Turbulence, Heat and Mass Transfer 10

Cost-vs-accuracy analysis of self-adaptive time-integration methods

J. Plana-Riu, F.X. Trias, C.D. Pérez-Segarra and A.Oliva

Heat and Mass Transfer Technological Centre, Technical University of Catalonia, ESEIAAT, Carrer de Colom 11, 08222 Terrassa (Barcelona), Spain, josep.plana.riu@upc.edu

Abstract – The integration in time of the semi-discrete Navier-Stokes equations has been historically bounded by the classic CFL condition, and later on, an innovative method has been defined by ensuring the eigenvalues of the method lay in the boundary of the stability region, yet the effects of this in the resolution have not been tested. In this abstract, an efficiency region in a set of schemes is defined so that the method can alternate among methods in order to keep the maximum efficiency subject to a set of conditions, such as a minimum order of accuracy or a minimum numerical dissipation.

1. Introduction

Solving and understanding turbulence has been one of the problems in which most efforts have been applied within the fluid dynamics community, which is mathematically modelled by the Navier-Stokes equations. Within the framework of the finite volume method (FVM), its space-discretized form is the starting point for most of the methods. This form reads, using the notation from [1], as follows,

$$M\mathbf{u}_s = \mathbf{0}_c, \qquad (1) \qquad\qquad \Omega\frac{d\mathbf{u}_c}{dt} + C(\mathbf{u}_s)\mathbf{u}_c - D\mathbf{u}_c + \Omega G_c \mathbf{p}_c = \mathbf{0}_c, \qquad (2)$$

where M is the face-to-cell divergence operator, Ω_c is a diagonal matrix containing the cell volumes so that $\Omega = I_3 \otimes \Omega_c$, C_c is the cell-to-cell convective operator so that $C = I_3 \otimes C_c$, D_c is the cell-to-cell diffusive operator so that $D = I_3 \otimes D_c$, G_c is the cell-to-cell gradient operator, $\mathbf{u}_s$ is the velocity field defined at the faces, and I_3 is the identity matrix of size 3. Traditionally, the solution of the semi-discretized equations has been obtained with a projection method [2], in which multiple schemes can be applied to advance the equations in time (e.g. second order Adams-Bashforth ,AB2; second- and third- order Runge-Kutta schemes, RK2,RK3; etc.), with their variations in their properties.

It is well-known that by performing the numerical solution of the Navier-Stokes equations, it is possible to introduce artificial dissipation due to the discretizations used. In the work of Verstappen and Veldman [3], and Trias et al. [1], the numerical dissipation due to the space discretization is treated, leading to symmetry-preserving space schemes both in staggered and collocated methods. Nonetheless, there is a reduced set of publications in which the numerical dissipation due to the time discretization is treated. Sanderse [4] proposed the use of symplectic (implicit) Runge-Kutta schemes in order to completely preserve energy. In order not to deal with implicit time integration, Capuano et al. [5] proposed pseudosymplectic schemes, which at the same time are explicit, making them more efficient. Both of these possible sources of numerical dissipation can be put altogehter in a so-called kinetic energy budget, in which the weight of this dissipation can be compared against the physical dissipation.

In order to properly set the time-step, Trias and Lehmkuhl [6] first proposed making use of the stability region so that the integration procedure can be optimized, having the eigenvalues of the method, bounded with Gershgorin's theorem, to be in the boundaries of its stability

region. This led to bigger time-steps while still having stable simulations when compared to the classical CFL method.

2. Runge-Kutta schemes applied to the incompressible Navier-Stokes equations

Starting from the semi-discrete Navier-Stokes equations, Sanderse and Koren [7] proposed the following method for the integration with Runge-Kutta (RK). Summarizing, an s-stage explicit RK can be applied to the integration of the Navier-Stokes equations as

$$\mathbf{u}_i^* = \mathbf{u}_n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathbf{F}_j, \;\; L\Psi_i = \frac{1}{\Delta t} D\mathbf{u}_i^*, \;\; \mathbf{u}_i = \mathbf{u}_i^* - \Delta t G\Psi_i, \quad i = 1,\ldots,s \tag{3}$$

$$\mathbf{u}_{n+1}^* = \mathbf{u}_n + \Delta t \sum_{i=1}^{s} b_i\mathbf{F}_i, \;\; L\Psi_{n+1} = \frac{1}{\Delta t} D\mathbf{u}_{n+1}^*, \;\; \mathbf{u}_{n+1} = \mathbf{u}_{n+1}^* - \Delta t G\Psi_i, \tag{4}$$

where $\mathbf{F}_i = \Omega^{-1}(D - C(\mathbf{u}_{s,i}))\mathbf{u}_{c,i}$, a_{ij}, b_i correspond to the Butcher tableau of the used method for an explicit Runge-Kutta method, $\mathbf{u}_i^*$ corresponds to the predictor velocity, and Φ to a pressure-like variable which corresponds to a first order approximation to the pressure.

If the coefficients from the Butcher tableau satisfy a certain conditions, a pseudosymplectic scheme can be obtained [5] in which energy is conserved for an order higher than the order of accuracy (e.g. a 3p5q(4) method corresponds to a third-order in accuracy, fifth in energy conservation, by using four stages).

Note that these methods will not have either the stability region $R(z), z \in \mathbb{C}$ of a classical RK3 or RK4, which can be easily computed by (5). Instead, their stability regions will need to be computed with its definition (6), in which $b = (b_1 \; b_2 \; \ldots \; b_s)^T$, $\mathbf{1} = (1 \; 1 \; \ldots \; 1)^T \in \mathbb{R}^s$, and $A = [a_{ij}] \in \mathbb{R}^{s\times s}$.

$$R(z) = 1 + \sum_{p=1}^{s} \frac{1}{p!} z^p, \tag{5}$$

$$R(z) = 1 + zb^T(I_s - zA)^{-1}\mathbf{1}. \tag{6}$$

As an example, according to Capuano et al. [5], the 3p5q(4) pseudosymplectic scheme is parameter-dependant, the domain on which is discussed in the paper. Hence, the envelope of the stability regions for all possible parameters will determine the stability region of the so-called α-3p5q(4) method, and can be compared against the RK3 and RK4 schemes, as in Figure 1.

3. From stability regions to computational efficiency regions

The concept of stability regions is widely spread in the numerical solution of ODEs, being Figure 1 a clear example of the concept, as it bounds the maximum value the eigenvalues can take so that the integration remains stable. Usually, the time-integration scheme does not change during the run. Nonetheless, the efficiency region is based on using a closed set of families of schemes (e.g. RK2, RK3, RK4, α-3p5q) so that the most efficient scheme (under certain conditions explained later on) is used at every time. This could be set by normalizing the stability regions, which radius is directly proportional to the maximum stable time-step, by the time required for the computation of a time-step. Hence, this time-step computation time can be directly estimated with the generalized implementation in the in-house code framework,

$$T_{\Delta t}(s) = (s+1)T_{SLAE} + \left[3(s-1)\frac{s+2}{2} + 6(s+1) + 12s\right] T_{axpy} + \left[6(s+1) + 60s\right] T_{SpMV}, \tag{7}$$

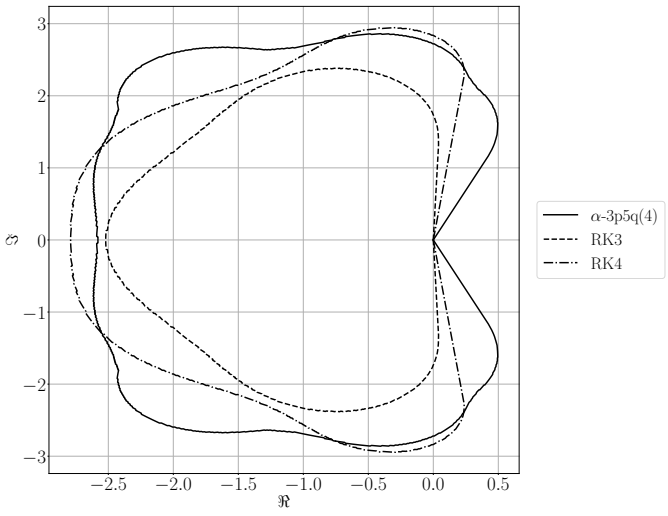

Figure 1: Comparison of the stability regions for a RK3, RK4 and the parameter-dependant α-3p5q(4).

which will work for an arbitrary number of stages s, where T_{SLAE} corresponds to the time spent computing the system of linear algebraic equations (SLAE), which corresponds to a Poisson equation; T_{axpy} is the time spent for a linear combination of vectors, and T_{SpMV} corresponds to the time spent for a sparse matrix-vector product. Hence, the radius of the efficiency region would be proportional to the maximum $\frac{\Delta t}{T_{\Delta t}^{\text{Method}}}$. Nonetheless, in order to simplify the notation,

$$T_{\Delta t}^{\text{Method}} = \tau_{\text{Method}} T_{\Delta t}^{\text{Euler}} = \tau_{\text{Method}} T_{\Delta t}(1) = \tau_{\text{Method}} (2T_{SLAE} + 24T_{axpy} + 72T_{SpMV}), \quad (8)$$

where τ_{Method} is the ratio between $T_{\Delta t}^{\text{Method}}$ and $T_{\Delta t}(1)$, so that the efficiency region will be proportional to $\frac{\Delta t}{\tau_{\text{Method}}}$. Note that (7) only depends on the number of stages, and not on the coefficients, and thus τ_{Method} will depend only in s as well. Hence, the efficiency region for a set of standard Runge-Kutta schemes (Euler, RK2, ..., RK5), with the stability region computed with (5) has been obtained in Figure 2. Note that the maximum efficiency for low angles within the complex plane is obtained with an Euler scheme, and thus, when adding the condition of minimum second-order in time integration the efficiency for low ϕ is notably reduced, while keeping the same efficiency at the most convective cases, and setting a limiting condition in the efficiency region.

Hence, the general method will englobe a set of schemes among which it will be capable of chosing one of the methods from the set such that the efficiency region is maximized in that case, being subject to all of the possible constraints defined by the method.

Note that the method can be straightforwardly extended to multi-step schemes such as an AB2 or AB3, which would have similar computation times compared to an Euler scheme, given that there will not be any additional stages compared to the presented Runge-Kutta methods.

3.1 Kinetic energy budget

Another limiting factor for the efficiency region might be the numerical dissipation generated by both space and time discretization. By assuming the space discretization is symmetry-

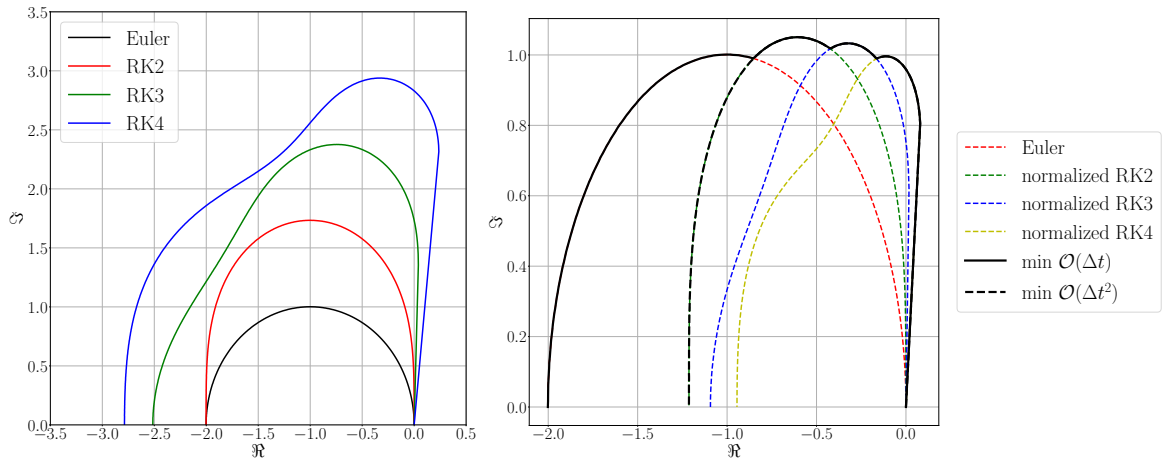

Figure 2: Stability (left) and efficiency (right) regions for a set of standard RK schemes. The efficiency region is computed considering that the solution of the Poisson equation corresponds to an 80% of the iteration time for an Euler scheme.

preserving, the energy variation can be assumed to be

$$\frac{\Delta E}{\Delta t} = \frac{1}{\mathrm{Re}}\Phi + \xi = (1 + f_{RK})\frac{1}{\mathrm{Re}}\Phi, \tag{9}$$

where $\frac{1}{\mathrm{Re}}\Phi$ is the physical dissipation rate, and $\xi = f_{RK}\frac{1}{\mathrm{Re}}\Phi$ is the artificial dissipation generated by the numerical scheme. Thus, a minimum in this parameter $f_{RK} > 0$ will determine how close the method is to be energy-conserving. Thus, a minimum value can be set to further limit the efficiency region.

References

[1] F. X. Trias, O. Lehmkuhl, A. Oliva, C. D. Pérez-Segarra, and R. W. Verstappen, "Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids," *Journal of Computational Physics*, vol. 258, pp. 246–267, 2014-02.

[2] A. J. Chorin, "Numerical Solution of the Navier-Stokes Equations," *Mathematics of Computation*, vol. 22, no. 104, pp. 745–762, 1968.

[3] R. W. Verstappen and A. E. Veldman, "Symmetry-preserving discretization of turbulent flow," *Journal of Computational Physics*, vol. 187, no. 1, pp. 343–368, 2003-05.

[4] B. Sanderse, "Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 233, no. 1, pp. 100–131, 2013.

[5] F. Capuano, G. Coppola, L. Rández, and L. de Luca, "Explicit Runge–Kutta schemes for incompressible flow with improved energy-conservation properties," *Journal of Computational Physics*, vol. 328, pp. 86–94, 2017-01.

[6] F. X. Trias and O. Lehmkuhl, "A self-adaptive strategy for the time integration of Navier-Stokes equations," *Numerical Heat Transfer, Part B: Fundamentals*, vol. 60, no. 2, pp. 116–134, 2011-08.

[7] B. Sanderse and B. Koren, "Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 231, no. 8, pp. 3041–3063, 2012-04.