zeros in the Butcher array for the sequence ADAD.

Note that for some methods, $c_i > 0$ occurs (cfr. 1D/DA–2S1E(2) and 1D/ADDA$_{1,2}$–3S2E(4) schemes.) In such cases, care must be taken when these methods are used in conjunction with time-dependent terms [73].

4.2.4 Order of accuracy verification

An order of accuracy study has been carried out to numerically verify the predicted orders of accuracy. In Fig. 4.2 and Fig. 4.3, numerical results are presented concerning the time-step convergence of the errors on energy conservation for various alternating schemes. Results are obtained by integrating the inviscid Burgers' equation with periodic boundary conditions on the domain $[0,1]$, discretized by $N = 100$ equidistant mesh points, with initial condition $u_0(x) = \sin(\pi x)$ and $\Delta t = 10^{-4}$. A fourth-order explicit centered scheme is used for spatial integration.

Figures 4.2 and 4.3 show the time-step convergence of the relative error on energy conservation

$$\varepsilon(t) = \frac{E(t) - E_0}{E_0}, \tag{4.12}$$

at $t_f = t_b/2$, where t_b is the break-time at which characteristic lines intersect.

In Fig. 4.2, the convergence of the relative error on energy conservation for different Runge-Kutta schemes with 2 and 3 stages is reported. The curve labelled 1D/AD-2S1E(2) is relative to the two-stage Runge-Kutta scheme reported in Table 4.2, while the curve labelled 1D/ADA-3S1E(3) is the classical Kutta scheme with the ADA alternation of forms. The curves displaying second-order convergence are relative to the schemes given in Table 4.4. The plot shows that all the schemes exhibit the correct scaling with time step Δt.

Figure 4.3 shows the time step convergence of energy conservation for the four-stage schemes. The scheme 1D/ADAA–4S1E(4) is the fourth-order RK scheme labelled as case III, for the value of the parameter $\theta = 1/12$. The scheme 1D/ADDA–4S1E(4) is the *classical* fourth-order Runge-Kutta scheme belonging to case V for $\theta = 1/3$. In all cases the theoretically predicted scaling is recovered.

4.3 Schemes for Navier-Stokes equations

In this Section, the strategy outlined in Table 4.1 is applied to the incompressible Navier-Stokes equations.

4.3.1 Three-stage methods

As in the 1D case, two families of new schemes are expected: 2S2E(3) (6 conditions) and 3S1E(3) (5 conditions) for a total number of $2^s - 2$ possible sequences of advective and divergence forms.

As regards 2S2E(3)–type schemes, no solutions were found, for any of the 6 sequences available. This result is in contrast with the one-dimensional case, where

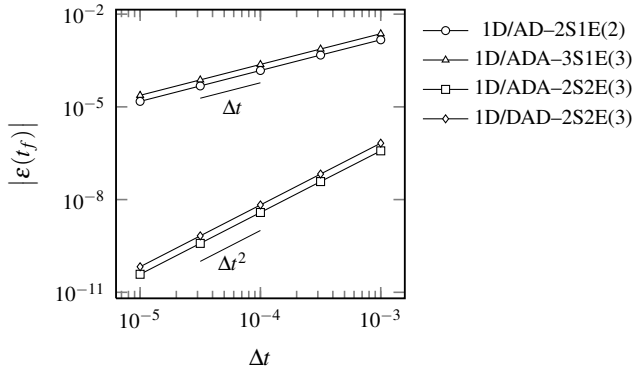


Figure 4.2: Convergence of the relative error on energy conservation as a function of the time step Δt for alternating Runge-Kutta schemes with 2 and 3 stages.

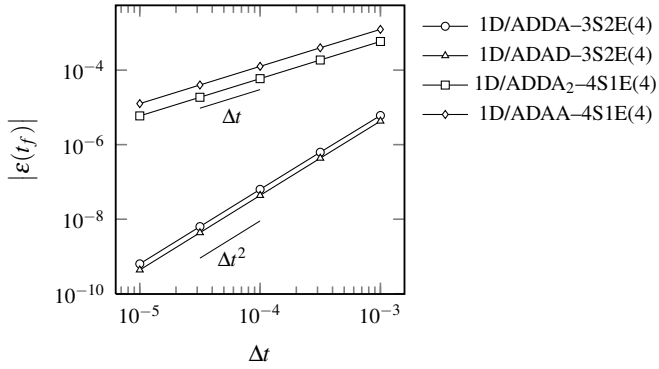


Figure 4.3: Convergence of the relative error on energy conservation as a function of the time step Δt for alternating Runge-Kutta schemes with 4 stages.

plenty of results were obtained. The reason is probably due to the inherent structure of the nonlinear system for three-stage methods, which favors the -2:1 ratio of the transformation relations between divergence and advective quadratic forms that occurs in a one-dimensional setting.

On the other hand, the derivation of 3S1E(3)–type schemes is straightforward and can be achieved by imposing the order-1 condition on energy conservation on the parametrized families given in [73]. For instance, the "case II" and "case III" one-parameter families of schemes reported in [73]

$$
\begin{array}{c|ccc}
 & \text{case II} & & \\
0 & & & \\
\frac{2}{3} & 0 & & \\
\frac{2}{3}-\frac{1}{4b_3} & \frac{1}{4b_3} & 0 & \\
\hline
 & \frac{1}{4} & \frac{3}{4}-b_3 & b_3
\end{array}
\qquad
\begin{array}{c|ccc}
 & \text{case III} & & \\
0 & & & \\
\frac{2}{3} & 0 & & \\
-\frac{1}{4b_3} & \frac{1}{4b_3} & 0 & \\
\hline
 & \frac{1}{4}-b_3 & \frac{3}{4} & b_3
\end{array}
\quad ,
$$

NS/DAD$_{\circlearrowleft}$–3S1E(3)

0			
$\frac{2}{3}$	0		
$-\frac{1}{3}$	1	0	
$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	

NS/DDA$_{\circlearrowleft}$–3S1E(3)

0			
$\frac{2}{3}$	0		
$\frac{1}{6}$	$\frac{1}{2}$	0	
$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	

Table 4.7: 3S1E(3)–type optimized Runge-Kutta schemes for Navier-Stokes equations.

generate 3S1E(3) schemes when the free parameter is fixed in such a way that the linear constraint on first-order energy conservation is enforced for each sequence of forms. The results of this analysis are that "case II" schemes obtained with $b_3 = 1/4$ or $b_3 = 1/2$ ensure first order conservation of energy when employed with the sequences DAD (or ADA) and DDA (or AAD), respectively. The "case III" schemes corresponding to $b_3 = -1/4$ or $b_3 = 1/2$, on the other hand, are 3S1E(3) schemes when employed with the sequences ADD (or DAA) and DDA (or AAD), respectively. These latter schemes possess negative b_i coefficients. This is not desirable in Runge-Kutta methods for stability reasons [73], hence the two schemes derived from case II are to be preferred and are reported in Table 4.7.

4.3.2 Four-stage methods

In four-stage methods, 10 degrees of freedom are available. Hence, either 4S1E(4) or 3S2E(4) schemes can be constructed, which have to be compared to 4S4E(8) skew-symmetric methods.

A large number of 4S1E(4) schemes can easily be obtained by applying the first-order condition on energy to the parametrized fourth-order families provided in Table 4.5. Perhaps the most convenient choice is to use an alternating version of the *classical* Runge-Kutta scheme (RK4), as it can be conveniently implemented. If one employs an ADAD or a DADA sequence inside the stages, the RK4 satisfies the first-order condition on energy conservation while retaining fourth-order accuracy on solution. Hereinafter, the alternating RK4 will be considered as the reference NS/ADAD$_{\circlearrowleft}$–4S1E(4), and is reported in Table 4.8.

In the case of 3S2E(4)–type schemes, solutions exist only for the two couples of sequences ADAD/DADA and ADDA/DAAD. The nonlinear systems have all four equations in common, which correspond to the third-order conditions already reported in Eq. (4.10). These equations have to be augmented by one linear constraint and three nonlinear equations to have second-order accuracy on energy conservation. The form of these equations depends on the specific sequence under study, see Appendix B. For the cases under study, the equations will be reported here for convenience. For the

couple of sequences ADAD/DADA one has, respectively,

$$\begin{aligned}
&b_1 - b_2 + b_3 - b_4 = 0,\\
&2a_{31}b_3 - 2a_{32}b_3 + 4a_{42}b_4 - 2(b_1+b_3)(b_1+b_4) = 0,\\
&(b_2+b_4)^2 - 2a_{32}b_3 = 0,\\
&(b_1+b_3)^2 + 2a_{31}b_3 - 4a_{21}b_2 - 4a_{41}b_4 - 4a_{43}b_4 = 0,
\end{aligned} \tag{4.13}$$

and

$$\begin{aligned}
&b_1 - b_2 + b_3 - b_4 = 0,\\
&4a_{31}b_3 - 2a_{21}b_2 - 2b_4(a_{41}+2a_{42}-2a_{43}) - 2(b_1+b_3)(b_1+b_4) = 0,\\
&(b_2+b_4)^2 + 2a_{42}b_4 - 4a_{32}b_3 = 0,\\
&(b_1+b_3)^2 - 2a_{21}b_2 - 2a_{41}b_4 - 2a_{43}b_4 = 0.
\end{aligned} \tag{4.14}$$

The two systems provide two identical families of solutions, each with two free parameters, which can be expressed by the following Butcher array

$$\begin{array}{c|cccc}
0 & & & \\
\frac{3\theta_1 \pm \mathcal{A}}{6\theta_1} & 0 & & \\
\frac{1}{8\theta_1} & \frac{1}{8\theta_1} & 0 & \\
-\frac{3\theta_1 - 16\theta_1\theta_2 + 4\theta_2\mathcal{B} + 4\theta_1\mathcal{B} \pm 2\mathcal{A}}{24\theta_1\theta_2} & \frac{1}{8\theta_2} & \frac{\theta_1 \pm \mathcal{A}}{6\theta_2} & 0 \\
\hline
\frac{1}{2} - \theta_1 & \frac{1}{2} - \theta_2 & \theta_1 & \theta_2
\end{array},$$

where

$$\mathcal{A} = 3\theta_2\sqrt{-\frac{\theta_1(10\theta_1 - 3)}{3\theta_2(2\theta_2 - 1)}}, \qquad \mathcal{B} = \theta_1 \pm \mathcal{A}. \tag{4.15}$$

The system admits solutions for

$$\begin{cases} \theta_1 \in]-\infty, 0[\cup [3/10, +\infty[\\ \theta_2 \in]0, 1/2[\end{cases} \cup \begin{cases} \theta_1 \in]0, 3/10] \\ \theta_2 \in]-\infty, 0[\cup]1/2, +\infty[. \end{cases} \tag{4.16}$$

In this range, the parameters θ_1 and θ_2 can be optimized to achieve various goals, e.g., low dissipation and dispersion characteristics, or a computationally efficient implementation. For the latter case, a convenient choice is $\theta_1 = 3/10$ and $\theta_2 = 7/20$, which leads to the Butcher array given in Table 4.8. However, this families of schemes have a limited stability range, see Section 4.3.3.

The sequences ADDA/DAAD lead to the following conditions:

$$\begin{aligned}
&b_1 - b_2 - b_3 + b_4 = 0,\\
&4a_{32}b_3 + 2a_{41}b_4 - 2a_{42}b_4 - 2a_{43}b_4 - 2(b_1+b_4)(b_2+b_3) = 0,\\
&(b_2+b_3)^2 - 2a_{42}b_4 - 2a_{43}b_4 = 0,\\
&(b_1+b_4)^2 + 2a_{41}b_4 - 4a_{21}b_2 - 4a_{31}b_3, = 0
\end{aligned} \tag{4.17}$$

NS/ADAD$_{\circlearrowleft}$–4S1E(4)

0				
$\frac{1}{2}$	0			
0	$\frac{1}{2}$	0		
0	0	1	0	
$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$	

NS/ADAD$_{\circlearrowleft}$–3S2E(4)

0				
$\frac{1}{2}$	0			
$\frac{5}{12}$	$\frac{5}{12}$	0		
0	$\frac{5}{14}$	$\frac{1}{7}$	0	
$\frac{1}{5}$	$\frac{3}{20}$	$\frac{3}{10}$	$\frac{7}{20}$	

NS/ADDA$_{\circlearrowleft}$–3S2E(4)

0				
$\frac{1}{3}$	0			
0	1	0		
$\frac{1}{3}$	0	$\frac{1}{3}$	0	
$\frac{1}{8}$	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{3}{8}$	

Table 4.8: 4S1E(4)– and 3S2E(4)–type schemes for Navier-Stokes equations.

and

$$\begin{aligned}
&b_1 - b_2 - b_3 + b_4 = 0,\\
&2a_{32}b_3 - 2a_{31}b_3 - 2a_{21}b_2 + 4a_{41}b_4 - 2\left(b_1 + b_4\right)\left(b_2 + b_3\right) = 0,\\
&\left(b_1 + b_4\right)^2 - 2a_{21}b_2 - 2a_{31}b_3 = 0,\\
&\left(b_2 + b_3\right)^2 + 2a_{32}b_3 - 4a_{42}b_4 - 4a_{43}b_4 = 0.
\end{aligned} \tag{4.18}$$

In order to solve the systems, it turns out to be convenient to add two conditions on some coefficients to set them to zero. For instance, adding such conditions on a_{31} and a_{42}, it is easy to derive the Butcher array reported in Table 4.8. In the following section, it is shown that the two-parameter family of schemes for the sequences ADAD/DADA has the absolute stability footprint of a three-stage scheme. On the contrary, the second scheme is fourth-order accurate for linear equations. Therefore, the latter will be the reference 3S2E(4) scheme for the numerical tests presented in the following chapter.

It is worth noting that many of the schemes presented in this section as well as in Section 4.2 can be further optimized to achieve multiple purposes, which were not taken into consideration in the present work. For instance, the remaining degrees of freedom of the proposed 3S2E–type methods can be exploited to optimize the dispersion and dissipation properties in wavenumber space [75–77], or to improve the accuracy of the pressure, as in [78]. Moreover, low-storage implementations [79] have not been taken into account.

4.3.3 Linear stability of new Runge-Kutta schemes

The absolute stability of the newly derived Runge-Kutta schemes can be studied in a classical way by evaluating the so-called *stability function* [73; 80]. The latter is derived by applying a RK scheme to a linear differential equation $u'(t) = \lambda u(t)$. For s-stage, s-order standard Runge-Kutta methods, the stability function is easily proved

to be a polynomial of degree s of the type

$$R(\sigma) = 1 + \sigma + \frac{\sigma^2}{2!} + ... + \frac{\sigma^s}{s!}, \tag{4.19}$$

where $\sigma = \lambda \Delta t$. If the Butcher array is indicated concisely as

$$\begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array},$$

then the stability function is given by

$$R(\sigma) = 1 + \sigma \mathbf{b}^T \left(\mathbf{I} - \sigma \mathbf{A}\right)^{-1} \mathbf{1}^T. \tag{4.20}$$

Equation (4.20) can be used to calculate the stability function of the newly derived Runge-Kutta schemes given in Section 4.3. In particular, one of the two-parameter families of NS/ADAD$_{\circlearrowleft}$-3S2E(4) schemes has the stability function

$$\begin{aligned} R(\sigma) = 1 + \sigma + \left(\frac{1}{2} - \frac{\sqrt{3}\theta_2 \mathcal{A}}{9\theta_1}\right)\sigma^2 + \left(\frac{1}{6} + \frac{\sqrt{3}\mathcal{A}}{36\theta_1}\right)\sigma^3 + \\ + \left(\frac{\sqrt{3}\mathcal{A}}{216\theta_1} - \frac{15\theta_1 - 9/2}{\theta_1\left(576\theta_2 - 288\right)} - \frac{24\theta_1 - 9}{576\theta_1}\right)\sigma^4. \end{aligned} \tag{4.21}$$

A term-to-term comparison between Eq. (4.20) and Eq. (4.21) shows that in order to match the low-order coefficients, it must result $\mathcal{A} = 0$, hence $\theta_1 = 3/10$. However, this choice leads to a violation of the fourth-order coefficient. The same result is obtained with the other two-parameter family. As a consequence, the NS/ADAD$_{\circlearrowleft}$–3S2E(4) schemes can not be fourth-order accurate for linear equations, and therefore have the linear stability footprint of a three-stage, third-order scheme.

On the contrary, by applying Eq. (4.20) to the Butcher array of NS/ADDA$_{\circlearrowleft}$–3S2E(4) schemes, as reported in Section 4.3.2, it can be found that it has the stability function of a four-stage, fourth-order scheme, and hence the same stability region. It is thus to be preferred to NS/ADAD$_{\circlearrowleft}$–3S2E(4) schemes.

5. Numerical results

In this chapter, the alternating strategy is applied to a number of numerical tests. The aim of the analysis is twofold: to prove the consistency of the theoretical framework and to assess the performances of the new schemes. For the Navier-Stokes equations, the NS/ADDA$_{\circlearrowleft}$–3S2E(4) will be considered as the reference 3S2E scheme, while the alternating RK4 will be the reference 4S1E method. For clarity, they will be indicated simply as 3S2E and 4S1E. The schemes are systematically compared to the RK4 in fully skew-symmetric form (i.e., a 4S4E(8) scheme). The final goal of the tests is to show that the new methods provide similar results to the skew-symmetric form at a reduced computational cost.

5.1 1D efficiency analysis

The performances of the schemes derived in Section 4.2 have been assessed by means of a comparison in terms of computational efficiency, i.e., on an Error-Cost plane.

Several choices can be made regarding the error. In accordance with the aim of the study, the kinetic energy conservation error is considered here. As a representative unit time, $t_f = t_b/2$ is chosen, where t_b is the break-time at which characteristic lines intersect for the inviscid Burgers' equation. The computational cost is calculated by means of the cost metric presented in Appendix A.

The results are reported in the graph of Fig. 5.1. The plot was constructed by analyzing the energy errors in a range of Courant numbers $C = u\Delta t/\Delta x$, in particular $0.1 < C < C_{\max}$, where $C_{\max}$ is the maximum bound allowed by linear stability for each scheme. The cost is then obtained by multiplying the number of floating-point operations by the number of time-steps needed to integrate the inviscid Burgers' equation from t_0 to t_f. The left-end of each curve corresponds to the simulation run at the maximum Courant number. In the upper portion of the graph, the corresponding Courant number dependence is also reported for convenience. One 3S1E, two 4S1E and two 3S2E alternating schemes are compared to three-stage and four-stage Runge-Kutta methods in fully skew-symmetric form. For the latter, the classical Kutta and RK4 methods are considered, respectively. The computational cost for all the curves is normalized with respect to the cost of the RK4 in skew-symmetric form run at the maximum Courant number. The results were obtained by integrating the inviscid Burgersplo' equation with periodic boundary conditions on the domain $[0,1]$, discretized by $N = 100$ equidistant mesh points, with initial condition $u_0(x) = \sin(\pi x)$. A fourth-order explicit central scheme was used for spatial integration. Varying the

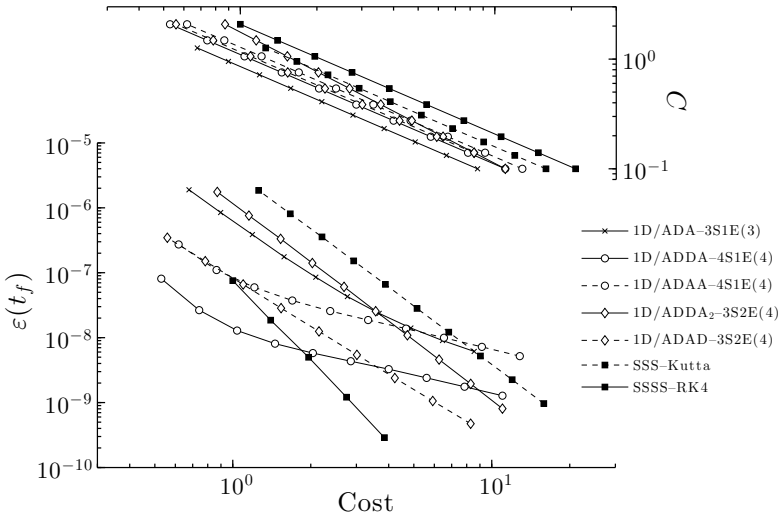


Figure 5.1: Comparison in terms of computational efficiency between classical and novel schemes.

number of mesh points did not significantly affect the qualitative picture in terms of efficiency comparison among the various schemes. Several conclusions can be drawn from the graph. For the smallest error levels, the classical RK4 in skew-symmetric form is the most efficient scheme, due to the high order of accuracy, together with the particularly compact Butcher tableau. However, there is a very large range of energy errors in which several alternating schemes are the most efficient ones. For such methods, the plot shows that the possible computational time saving can be up to 50%, although this value can be rather case dependent. In particular, the ADDA–4S1E, i.e. the alternating version of the classical RK4 scheme, results as the most efficient method. This result is particularly remarkable, since many computational codes built upon the skew-symmetric RK4 can be easily recast in a more efficient procedure by simply switching the SSSS sequence to the alternating ADDA one. By using the same arguments, it can also be concluded that the three-stage Kutta method in skew-symmetric form can be profitably substituted by its alternating counterpart. It is worth reminding that for classical skew-symmetric spatial discretizations, the global energy error is constituted only by the temporal part, while for the new methods it is a mix of spatial and temporal contributions.

Further insight can be gained from the following considerations. The crossover point in efficiency is the result of a balance between two effects: the reduction in computational cost of the alternating procedure and the higher formal order of accuracy on energy conservation of the skew-symmetric schemes. At lower values of C the formal order of accuracy prevails, while at higher values the errors of the two approaches become comparable and the maximum time saving is reached.

In the case previously discussed, the crossover point in efficiency occurs for Courant

numbers greater than ≈ 0.4. It has to be underlined that this point can shift somewhere to the left or right depending on various parameters, such as the final integration time or the initial condition. However, additional numerical tests (not reported here) show that the main trends are retained in a variety of situations. The fact that the better performances of the novel schemes occur in the range of moderate values of the Courant number is advantageous, since this is the range of practical interest for many applications.

As an alternative, the performances of the ADAD–3S2E scheme are also particularly good. In addition, one has to consider that 3S2E schemes have two remaining degrees of freedom, that can be further exploited to achieve specific targets, depending on the problem under study.

5.2 2D periodic inviscid flow

Two-dimensional inviscid flow simulations are carried out on a periodic domain to confirm the theoretical results obtained in the previous sections.

5.2.1 Random flow

At first, an order of accuracy analysis is performed. To this aim, following [47], a square region of size $2\pi \times 2\pi$ is considered, discretized on a mesh of 16×16 mesh points. The solenoidal initial velocity field is constructed from a stream function of random numbers, normalized such that $\mathbf{1}^T \mathbf{u}_x = \mathbf{1}^T \mathbf{u}_y = 0$ and $E_0 = 1.0$, where $E_0 = E(t = 0)$. A random initial condition guarantees that the flowfield is devoid of any symmetries, and hence prevents any apparent order improvement. A centered second-order finite-difference scheme is used for convection; for time-advancement, the various Runge-Kutta schemes developed in Section 4.3 are used.

Figure 5.2 (left) shows the time step convergence of the energy error for three Runge-Kutta methods measured at $t_f = 10$. The slopes confirm the predicted orders of accuracy for the new schemes 4S1E(4) and 3S2E(4), proving the consistency of the theoretical framework. The divergence form gives a constant error, which corresponds to the spatial contribution of Eq. (3.11); a similar behavior is found for the advective form (not shown here). The first part of the time-evolution of the energy error is shown in Fig. 5.2. The non-conservative formulations show a large error growing steeply during the first instants of the simulation. On the contrary, the skew-symmetric method as well as the alternating Runge-Kutta schemes remain bounded; in particular, the 3S2E(4) scheme is identical to the skew-symmetric method within the plotting accuracy. It is interesting to observe that, in the very first part of the simulation, the divergence and advective forms provide errors which are approximately equal and of opposite sign, as predicted by Eq. (2.25), that is a linear approximation. Then, nonlinearity takes place and the the violation in energy-conservation leads to a sudden blow-up.

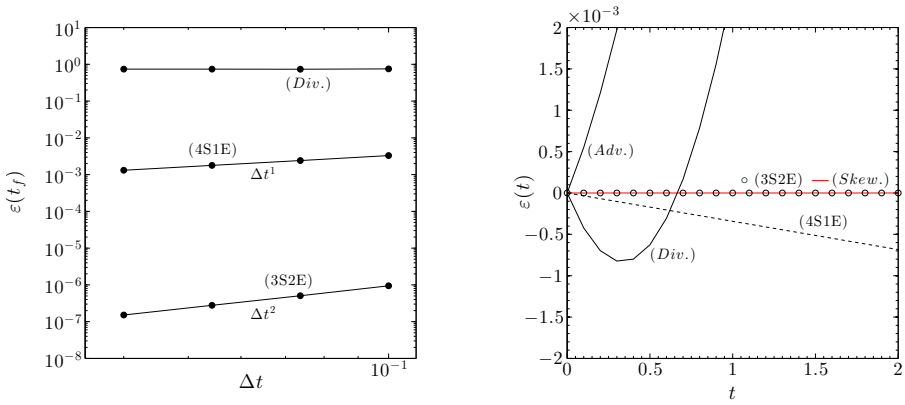


Figure 5.2: Results for 2D periodic, inviscid flow simulations. *(Left)* Time-step convergence of the relative error on energy conservation *(Right)* Time evolution of kinetic energy conservation error; all simulations have been performed with $\Delta t = 0.1$.

Scheme	$\varepsilon(t=2)$
$(Div.)$	0.0315
$(Adv.)$	0.0200
$(Skew.)$	-5.3157×10^{-9}
(3S2E)	-4.9925×10^{-8}
(4S1E)	-6.8404×10^{-4}

Table 5.1: Kinetic energy conservation errors at $t = 2$ (see Figure 5.2) for different Runge-Kutta methods and convective forms.

Table 5.1 gives a more detailed comparison among the errors provided by the various methods.

5.2.2 Temporal mixing layer

The temporal mixing layer is a well-known test case to study the properties of a numerical algorithm in terms of handling steep gradients and formation of smaller scales. The domain is a square region of size $2\pi \times 2\pi$ with periodic boundary conditions. A very coarse mesh of 20^2 points is used to prove the robustness of the various schemes. Simulations are carried out with a constant time-step $\Delta t = 10^{-2}$ and no viscosity. The same numerical setting used for the previous test is adopted here. The initial flowfield

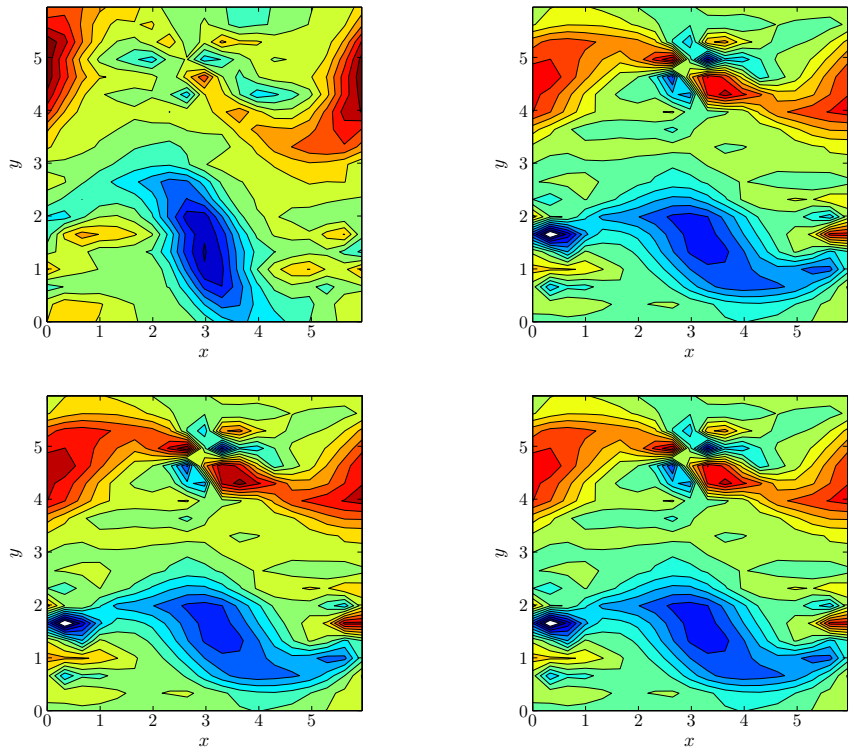


Figure 5.3: Vorticity contours of mixing layer at t=8. Iso-contours ranging from -4 to 4 with steps 0.5. *(Top-left)* Convective term computed in advective form. *(Top-right)* Convective term computed in skew-symmetric form. *(Bottom-left)* Method 4S1E(4), sequence DADA. *(Bottom-right) Method 3S2E(4), sequence DAAD.*

is given by

$$u = \begin{cases} \tanh\left(\dfrac{y-\pi/2}{\delta}\right), & y \leq \pi, \\ \tanh\left(\dfrac{3\pi/2-y}{\delta}\right), & y > \pi, \end{cases} \tag{5.1}$$

$$v = \varepsilon \sin(x). \tag{5.2}$$

The perturbation on the v-velocity is used to promote the roll-up of the shear layer; as in [66], $\varepsilon = 0.05$ is used. A single instability mode is activated by choosing $\delta = \pi/15$ [81].

In Fig. 5.3, the iso-contours of the vorticity field at $t = 8$ are shown for the advective (top-left) and the skew-symmetric form (top-right). The latter is identical to the ones resulting from 4S1E(4) and 3S2E(4) schemes (bottom figures). While for

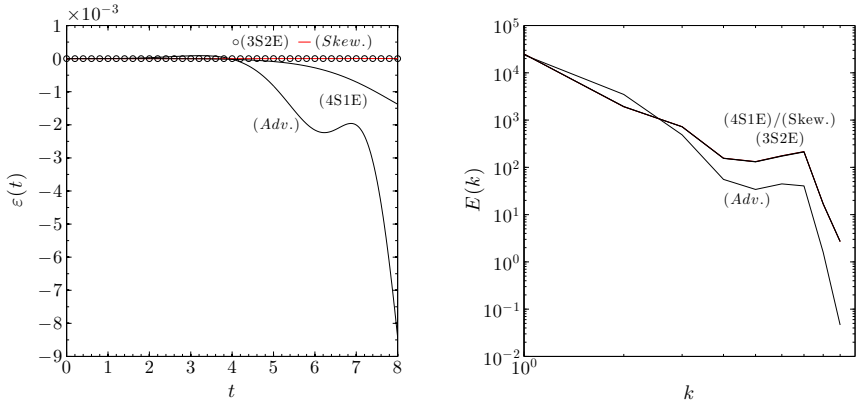


Figure 5.4: Results for the mixing layer. *(Left)* Time-evolution of kinetic energy conservation error. *(Right)* Two-dimensional energy spectra at $t = 8$.

this test the divergence form blew-up due to over-production of energy, the advective form dissipates the kinetic energy of the flow, and the resulting vorticity field is much more smeared in comparison to the conservative formulations. Further insight is given by observing the time-evolution of the energy error $\varepsilon(t)$ as well as the two-dimensional energy spectra calculated at $t = 8$, both shown in Fig. 5.4. The advective form provides a large amount of dissipation, especially at the highest wavenumbers. The alternating schemes perform remarakably well, particularly the 3S2E(4), which is again identical to the skew-symmetric method within the plotting accuracy.

5.3 Homogeneous Isotropic Turbulence (HIT)

5.3.1 Inviscid HIT

As a second test-case, simulations of inviscid homogeneous isotropic turbulence (HIT) are performed. A standard pseudo-spectral code has been adopted, using a projection method in spectral space to enforce the divergence-free condition. The pseudo-spectral solver uses a classical parallel design based on two-dimensional subdivisions. The 3D Fast Fourier Transform (FFT) is split in three one-dimensional transforms. The parallelization is carried out by subdividing the 3D geometry along only 2 directions: along Y and Z in the real space and along X and Y on the spectral space. The code has been recently used to test a hybrid spectral-particle method for the turbulent transport of a passive scalar [82]. All derivatives are performed in spectral space and all products are performed in physical space. Modified wavenumbers are used to emulate centered second-order finite-difference schemes [53]. The tests are performed without subgrid-scale (SGS) model and the simulations are not de-aliased. The initial condition is designed to satisfy a given energy spectrum, typical of decaying isotropic turbulence. A mesh of 32^3 grid points on a domain of length 2π is used. The simula-

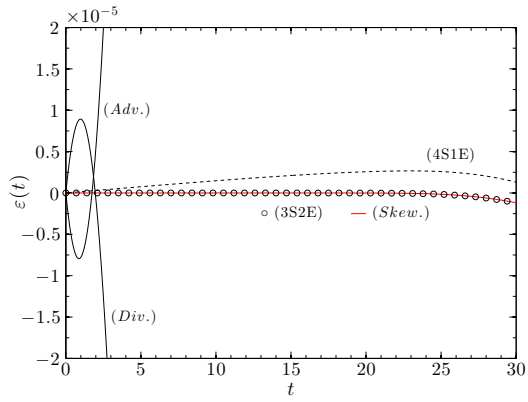


Figure 5.5: Time evolution of kinetic energy conservation error for inviscid HIT.

Scheme	$\varepsilon(t=30)$	T_{CPU}/T_{skew}
$(Div.)$	$+\infty$	0.6429
$(Adv.)$	$+\infty$	0.4810
$(Skew.)$	-1.1816×10^{-6}	1
(3S2E)	-1.1896×10^{-6}	0.5690
(4S1E)	$+1.3044\times10^{-6}$	0.5690

Table 5.2: Kinetic energy conservation errors at $t=30$ (see Figure 5.5) and corresponding normalized CPU time for different Runge-Kutta methods and convective forms

tions differ only by the form of the convective term and the Runge-Kutta scheme used for time advancement.

The time evolution of the turbulence kinetic energy is shown in Fig. 5.5. The advective and divergence forms soon diverge due to violation of kinetic energy conservation. On the other hand, the energy-preserving formulations are stable. In particular, the 3S2E(4) scheme is very close to the skew-symmetric form, while the 4S1E(4) shows a slight over-production of energy. These considerations are confirmed by the computational results shown in Table 5.2, in which the CPU times are also reported.

In this test, the projection method does not require any iterative algorithm, hence computation of the convective term has an important impact on the global computational time. The proposed Runge-Kutta schemes allow a saving in computational time of about 43% with respect to the skew-symmetric form. This important time-saving is independent of the mesh-size. In the spectral algorithm used in this work, it is due to the fact that the skew-symmetric form needs roughly to perform twice more forward and inverse FFTs for each sub-step of the Runge-Kutta scheme than those required by divergence or advective forms alone. It was proved that also in a finite-difference context the computational time required by the skew-symmetric form is approximately

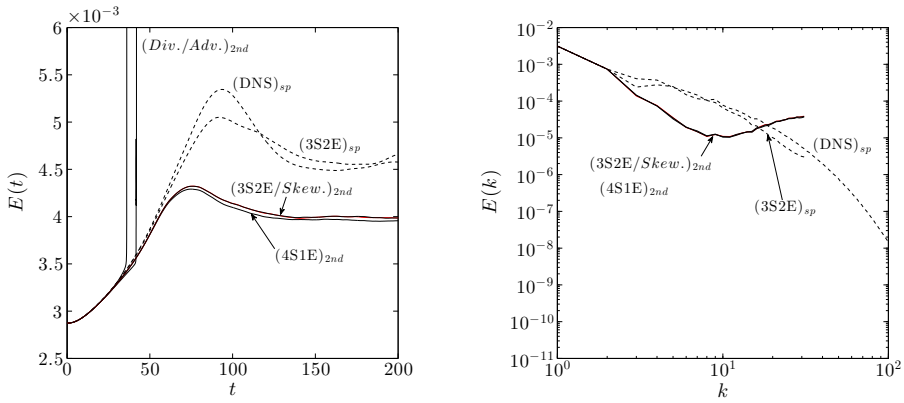


Figure 5.6: Results for forced HIT. (*Left*) Time evolution of kinetic energy. (*Right*) Three-dimensional energy spectra.

doubled [83]. Finally, it is noteworthy that for the 3S2E(4) scheme used, the error on energy is dissipative, hence the scheme is stable for long-time simulations.

5.3.2 Large-eddy simulation of forced HIT

In the last test, large-eddy simulation of forced HIT is performed and compared with direct numerical simulations. The same pseudo-spectral code of the previous test is used here, with the diffusive term being computed exactly. The forcing strategy consists only in "freezing" the low wavenumbers ($|\vec{k}| < 3$), to avoid the introduction of randomness as in classic forcing schemes [84]. In this test, centered second-order finite-difference schemes are mimicked by using modified wavenumbers. A dynamic Smagorinsky model is used to account for subgrid-scales [14; 85]. For the DNS the Navier-Stokes equations are discretized using 256^3 grid points on a domain of length 2π, whereas the LES are performed on 64^3 grid points. The Reynolds number based on the Taylor micro-scale is around 90, which is a typical value for DNS to be performed on 256^3 grid points. The time step is computed from the CFL condition, with a maximum CFL number of 0.5. All the results shown in this work correspond to no-dealiased simulations. Similar results are found with a dealiasing procedure, because the dealiasing only impacts the highest wavenumbers which are already neglected by the use of lower-order schemes. Note that for LES, the skew-symmetric form adds only 15% of additional CPU time, because the SGS model computation with the dynamic procedure has a significant impact on overall CPU time. It is known that the use of a SGS model written in physical space has a large impact on the computational time, especially when a dynamic procedure is used to compute the model coefficient. In fact, a pseudo-spectral code needs to compute various forward and inverse FFTs to apply the model. The SGS model contribution to the overall computational cost could be reduced by using spectral SGS models.

Figure 5.6 shows the kinetic energy evolution in time and the three-dimensional energy spectra for various Runge-Kutta schemes and convective forms. Two spatial schemes are considered for LES: second-order and spectral differentiation. A spectral DNS curve is also shown for comparison. In the spectral case (dashed curves) all methods coincide, both for LES and DNS. Differences emerge when a second-order modified wavenumber is used: the divergence and advective forms are unstable, while the 4S1E(4) scheme gives a slight under-prediction of energy. In both cases, the 3S2E(4) scheme is confirmed to be practically equivalent to the skew-symmetric form, its predictions being identical within the plotting accuracy. Similar results can be found in wavenumber space, showing that the beneficial aliasing properties of the skew-symmetric form are also retained by the new schemes.

6. Conclusions

A novel time-advancing strategy has been developed with the aim of reproducing the energy-conservation properties of skew-symmetric-based schemes at a reduced computational cost. The method is based on a Runge-Kutta scheme in which the divergence or advective forms for convection are suitably alternated within the stages to retain the conservation properties of the skew-symmetric form up to a certain order of accuracy. A comprehensive theoretical framework has been established to derive new alternating Runge-Kutta methods with prescribed order of accuracy on both solution and energy-conservation. It is found that one and three additional constraints are required to obtain first- and second-order accuracy on energy conservation, respectively. The novel method has been applied both to the Burgers' equation and the incompressible Navier-Stokes equations. By being efficient and discretely energy-preserving at the same time, the new method is particularly suitable for DNS and LES of turbulent flows.

Extensive numerical tests have shown that the newly derived schemes can be more efficient than traditional methods, and that the skew-symmetric form results are well reproduced in a variety of situations.

The advantage in terms of CPU time depends upon the details of the algorithm. In general, the new schemes are highly convenient is situations where the computation of the nonlinear term takes a large part of the overall CPU time. For instance, in a periodic, fully-spectral framework, the Poisson equation as well as the viscous terms need a negligible computational effort, and the time-saving proved to reach up to 43% of the total time. The method is then very suitable for spectral codes for direct numerical simulations of homogeneous isotropic turbulence. In finite-difference discretizations, with pressure calculated in physical space by an iterative algorithm, the time saving is expected to drop to about 10%. However, a further improvement could be achieved by the use of the approximate projection method presented in Appendix C, in which the Poisson equation is solved only at the final step of the Runge-Kutta iteration.

Among the various schemes that have been derived, the ones labeled as 3S2E(4)–type, i.e., third-order accurate on solution and second-order accurate on energy conservation, yielded remarkably good results, and are thus highly recommended to be used in place of the more costly skew-symmetric form. The alternating version of the classical RK4 also proved to be particularly efficient.

The ideas presented in this work can be extended in many ways. A promising subject of future work would be the application of the underlying idea to situations with at least one direction of non-homogeneity, e.g. a channel flow. Discretizations

that satisfy a discrete summation-by-parts rule [86], could be coupled to the alternating paradigm proposed here, leading to cost-effective methods. Also, the application of the novel method to compressible flows, in which no costly pressure Poisson equations have to be solved, would lead to a significant gain in computational efficiency.

A. Cost metric

In this Appenix, a proper cost metric is developed. The aim is to evaluate the computational effort required to integrate the non-linear convective term by means of a Runge-Kutta scheme. Explicit centred finite-difference schemes are considered; the advective, divergence and skew-symmetric forms are considered.

A.1 Operations count

In a one-dimensional setting, an explicit central difference discretization of the non-linear convective term reads, for the i-th grid point:

$$\partial_x \frac{u^2}{2}\bigg|_i \approx \sum_{l=1}^{L} w_l \left(u_{i+l}^2 - u_{i-l}^2\right), \tag{A.1}$$

$$u\partial_x u|_i \approx u_i \sum_{l=1}^{L} w_l \left(u_{i+l} - u_{i-l}\right), \tag{A.2}$$

in case of divergence and advective form, respectively. In (A.1) and (A.2), L determines the size of the computational stencil and thus the accuracy of the approximation (i.e., $2L$), while it is implicitly assumed that the coefficients of the scheme already take into account the grid spacing. On a grid of N elements, the number of floating-point operations required to calculate the above terms (*spatial* operations) is equal for the divergence and advective terms. It involves N multiplications to evaluate the products u^2 or $u \cdot \partial u$, plus L multplications and L sums per node to calculate the difference formula, yielding a total of

$$O_{\mathrm{a,d}}^{sp} = N(1+2L). \tag{A.3}$$

It is assumed that additions and multiplications take roughly the same computational effort. The skew-symmetric form requires the evaluation of advective and divergence forms separately, together with a linear combination of these quantities, for a total of:

$$O_{\mathrm{s}}^{sp} = 2N(1+2L) + 3N. \tag{A.4}$$

If an explicit s-stage Runge-Kutta scheme is adopted for time-advancement:

$$\begin{cases} u^{n+1} = u^n - \Delta t \sum_{k=1}^{s} b_k \mathcal{N}(U^k) \\ U^k = u^n - \Delta t \sum_{j=1}^{k-1} a_{kj} \mathcal{N}(U^j), \end{cases} \tag{A.5}$$

then the number of floating-point operations contains, assuming that Δt is already taken into account by the coefficients:

- sO^{sp} evaluations of the non-linear term;
- $\hat{b}N$ sums and $\hat{b}N$ multiplications for the final update, where $\hat{b}$ is the number of non-zero b_k coefficients;
- $\hat{a}N$ multiplications between the non-zero (and non unity) coefficients a_{kj} and the terms $\mathcal{N}(U^k)$;
- $(s-1)N$ sums between u^n and the other terms, within the stages, except for the first stage;
- $(\hat{a}-1)N$ sums between the remaining terms, within the stages.

The *total* (i.e. spatial and temporal) number of operations is then

$$O^{tot}_{\mathrm{a,d}} = 2N(s+Ls+\hat{a}+\hat{b}-1), \tag{A.6}$$

for advective or divergence forms, while for the skew-symmetric form is

$$O^{tot}_{\mathrm{s}} = 2N(3s+2Ls+\hat{a}+\hat{b}-1). \tag{A.7}$$

By comparing Eqs. (A.6) and (A.7), an average factor of order 2 can be established between the approaches.

A.2 Numerical validation

The developed cost metric has been validated against the measured CPU times obtained with a Fortran 90 numerical code. Two reference Runge-Kutta schemes have been considered for validation, namely the three-stage Kutta scheme (RK3) and the classical RK4 from Table 2.2, for both the convective, divergence and skew-symmetric cases, and for a second- and fourth-order difference scheme. The measured CPU times have been averaged over 10 runs. Various parameters have been varied during the numerical experiments (e.g., the number of nodes, the size of the arrays, the number of time-steps, etc.) showing minor effect on the results. The comparison between the cost metric and the CPU times is reported in Table A.1. The agreement between the operation count and the actual CPU times is satisfactory, if one considers that the cost metric does not take into account issues related to storage or access to array variables. The results may also depend slightly upon the optimization level of the compiler and on the architecture of the computer.

Scheme	$\hat{a}$	$\hat{b}$	s	$L=1$				$L=2$			
				$O_{\mathrm{a,d}}^{tot}$	T_{cpu}	O_{s}^{tot}	T_{cpu}	$O_{\mathrm{a,d}}^{tot}$	T_{cpu}	O_{s}^{tot}	T_{cpu}
RK3	2	2	3	18	1.00	32	1.46	24	1.29	48	2.19
RK4	2	2	4	22	1.28	46	1.90	30	1.69	62	2.86

Table A.1: Comparison between cost metric estimates and measured CPU times for two Runge-Kutta schemes and various spatial discretizations. The CPU times have been normalized with respect to the RK3 scheme with second-order central difference in advective/divergence form. For convenience, the number of operations is reported for $N = 1$.

B. Energy-conservation relations

In the following, the first equation represents the first-order constraint; the subsequent three are the relations needed to enforce second-order conservation of energy.

B.1 1D Burgers' equation

B.1.1 Three-stage schemes

ADA

$$\begin{aligned}
&2b_1 - b_2 + 2b_3 = 0 \\
&b_1^2 - 2b_2a_{21} + 2b_3a_{31} + 2b_1b_3 + b_3^2 = 0 \\
&b_3\left(a_{32} - 2a_{31}\right) + b_2b_3 + b_1b_2 = 0 \\
&b_2^2 - 4b_3a_{32} = 0
\end{aligned} \tag{B.1}$$

AAD

$$\begin{aligned}
&2b_1 + 2b_2 - b_3 = 0 \\
&2a_{21}b_2 - b_1b_3 - b_2b_3 = 0 \\
&b_3^2 = 0 \\
&-2b_1^2 - 4b_1b_2 - 2b_2^2 - 4a_{21}b_2 + 4a_{31}b_3 + 4a_{32}b_3 = 0
\end{aligned} \tag{B.2}$$

ADD

$$\begin{aligned}
&-2b_1 + b_2 + b_3 = 0 \\
&9a_{32}b_3 - 8b_1b_2 - 8b_1b_3 = 0 \\
&-b_2^2 - 2b_2b_3 - b_3^2 - a_{32}b_3 = 0 \\
&-4b_1^2 + 8a_{21}b_2 + 8a_{31}b_3 - a_{32}b_3 = 0
\end{aligned} \tag{B.3}$$

DAA

$$\begin{aligned}
&-b_1 + 2b_2 + 2b_3 = 0 \\
&2a_{32}b_3 - a_{31}b_3 - a_{21}b_2 - b_1b_2 - b_1b_3 = 0 \\
&-b_1^2 + 4a_{21}b_2 + 4a_{31}b_3 = 0 \\
&-2b_2^2 - 4b_2b_3 - 2b_3^2 - 4a_{32}b_3 = 0
\end{aligned} \tag{B.4}$$

DAD

$$
\begin{aligned}
&b_1 - 2b_2 + b_3 = 0 \\
&a_{31}b_3 - a_{21}b_2 - b_1b_2 - b_2b_3 = 0 \\
&-b_1^2 - 2b_1b_3 - b_3^2 + 4a_{21}b_2 = 0 \\
&-2b_2^2 + 4a_{32}b_3 = 0
\end{aligned}
\tag{B.5}
$$

DDA

$$
\begin{aligned}
&b_1 + b_2 - 2b_3 = 0 \\
&9a_{21}b_2 - 8a_{31}b_3 - 8a_{32}b_3 - 8b_1b_3 - 8b_2b_3 = 0 \\
&-b_1^2 - 2b_1b_2 - b_2^2 - a_{21}b_2 + 4a_{31}b_3 + 4a_{32}b_3 = 0 \\
&4b_3^2 + a_{21}b_2 = 0
\end{aligned}
\tag{B.6}
$$

B.1.2 Four-stage schemes

AAAD

$$
\begin{aligned}
&2b_1 + 2b_2 + 2b_3 - b_4 = 0 \\
&2b_2a_{21} + 2b_3(a_{31} + a_{32}) - b_4(b_1 + b_2 + b_3) = 0 \\
&b_4^2 = 0 \\
&2b_2a_{21} + 2b_3(a_{31} + a_{32}) - 2b_4(a_{41} + a_{42} + a_{43}) + (b_1 + b_2 + b_3)^2 = 0
\end{aligned}
\tag{B.7}
$$

AADA

$$
\begin{aligned}
&2b_1 + 2b_2 - b_3 + 2b_4 = 0 \\
&2b_2a_{21} + b_4(2a_{41} + 2a_{42} - a_{43}) - b_3(b_1 + b_2 + b_4) = 0 \\
&b_3^2 - 4b_4a_{43} = 0 \\
&2b_2a_{21} - 2b_3(a_{31} + a_{32}) + 2b_4(a_{41} + a_{42}) + (b_1 + b_2 + b_4)^2 = 0
\end{aligned}
\tag{B.8}
$$

AADD

$$
\begin{aligned}
&2b_1 + 2b_2 - b_3 - b_4 = 0 \\
&2b_2a_{21} + b_4a_{43} - (b_1 + b_2)(b_3 + b_4) = 0 \\
&(b_3 + b_4)^2 = 0 \\
&2b_2a_{21} - 2b_3(a_{31} + a_{32}) - 2b_4(a_{41} + a_{42}) + (b_1 + b_2)^2 = 0
\end{aligned}
\tag{B.9}
$$

ADAA

$$
\begin{aligned}
&2b_1 - b_2 + 2b_3 + 2b_4 = 0\\
&b_3(2a_{31} - a_{32}) + b_4\,(2a_{41} - a_{42} + 2a_{43}) - b_2\,(b_1 + b_3 + b_4) = 0\\
&-b_3^2 + 4b_3a_{32} + 4b_4a_{42} = 0\\
&2b_2a_{21} - 2b_3a_{31} - 2b_4(a_{41} - a_{43}) - (b_1 + b_3 + b_4)^2 = 0
\end{aligned}
\tag{B.10}
$$

ADAD

$$
\begin{aligned}
&2b_1 - b_2 + 2b_3 - b_4 = 0\\
&8b_3(2a_{31} - a_{32}) + 9b_4a_{42} - 8\,(b_1 + b_3)\,(b_2 + b_4) = 0\\
&4b_3a_{32} - b_4a_{42} - (b_2 + b_4)^2 = 0\\
&16b_2a_{21} - 16b_3a_{31} + 2b_4\,(8a_{41} - a_{42} + 8a_{43}) - 8(b_1 + b_3)^2 = 0
\end{aligned}
\tag{B.11}
$$

DAAA

$$
\begin{aligned}
&-b_1 + 2b_2 + 2b_3 + 2b_4 = 0\\
&4b_2a_{21} + 4b_3a_{31} + 4b_4a_{41} - b_1^2 = 0\\
&2b_3a_{32} + 2b_4(a_{42} + a_{43}) + (b_2 + b_3 + b_4)^2 = 0\\
&b_2a_{21} + b_3(a_{31} - 2a_{32}) + b_4(a_{41} - 2a_{42} - 2a_{43}) + b_1\,(b_1 + b_3 + b_4) = 0
\end{aligned}
\tag{B.12}
$$

DADA

$$
\begin{aligned}
&-b_1 + 2b_2 - b_3 + 2b_4 = 0\\
&b_2a_{21} - b_3a_{31} + b_4(a_{41} - 2a_{42} + a_{43})(b_1 + b_3)(b_2 + b_4) = 0\\
&4b_2a_{21} + 4b_4(a_{41} + a_{43}) - (b_1 + b_3)^2 = 0\\
&2b_3a_{32} - 2b_4a_{42} - (b_2 + b_4)^2 = 0
\end{aligned}
\tag{B.13}
$$

DADD

$$
\begin{aligned}
&b_1 - 2b_2 + b_3 + b_4 = 0\\
&b_2a_{21} - b_3a_{31} - b_4(a_{41} + a_{43}) + b_2\,(b_1 + b_3 + b_4) = 0\\
&4b_2a_{21} + (b_1 + b_3 + b_4)^2 = 0\\
&2b_3a_{32} + 2b_4a_{42} - b_2^2 = 0
\end{aligned}
\tag{B.14}
$$

DDAA

$$
\begin{aligned}
&b_1 + b_2 - 2b_3 - 2b_4 = 0\\
&b_2a_{21} - 4b_3(a_{31} + a_{32}) - 4b_4(a_{41} + a_{42}) + (b_1 + b_2)^2 = 0\\
&b_2a_{21} + 8b_4a_{43} + 8(b_3 + b_4)^2 = 0\\
&9b_2a_{21} - 8b_3(a_{31} + a_{32}) - 8b_4(a_{41} + a_{42} - 2a_{43}) - 8(b_1 + b_2)(b_3 + b_4) = 0
\end{aligned}
\tag{B.15}
$$

DDAD

$$
\begin{aligned}
&b_1+b_2-2b_3+b_4=0\\
&9b_2a_{21}-8b_3(a_{31}+a_{32})+9b_4(a_{41}+a_{42})-8b_3(b_1+b_2+b_4)=0\\
&b_2a_{21}-4b_3(a_{31}+a_{32})+b_4(a_{41}+a_{42})+(b_1+b_2+b_4)^2=0\\
&b_2a_{21}+b_4(a_{41}+a_{42}-8a_{43})+4b_3^2=0
\end{aligned}
\tag{B.16}
$$

DDDA

$$
\begin{aligned}
&b_1+b_2+b_3-2b_4=0\\
&b_2a_{21}-b_3a_{31}-b_4(a_{41}+a_{43})+b_2\,(b_1+b_3+b_4)=0\\
&4b_2a_{21}+(b_1+b_3+b_4)^2=0\\
&2b_3a_{32}+2b_4a_{42}-b_2^2=0
\end{aligned}
\tag{B.17}
$$

ADDD

$$
\begin{aligned}
&-2b_1+b_2+b_3+b_4=0\\
&9b_3a_{32}+9b_4\,(a_{42}+a_{43})-8b_1\,(b_2+b_3+b_4)=0\\
&b_3a_{32}+b_4\,(a_{42}+a_{43})+(b_2+b_3+b_4)^2=0\\
&8b_2a_{21}+b_3\,(8a_{31}-a_{32})+b_4\,(8a_{41}-a_{42}-a_{43})-4b_1^2=0
\end{aligned}
\tag{B.18}
$$

DAAD

$$
\begin{aligned}
&-b_1+2b_2+2b_3-b_4=0\\
&b_2a_{21}+b_3\,(a_{31}-2a_{32})-b_4a_{41}+(b_1+b_4)\,(b_2+b_3)=0\\
&2b_2a_{21}+2b_3a_{31}-\frac{1}{2}\,(b_1+b_4)^2=0\\
&2b_3a_{32}+2b_4\,(a_{42}+a_{43})+(b_2+b_3)^2=0
\end{aligned}
\tag{B.19}
$$

ADDA

$$
\begin{aligned}
&2b_1-b_2-b_3+2b_4=0\\
&9b_3a_{32}+8b_4\,(2a_{41}-a_{42}-a_{43})-8\,(b_1+b_4)\,(b_2+b_3)=0\\
&b_3a_{32}-4b_4\,(a_{42}+a_{43})+(b_2+b_3)^2=0\\
&8b_2a_{21}+b_3\,(8a_{31}-a_{32})-8b_4a_{41}-8\,(b_1+b_4)^2=0
\end{aligned}
\tag{B.20}
$$

B.2 Navier-Stokes equations

B.2.1 Three-stage schemes

ADA

$$\begin{aligned} & b_1 - b_2 + b_3 = 0 \\ & 2a_{31}b_3 - 2a_{32}b_3 - 2b_1b_2 - 2b_2b_3 = 0 \\ & -b_2^2 + 2a_{32}b_3 = 0 \\ & -b_1^2 - 2b_1b_3 - b_3^2 - 2a_{31}b_3 + 4a_{21}b_2 = 0 \end{aligned} \tag{B.21}$$

AAD

$$\begin{aligned} & b_1 + b_2 - b_3 = 0 \\ & 2a_{21}b_2 - 2b_1b_3 - 2b_2b_3 = 0 \\ & -b_3^2 = 0 \\ & -b_1^2 - 2b_1b_2 - b_2^2 - 4a_{21}b_2 + 4a_{31}b_3 + 4a_{32}b_3 = 0 \end{aligned} \tag{B.22}$$

ADD

$$\begin{aligned} & -b_1 + b_2 + b_3 = 0 \\ & 4a_{32}b_3 - 2b_1b_2 - 2b_1b_3 = 0 \\ & -b_2^2 - 2b_2b_3 - b_3^2 = 0 \\ & -b_1^2 + 4a_{21}b_2 + 4a_{31}b_3 = 0 \end{aligned} \tag{B.23}$$

DAA

$$\begin{aligned} & -b_1 + b_2 + b_3 = 0 \\ & 2a_{32}b_3 - 2a_{31}b_3 - a_{21}b_2 - 2b_1b_2 - 2b_1b_3 = 0 \\ & -b_1^2 + 2a_{21}b_2 + 2a_{31}b_3 = 0 \\ & -b_2^2 - 2b_2b_3 - b_3^2 - 2a_{32}b_3 = 0 \end{aligned} \tag{B.24}$$

DAD

$$\begin{aligned} & b_1 - b_2 + b_3 = 0 \\ & 4a_{31}b_3 - a_{21}b_2 - 2b_1b_2 - 2b_2b_3 = 0 \\ & -b_1^2 - 2b_1b_3 - b_3^2 + 2a_{21}b_2 = 0 \\ & -b_2^2 + 4a_{32}b_3 = 0 \end{aligned} \tag{B.25}$$

DDA

$$
\begin{aligned}
&b_1 + b_2 - b_3 = 0 \\
&3a_{21}b_2 - 2a_{31}b_3 - 2a_{32}b_3 - 2b_1b_3 - 2b_2b_3 = 0 \\
&2b_1^2 + 4b_1b_2 + 2b_2^2 + a_{21}b_2 - 4a_{31}b_3 - 4a_{32}b_3 = 0 \\
&2b_3^2 + a_{21}b_2 = 0
\end{aligned}
\tag{B.26}
$$

B.2.2 Four-stage schemes

AAAD

$$
\begin{aligned}
&b_1 + b_2 + b_3 - b_4 = 0 \\
&2a_{21}b_2 + 2a_{31}b_3 + 2a_{32}b_3 - 2b_1b_4 - 2b_2b_4 - 2b_3b_4 = 0 \\
&b_4^2 = 0 \\
&4a_{41}b_4 - 2a_{31}b_3 - 2a_{32}b_3 - 2a_{21}b_2 + \\
&\quad + 4a_{42}b_4 + 4a_{43}b_4 - 2b_1b_2 - 2b_1b_3 - 2b_2b_3 - b_1^2 - b_2^2 - b_3^2 = 0
\end{aligned}
\tag{B.27}
$$

AADA

$$
\begin{aligned}
&b_1 + b_2 - b_3 + b_4 = 0 \\
&2a_{21}b_2 + 2a_{41}b_4 + 2a_{42}b_4 - 2a_{43}b_4 - 2b_1b_3 - 2b_2b_3 - 2b_3b_4 = 0 \\
&b_3^2 - 2a_{43}b_4 = 0 \\
&+ 2a_{21}b_2 - 4a_{32}b_3 + 2a_{41}b_4 - 4a_{31}b_3 + \\
&\quad + 2a_{42}b_4 + 2b_1b_2 + 2b_1b_4 + 2b_2b_4 + b_1^2 + b_2^2 + b_4^2 = 0
\end{aligned}
\tag{B.28}
$$

AADD

$$
\begin{aligned}
&-b_1 - b_2 + b_3 + b_4 = 0 \\
&2a_{21}b_2 + 4a_{43}b_4 - 2b_1b_3 - 2b_1b_4 - 2b_2b_3 - 2b_2b_4 = 0 \\
&b_3^2 + 2b_3b_4 + b_4^2 = 0 \\
&b_1^2 + 2b_1b_2 + b_2^2 + 2a_{21}b_2 - 4a_{31}b_3 - 4a_{32}b_3 + 4a_{41}b_4 + 4a_{42}b_4 = 0
\end{aligned}
\tag{B.29}
$$

ADAA

$$
\begin{aligned}
&b_1 - b_2 + b_3 + b_4 = 0 \\
&2a_{31}b_3 - 4a_{21}b_2 + 2a_{41}b_4 + 2a_{43}b_4 + \\
&\quad + 2b_1b_3 + 2b_1b_4 + 2b_3b_4 + b_1^2 + b_3^2 + b_4^2 = 0 \\
&b_2^2 - 2a_{32}b_3 - 2a_{42}b_4 = 0 \\
&2a_{31}b_3 - 2a_{32}b_3 + 2a_{41}b_4 - 2a_{42}b_4 + 2a_{43}b_4 - 2b_1b_2 - 2b_2b_3 - 2b_2b_4 = 0
\end{aligned}
\tag{B.30}
$$

ADAD

$$
\begin{aligned}
&b_1 - b_2 + b_3 - b_4 = 0 \\
&2a_{31}b_3 - 2a_{32}b_3 + 4a_{42}b_4 - 2b_1b_2 - 2b_1b_4 - 2b_2b_3 - 2b_3b_4 = 0 \\
&-b_2^2 - 2b_2b_4 - b_4^2 + 2a_{32}b_3 = 0 \\
&-b_1^2 - 2b_1b_3 - b_3^2 - 2a_{31}b_3 + 4a_{21}b_2 + 4a_{41}b_4 + 4a_{43}b_4 = 0
\end{aligned}
\tag{B.31}
$$

DAAD

$$
\begin{aligned}
&b_1 - b_2 - b_3 + b_4 = 0 \\
&-b_1^2 - 2b_1b_4 - b_4^2 + 2a_{21}b_2 + 2a_{31}b_3 = 0 \\
&-b_2^2 - 2b_2b_3 - b_3^2 - 2a_{32}b_3 + 4a_{42}b_4 + 4a_{43}b_4 = 0 \\
&2a_{32}b_3 - 2a_{31}b_3 - 2a_{21}b_2 + 4a_{41}b_4 - 2b_1b_2 - 2b_1b_3 - 2b_2b_4 - 2b_3b_4 = 0
\end{aligned}
\tag{B.32}
$$

DADA

$$
\begin{aligned}
&-b_1 + b_2 - b_3 + b_4 = 0 \\
&4a_{31}b_3 - 2a_{21}b_2 - 2a_{41}b_4 + 2a_{42}b_4 + \\
&\quad - 2a_{43}b_4 - 2b_1b_2 - 2b_1b_4 - 2b_2b_3 - 2b_3b_4 = 0 \\
&-b_1^2 - 2b_1b_3 - b_3^2 + 2a_{21}b_2 + 2a_{41}b_4 + 2a_{43}b_4 = 0 \\
&-b_2^2 - 2b_2b_4 - b_4^2 - 2a_{42}b_4 + 4a_{32}b_3 = 0
\end{aligned}
\tag{B.33}
$$

DADD

$$
\begin{aligned}
&b_1 - b_2 + b_3 + b_4 = 0 \\
&4a_{31}b_3 - 2a_{21}b_2 + 4a_{41}b_4 + 4a_{43}b_4 - 2b_1b_2 - 2b_2b_3 - 2b_2b_4 = 0 \\
&-b_1^2 - 2b_1b_3 - 2b_1b_4 - b_3^2 - 2b_3b_4 - b_4^2 + 2a_{21}b_2 = 0 \\
&-b_2^2 + 4a_{32}b_3 + 4a_{42}b_4 = 0
\end{aligned}
\tag{B.34}
$$

DDAA

$$
\begin{aligned}
&b_1 + b_2 - b_3 - b_4 = 0 \\
&4a_{21}b_2 - 2a_{31}b_3 - 2a_{32}b_3 - 2a_{41}b_4 + \\
&\quad - 2a_{42}b_4 + 2a_{43}b_4 - 2b_1b_3 - 2b_1b_4 - 2b_2b_3 - 2b_2b_4 = 0 \\
&-b_1^2 - 2b_1b_2 - b_2^2 + 2a_{31}b_3 + 2a_{32}b_3 + 2a_{41}b_4 + 2a_{42}b_4 = 0 \\
&-b_3^2 - 2b_3b_4 - b_4^2 - 2a_{43}b_4 = 0
\end{aligned}
\tag{B.35}
$$

ADDD

$$
\begin{aligned}
&-b_1 + b_2 + b_3 + b_4 = 0 \\
&4a_{32}b_3 + 4a_{42}b_4 + 4a_{43}b_4 - 2b_1b_2 - 2b_1b_3 - 2b_1b_4 = 0 \\
&-b_2^2 - 2b_2b_3 - 2b_2b_4 - b_3^2 - 2b_3b_4 - b_4^2 = 0 \\
&-b_1^2 + 4a_{21}b_2 + 4a_{31}b_3 + 4a_{41}b_4 = 0
\end{aligned}
\tag{B.36}
$$

DDAD

$$
\begin{aligned}
&b_1+b_2-b_3+b_4=0\\
&4a_{21}b_2-2a_{31}b_3-2a_{32}b_3+4a_{41}b_4+\\
&\quad+4a_{42}b_4-2b_1b_3-2b_2b_3-2b_3b_4=0\\
&-b_1^2-2b_1b_2-2b_1b_4-b_2^2-2b_2b_4-b_4^2+2a_{31}b_3+2a_{32}b_3=0\\
&-b_3^2+4a_{43}b_4=0
\end{aligned}
\tag{B.37}
$$

DDDA

$$
\begin{aligned}
&b_1+b_2+b_3-b_4=0\\
&4a_{21}b_2+4a_{31}b_3+4a_{32}b_3-2a_{41}b_4+\\
&\quad-2a_{42}b_4-2a_{43}b_4-2b_1b_4-2b_2b_4-2b_3b_4=0\\
&b_4^2=0\\
&b_1^2+2b_1b_2+2b_1b_3+b_2^2+2b_2b_3+b_3^2-2a_{41}b_4-2a_{42}b_4-2a_{43}b_4=0
\end{aligned}
\tag{B.38}
$$

ADDA

$$
\begin{aligned}
&-b_1+b_2+b_3-b_4=0\\
&-b_2^2-2b_2b_3-b_3^2+2a_{42}b_4+2a_{43}b_4=0\\
&-b_1^2-2b_1b_4-b_4^2-2a_{41}b_4+4a_{21}b_2+4a_{31}b_3=0\\
&4a_{32}b_3+2a_{41}b_4-2a_{42}b_4-2a_{43}b_4-2b_1b_2-2b_1b_3-2b_2b_4-2b_3b_4=0
\end{aligned}
\tag{B.39}
$$

DAAA

$$
\begin{aligned}
&-b_1+b_2+b_3+b_4=0\\
&2a_{32}b_3-2a_{31}b_3-2a_{21}b_2-2a_{41}b_4+\\
&\quad+2a_{42}b_4+2a_{43}b_4-2b_1b_2-2b_1b_3-2b_1b_4=0\\
&-b_1^2+2a_{21}b_2+2a_{31}b_3+2a_{41}b_4=0\\
&-2a_{32}b_3-2a_{42}b_4-2a_{43}b_4-2b_2b_3-2b_2b_4-2b_3b_4-b_2^2-b_3^2-b_4^2=0
\end{aligned}
\tag{B.40}
$$

C. An approximate projection method for the incompressible Navier-Stokes equations

An algorithm for computing unsteady incompressible flow is developed and analyzed. The method is based on a Runge-Kutta procedure of an arbitrary number of stages. In order to save computational time, the Poisson equation for pressure is solved only at the final sub-step as in [87]; however, a better approximation for pressure is used at intermediate stages. Numerical tests show that this modification increases the robustness as well as the formal accuracy of the method, and is thus suitable for finite-difference algorithms based on Runge-Kutta time-stepping schemes.

C.1 Introduction

In the context of time-advancing algorithms for the incompressible Navier-Stokes equations, reported here for convenience

$$\frac{\partial u_i}{\partial t} + \mathcal{N}_i(u) = -\frac{\partial p}{\partial x_i} + \frac{1}{\mathrm{Re}} \frac{\partial^2 u_i}{\partial x_j \partial x_j}, \tag{C.1}$$

$$\frac{\partial u_i}{\partial x_i} = 0, \tag{C.2}$$

the divergence-free condition is commonly enforced by means of fractional-step methods [72]. When a Runge-Kutta scheme is used for time-integration, the Poisson equation for pressure has to be solved at each RK sub-stage to impose the incompressibility constraint and retain the temporal order of accuracy of the procedure. In order to save computational time, Le & Moin proposed a modified algorithm in which velocity was projected only at the final step [87] . During the intermediate sub-stages, the pressure field was approximated to first-order, leading to an overall second-order accuracy and a time-saving of about 40%. On the other hand, the robustness of the method was found to be deficient, due to non-zero divergence of intermediate velocity fields, which can lead to sudden blow-up [88].

In the following, a method is proposed in which the Poisson equation is solved only at the final RK step, as in [87], but with improved robustness and accuracy.

C.2 Numerical Method

The numerical procedure is based on a family of standard explicit Runge-Kutta schemes for time-integration. A s-stage time advancement of Eqs. (C.1) and (C.2) reads

$$\left\{\begin{array}{rcl} \mathbf{u}_i &=& \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\left[\mathcal{L}(\mathbf{u}_j) - \mathcal{N}(\mathbf{u}_j)\right] - \Delta t\ \mathcal{G}(\mathbf{p}_i), \\ \mathbf{u}^{n+1} &=& \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i\left[\mathcal{L}(\mathbf{u}_i) - \mathcal{N}(\mathbf{u}_i)\right] - \Delta t\ \mathcal{G}(\mathbf{p}^{n+1}), \end{array}\right. \tag{C.3}$$

where $\mathbf{u}$ and $\mathbf{p}$ are the discretized velocity and pressure fields, while $\mathcal{L}$, $\mathcal{N}$ and $\mathcal{G}$ represent proper approximations to the Laplacian, non-linear and gradient operators, respectively.

In a classical fractional-step method, the pressure $\mathbf{p}_i$ in Eq. (C.3) is calculated by means of the usual splitting

$$\left\{\begin{array}{rcl} \mathbf{u}_i^* &=& \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\left[\mathcal{L}(\mathbf{u}_j) - \mathcal{N}(\mathbf{u}_j)\right] \\ \mathcal{D}\mathcal{G}(\mathbf{p}_i) &=& \dfrac{1}{\Delta t}\mathcal{D}(\mathbf{u}_i^*) \\ \mathbf{u}_i &=& \mathbf{u}_i^* - \Delta t\ \mathcal{G}(\mathbf{p}_i), \quad i = 2,\ldots,s, \end{array}\right. \tag{C.4}$$

where $\mathcal{D}$ is the discrete divergence operator. The algorithm which results from combining Eqs. (C.4) and (C.3) will be referred to as FS. In this case, s Poisson equations have to be solved for pressure: $(s-1)$ for the intermediate stages and a final one after the last step to project $\mathbf{u}^{n+1,*}$ (assuming $\mathbf{u}^n$ is divergence-free).

A significant saving in CPU time can be obtained if one seeks to solve only one Poisson equation after the last step. This task can be accomplished by using a suitable approximation for pressure within the mid-stages, e.g.

$$\mathbf{p}_i \approx \hat{\mathbf{p}}_i = F(\mathbf{p}^n, \mathbf{p}^{n-1}). \tag{C.5}$$

In particular, two formulas will be considered here:

(*a*) $\hat{\mathbf{p}}_i = \mathbf{p}^n$;
(*b*) $\hat{\mathbf{p}}_i = 2\mathbf{p}^n - \mathbf{p}^{n-1}$,

which correspond to constant and linear extrapolation from previous values, respectively. The splitting in Eq. (C.4) becomes a single equation,

$$\hat{\mathbf{u}}_i = \mathbf{u}^n + \Delta t \sum_{j=1}^{i-1} a_{ij}\left[\mathcal{L}(\mathbf{u}_j) - \mathcal{N}(\mathbf{u}_j)\right] - c_i \Delta t\ \mathcal{G}(\hat{\mathbf{p}}_i), \tag{C.6}$$

where $c_i = \sum_j a_{ij}$. By combining Eq. (C.6) with Eq. (C.4), one obtains a procedure in which no Poisson equations have to be solved for stages $i = 2,\ldots,s$. A final projection

step is still required for $\mathbf{u}^{n+1,*}$. The resulting algorithms will be labeled FSa and FSb depending on the approximation used for $\hat{\mathbf{p}}_i$. Note that the method FSa is actually a generalization of the procedure proposed by [87], hence is globally second-order accurate, assuming the time-integration scheme is at least second-order. It can be shown that the new method FSb is third-order accurate in time on velocity, as it will be demonstrated numerically in the next section.

C.3 Results

C.3.1 2D Taylor-Green Vortex

The 2D Taylor-Green Vortex (TGV) is simulated numerically to confirm the theoretical results presented in the previous sections. The Navier-Stokes equations are solved in a periodic domain of length 2π, discretized on a mesh of 32^2 points. The exact solution is given by

$$\begin{aligned} u(x,y,t) &= -\cos(x)\sin(y)e^{-2\nu t}, \\ v(x,y,t) &= \sin(x)\cos(y)e^{-2\nu t}, \\ p(x,y,t) &= -\frac{1}{4}\left(\cos(2x)+\cos(2y)\right)e^{-4\nu t}. \end{aligned} \tag{C.7}$$

Starting from $t = 0$, Eqs. (C.7) are solved numerically with a second-order centered scheme for both convective and viscous terms. A standard three-stage, third-order Runge-Kutta method is used for time-integration. The three algorithms presented in Section C.2 are used. The temporal order of accuracy is best revealed for vanishing viscosity and is shown in Figure C.1 (left). The results confirm the theoretical predictions. In Figure C.1 (right), the residual divergence of the intermediate velocity fields is compared for methods FSa and FSb in case of $\mathrm{Re} = 1/\nu = 1000$. After the second sub-step, the method FSb provides a substantially lower residual with respect to FSa.

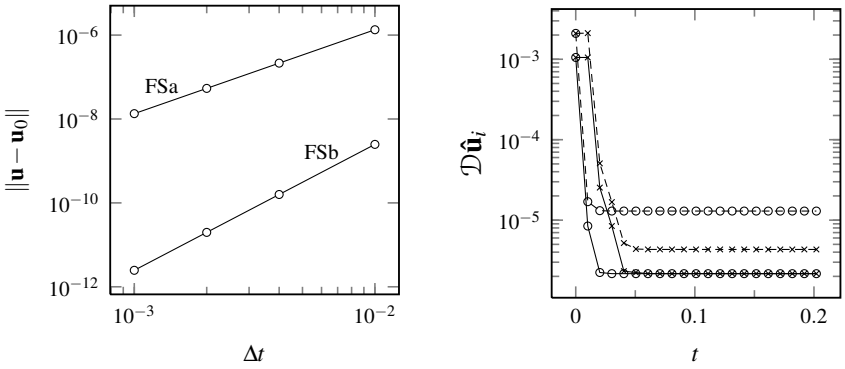


Figure C.1: Results for 2D TGV. (*Left*) Time-step convergence on velocity error for the two approximate projection methods, $\mathrm{Re} \to \infty$. (*Right*) Time evolution of residual divergence after steps $i = 1$ (solid) and $i = 2$ (dashed) for the methods FSa (circles) and FSb ($\times$), $\mathrm{Re} = 1000$.

References

[1] M. VAN DYKE. *An album of Fluid Motion*. The Parabolic Press, 2005. 19

[2] H. TENNEKES AND J.L. LUMLEY. *A First Course in Turbulence*. MIT Press, 1972. 19

[3] T. POINSOT AND D. VEYNANTE. *Theoretical and numerical combustion*. 2012. 19, 21

[4] G. TRYGGVASON, R. SCARDOVELLI, AND S. ZALESKI. *Direct Numerical Simulation of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011. 19

[5] P. MOIN AND J. KIM. **Tackling turbulence with supercomputers**. *Sci. Am.*, **276**(1), 1997. 19

[6] S. POPE. *Turbulent flows*. Cambridge University Press, 2000. 19

[7] P. MOIN AND K. MAHESH. **Direct numerical simulation: a tool in turbulence research**. *Annu. Rev. Fluid Mech.*, **30**:539–578, 1998. 19

[8] P. MOIN, C.W. HAMMAN, AND G. IACCARINO. **Turbulence: Validation and Verification and Uncertainty Quantification analysis of a multi-scale complex system**. 19

[9] D.C. WILCOX. *Turbulence Modeling for CFD*. DCW Industries, 1998. 19

[10] J.W. DEARDORFF. **A numerical study of three-dimensional turbulent channel flow at large Reynolds numbers**. *J. Fluid Mech.*, **41**:453–480, 1970. 19

[11] P. SAGAUT. *Large eddy simulation for incompressible flows*. Springer, 2002. 20

[12] R.S. ROGALLO AND P. MOIN. **Numerical simulation of turbulent flows**. *Annu. Rev. Fluid Mech.*, **16**:99–137, 1984. 20

[13] J.W. SMAGORINSKY. **General circulation experiments with the primitive equations I. The basic experiment**. *Mon. Weather Rev.*, **91**:99–164, 1963. 20

[14] M. GERMANO, U. PIOMELLI, P. MOIN, AND W. H. CABOT. **A dynamic subgrid-scale eddy viscosity model**. *Phys. Fluids A*, **3**:1760–1765, 1991. 20, 72

[15] S.A. ORSZAG AND G.S. PATTERSON. **Numerical simulation of three-dimensional isotropic turbulence**. *Phys. Rev. Lett.*, **28**:76–79, 1972. 20

[16] J. KIM, P. MOIN, AND R. MOSER. **Turbulence statistics in fully-developed channel flow at low Reynolds number**. *J. Fluid Mech.*, **177**:133–166, 1987. 20

[17] H. LE AND P. MOIN. **Direct numerical simulation of turbulent flow over a bacward facing step**. *J. Fluid Mech.*, **330**:349–374, 1997. 20

[18] X. WU AND P. MOIN. **Direct numerical simulation of turbulence in a nominally zero-pressure-gradient flat-plate boundary layer**. *J. Fluid Mech.*, **630**:5–41, 2009. 20

[19] X. WU AND P. DURBIN. **Evidence of longitudinal vortices evolved from distorted wakes in a turbine passage**. *J. Fluid Mech.*, **446**:199–228, 2001. 20

[20] P. MOIN. **Advances in large eddy simulation methodology for complex flows**. *Int. J. Heat Fluid Fl.*, **23**:710–720, 2002. 21

[21] F.F. GRINSTEIN, L.G. MARGOLIN, AND W.J. RIDER. *Implicit Large Eddy Simulation*. Cambridge University Press, 2007. 22

[22] P.J. MASON. **Large-eddy simulation of the convective atmospheric boundary layer**. *J. Atmos. Sci.*, **46**(11):1492–1516, 2002. 21

[23] J.C. OEFELEIN. **Large eddy simulation of turbulent combustion processes in propulsion and power systems**. *Prog. Aerosp. Sci.*, **42**:2–37, 2006. 21

[24] S. CANDEL, D. DUROX, T. SCHULLER, N. DARABIHA, L. HAKIM, AND T. SCHMITT. **Advances in combustion and propulsion applications**. *Eur. J. Mech. B-Fluid*, **40**:87–106, 2013. 21

[25] H. PITSCH. **Large-eddy simulation of turbulent combustion**. *Annu. Rev. Fluid Mech.*, **38**:453–482, 2006. 21

[26] **CTR**. `http://ctr.stanford.edu/`. 21

[27] **CERFACS**. `http://pantar.cerfacs.fr/4-26334-The-AVBP-code.php`. 21

[28] **CORIA**. `http://www.coria-cfd.fr/index.php/SiTCom-B`. 21

[29] S. BORRELLI, P. DE MATTEIS, A. SCHETTINO, AND F. FERRIGNO. **The HYPROB Program: mastering key technologies, design and testing capabilities for space transportation rocket propulsion evolution**. *Proceedings of the International Astronautical Congress*, 2012. 21

[30] F. CAPUANO, A. MASTELLONE, S. DI BENDETTO, L. CUTRONE, AND A. SCHETTINO. **Preliminary developments towards a high-order and efficient LES code for propulsion applications**. *Proceedings of the jointly organized WCCM XI - ECCM V - ECFD VI*, pages 7569–7580, 2014. 21

[31] C. FUREBY. **Towards the use of large eddy simulation in engineering**. *Prog. Aerosp. Sci.*, **44**:381–396, 2008. 21, 23

[32] N.J. GEORGIADIS, D.P. RIZZETTA, AND C. FUREBY. **Large-eddy simulation: current capabilities, recommended practices, and future research**. *AIAA J.*, **48**:1772–1784, 2010. 21

[33] C. CANUTO, M.Y. HUSSAINI, A. QUARTERONI, AND TH.A. ZANG. *Spectral Methods. Fundamentals in Single Domains*. Springer, 2006. 22

[34] M.M. RAI AND P. MOIN. **Direct simulations of turbulent flow using finite-difference schemes**. *J. Comput. Phys.*, **96**:15–53, 1991. 22

[35] R. VICHNEVETSKY AND J.B. BOWLES. *Fourier Analysis of Numerical Approximation of Hyperbolic Equations*. SIAM, 1982. 22

[36] S. GHOSAL. **An analysis of numerical errors in large-eddy simulations of turbulence**. *J. Comput. Phys.*, **125**:187–206, 1996. 22

[37] S. K. LELE. **Compact finite difference schemes with spectral–like resolution**. *J. Comput. Phys.*, **103**:16–42, 1992. 22, 40

[38] P. MOIN. *Fundamentals of Engineering Numerical Analysis*. Cambridge University Press, 2011. 22

[39] R. MITTAL AND P. MOIN. **Suitability of upwind–biased finite difference schemes for large–eddy simulation of turbulent flows**. *J. Comput. Phys.*, **35**:1415–1417, 1997. 22, 23

[40] N. A. PHILLIPS. **An example of nonlinear computational instability**. In *The Atmosphere and the Sea in Motion*, pages 501 – 504. Rockefeller Institute Press, Oxford University Press, 1959. 23

[41] M.R. VISBAL AND D.V. GAITONDE. **On the use of higher-order finite-difference schemes on curvilinear and deforming meshes**. *J. Comput. Phys.*, **181**:155–185, 2002. 23

[42] H. CHOI AND P. MOIN. **Effects of the computational time step on numerical solutions of turbulent flow**. *J. Comput. Phys.*, **113**:1–4, 1994. 23, 32

[43] B. KOREN, R. ABGRALL, P. BOCHEV, J. FRANK, AND B. PEROT. **Physics–compatible numerical methods**. *J. Comput. Phys.*, **257**:1039, 2013. 23

[44] J. B. PEROT. **Discrete conservation properties of unstructured mesh schemes**. *Annu. Rev. Fluid. Mech.*, **43**:299–318, 2011. 23

[45] D. YOU, F. HAM, AND P. MOIN. **Discrete conservation principles in large-eddy simulation with application to separation control over an airfoil**. *Phys. Fluids*, **20**(10), 2008. 24

[46] F.H. HARLOW AND J.E. WELCH. **Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface**. *Phys. Fluids*, **8**(12), 1965. 25

[47] Y. MORINISHI, T. S. LUND, O. V. VASILYEV, AND P. MOIN. **Fully conservative higher order finite difference schemes for incompressible flows**. *J. Comput. Phys.*, **143**:90–124, 1998. 25, 28, 30, 31, 67

[48] K. MAHESH, G. CONSTANTINESCU, AND P. MOIN. **A numerical method for large-eddy simulation in complex geometries**. *J. Comput. Phys.*, **197**:215–240, 2004. 25

[49] O.V. VASILYEV. **High order finite difference schemes on non-uniform meshes with good conservation properties**. *J. Comput. Phys.*, **157**:746–761, 2000. 25

[50] R. W. C. P. VERSTAPPEN AND A. E. P. VELDMAN. **Symmetry–preserving discretization of turbulent flow**. *J. Comput. Phys.*, **187**:343–368, 2003. 25, 29

[51] G. A. BLAISDELL, E. T. SPYROPOULOS, AND J. H. QIN. **The effect of the formulation of nonlinear terms on aliasing errors in spectral methods**. *Applied Numerical Mathematics*, **21**(3):207 – 219, 1996. 25

[52] T. A. ZANG. **On the rotation and skew-symmetric forms for incompressible flow simulations**. *Appl. Numer. Math.*, **7**:27–40, 1991. 25, 28

[53] A. G. KRAVHCENKO AND P. MOIN. **On the effect of numerical errors in large eddy simulations of turbulent flows**. *J. Comput. Phys.*, **131**:310–322, 1997. 25, 28, 30, 70

[54] N. N. MANSOUR, P. MOIN, W. C. REYNOLDS, AND J. H. FERZIGER. **Improved methods for large eddy simulations of turbulence**. *Turb. Shear Flows*, **1**:386–401, 1979. 25

[55] K. HORIUTI. **Comparison of conservative and rotational forms in large eddy simulation of turbulent channel flow**. *J. Comput. Phys*, **71**:343–370, 1987. 25

[56] W. J. FEIEREISEN, W. C. REYNOLDS, AND J. H. FERZIGER. **Numerical Simulation of Compressible, Homogeneous Turbulent Shear Flow**. Technical report, Stanford University. 25

[57] Y. MORINISHI. **Skew-symmetric form of convective terms and fully conservative finite difference schemes for variable density low-Mach number flows**. *J. Comput. Phys*, **229**:276–300, 2010. 25

[58] A. E. HONEIN AND P. MOIN. **Higher entropy conservation and numerical stability of compressible turbulence simulations**. *J. Comput. Phys.*, **201**:531–545, 2004. 25

[59] S. PIROZZOLI. **Generalized conservative approximations of split convective derivative operators**. *J. Comput. Phys.*, **229**(19):7180 – 7190, 2010. 25

[60] F. DUCROS, F. LAPORTE, T. SOULÈRES, V. GUINOT, P. MOINAT, AND B. CARUELLE. **High-Order Fluxes for Conservative Skew-Symmetric-like Schemes in Structured Meshes: Application to Compressible Flows**. *J. Comput. Phys.*, **161**:114–139, 2000. 25

[61] J. C. KOK. **A high-order low-dispersion symmetry-preserving finite-volume method for compressible flow on curvilinear grids**. *J. Comput. Phys.*, **228**:6811–6832, 2009. 25

[62] S. LAIZET AND E. LAMBALLAIS. **High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy**. *J. Comput. Phys.*, **228**:5989–6015, 2009. 25

[63] W. CHENG AND R. SAMTANEY. **A high-resolution code for large eddy simulation of incompressible turbulent boundary layer flows**. *Comput. Fluids*, **92**:82–92, 2014.

[64] J. F. GIBSON. **Channelflow: A spectral Navier-Stokes simulator in C++**. Technical report, U. New Hampshire, 2014. `Channelflow.org`. 25

[65] G. E. KARNIADAKIS AND S. SHERWIN. *Spectral/hp Element Methods for Computational Fluid Dynamics*. Oxford University Press, 2005. 25

[66] B. SANDERSE. **Energy–conserving Runge-Kutta methods for the incompressible Navier-Stokes equations**. *J. Comput. Phys.*, **233**:100–131, 2013. 25, 34, 69

[67] J.M. BURGERS. **A mathematical model illustrating the theory of turbulence**. *Adv. Appl. Mech.*, **1**:171–199, 1948. 27

[68] M.D. LOVE. **Subgrid modelling studies with Burgers' equation**. *J. Fluid Mech.*, **100**:87–110, 1980. 27

[69] S. KIDA. **Asymptotic properties of Burgers turbulence**. *J. Fluid Mech.*, **93**:337–377, 1979. 27

[70] F.X. TRIAS, O. LEHMKUHL, A. OLIVA, C.D. PÉREZ-SEGARRA, AND R.W.C.P. VERSTAPPEN. **Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids**. *J. Comput. Phys.*, **258**(0):246 – 267, 2014. 29

[71] A.J. CHORIN. **Numerical solution of the Navier-Stokes equations**. *Math. Comp.*, **22**:745–762, 1968. 32

[72] J. KIM AND P. MOIN. **Application of a fractional-step method to incompressible Navier-Stokes equations**. *J. Comput. Phys.*, **59**:308–323, 1985. 32, 89

[73] J. C. BUTCHER. *Numerical Methods for Ordinary Differential Equations*. Wiley, 2004. 33, 52, 53, 55, 56, 59, 60, 61, 63

[74] E. HAIRER, G. WANNER, AND C. LUBICH. *Geometric Numerical integration*. Springer, 2006. 34

[75] F. Q. HU, M. Y. HUSSAINI, AND J. L. MANTHEY. **Low-Dissipation and Low-Dispersion Runge-Kutta Schemes for Computational Acoustics**. *J. Comput. Phys.*, **124**:177–191, 1996. 63

[76] M. BERNARDINI AND S. PIROZZOLI. **A general strategy for the optimization of Runge-Kutta schemes for wave propagation phenomena**. *J. Comput. Phys.*, **228**:4182–4199, 2009.

[77] A. NAJAFI-YAZDI AND L. MONGEAU. **A low-dispersion and low-dissipation implicit Runge-Kutta scheme**. *J. Comput. Phys.*, **233**:315–323, 2013. 63

[78] B. SANDERSE AND B. KOREN. **Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations**. *J. Comput. Phys.*, **231**:3041–3063, 2012. 63

[79] J.H. WILLIAMSON. **Low-storage Runge-Kutta schemes**. *J. Comput. Phys.*, **35**:45–86, 1980. 63

[80] D. F. GRIFFITHS AND D. J. HIGHAM. *Numerical Methods for Ordinary Differential Equations*. Springer, 2010. 63

[81] H. KUERTEN B. VREMAN, B. GEURTS. **Comparison of numerical schemes in large-eddy simulation of the temporal mixing layer**. *Int. J. Num. Meth. Fluids*, **22**:297–311, 1996. 69

[82] J.-B. LAGAERT, G. BALARAC, AND G.-H. COTTET. **Hybrid spectral-particle method for the turbulent transport of a passive scalar**. *J. Comput. Phys.*, **260**:127–142, 2014. 70

[83] F. CAPUANO, G. COPPOLA, AND L. DE LUCA. **An efficient time advancing strategy for energy-preserving simulations**. *J. Comput. Phys.*, 2014. Under Review. 72

[84] E. D. SIGGIA AND G. S. PATTERSON. **Intermittency effects in a numerical simulation of stationary three-dimensional turbulence**. *J. Fluid Mech.*, **86**:567–592, 1978. 72

[85] D.K. LILLY. **A proposed modification of the Germano subgrid-scale closure method.** *Phys. Fluids A*, **4**:633–635, 1992. 72

[86] M. SVARD AND J. NORDSTROM. **Review of summation-by-parts schemes for initial–boundary-value problems**. *J. Comput. Phys.*, **268**:17–38, 2014. 76

[87] H. LE AND P. MOIN. **An improvement of fractional step methods for the incompressible Navier-Stokes equations**. *J. Comput. Phys.*, **92**:369–379, 1991. 89, 91

[88] S. BOSE. Private communication, 2014. 89