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

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

Heat and Mass Transfer Technological Centre
Technical University of Catalonia

10th International Symposium on Turbulence, Heat and Mass Transfer

September 11th-15th, 2023
Rome, Italy

Centre Tecnològic de Transferència de Calor
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Contents

Introduction

CFL[1]

First used method to ensure the stability of an explicit integration

$$\frac{du}{dt} + u\frac{du}{dx} = 0$$

$$\downarrow$$

$$\left(\frac{u\Delta t}{\Delta x}\right)_{max} \leq 1$$

[1] Courant, R, Friedrichs, K, and Lewy, H. (1927), "Über die partiellen Differenzengleichungen der matematischen Physik". Mathematische Annalen 100 (1), pp. 32-74

[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Introduction

CFL[1]

First used method to ensure the stability of an explicit integration

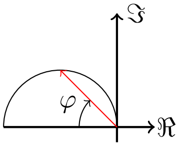
$$\frac{du}{dt} + u\frac{du}{dx} = 0$$

$$\downarrow$$

$$\left(\frac{u\Delta t}{\Delta x}\right)_{max} \leq 1$$

SAT[2]

Computation of the eigenbounds in the predictor velocity step to set the maximum stable Δt



[1] Courant, R, Friedrichs, K, and Lewy, H. (1927), "Über die partiellen Differenzengleichungen der matematischen Physik". Mathematische Annalen 100 (1), pp. 32-74

[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Introduction

CFL[1]

First used method to ensure the stability of an explicit integration

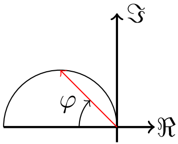
$$\frac{du}{dt} + u\frac{du}{dx} = 0$$

$$\downarrow$$

$$\left(\frac{u\Delta t}{\Delta x}\right)_{max} \leq 1$$

SAT[2]

Computation of the eigenbounds in the predictor velocity step to set the maximum stable Δt



And now?

[1] Courant, R, Friedrichs, K, and Lewy, H. (1927), "Über die partiellen Differenzengleichungen der matematischen Physik". Mathematische Annalen 100 (1), pp. 32-74

[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Runge-Kutta applied to Navier-Stokes

Starting point...

$$M\mathrm{u}_s = 0_c$$

$$\Omega \frac{d\mathrm{u}_c}{dt} + C(\mathrm{u}_s)\mathrm{u}_c - D\mathrm{u}_c + \Omega G_c \mathrm{p}_c = 0_c$$

- Putting together both expressions...

$$\frac{d\mathrm{u}_c}{dt} = \underbrace{(I_n - GL^{-1}M)}_{\text{Projection operator}, P} F(\mathrm{u}_s)\mathrm{u}_c$$

- Hard to compute $PF(\mathrm{u}_s)$, thus projection method is used.

According to Sanderse and Koren [3],

$$\mathrm{u}_i^* = \mathrm{u}_n + \Delta t \sum_{j=1}^{i-1} a_{ij}\mathrm{F}_j \qquad \mathrm{u}_{n+1}^* = \mathrm{u}_n + \Delta t \sum_{i=1}^{s} b_i \mathrm{F}_i$$

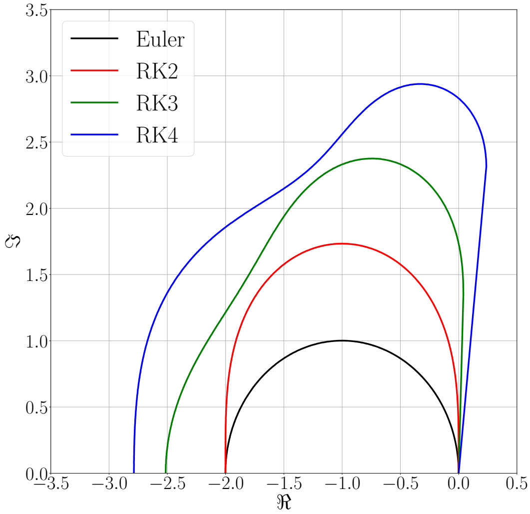
$$L\Psi_i = \frac{1}{\Delta t} D\mathrm{u}_i^* \qquad L\Psi_{n+1} = \frac{1}{\Delta t} D\mathrm{u}_{n+1}^*$$

$$\mathrm{u}_i = \mathrm{u}_i^* - \Delta t G\Psi_i \qquad \mathrm{u}_{n+1} = \mathrm{u}_{n+1}^* - \Delta t G\Psi_{n+1}$$

[3] Sanderse, B., Koren, B. (2012), "Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations", Journal of Computational Physics 231 (8), pp. 3041-3063

Stability region of Runge-Kutta

- Coefficients a_{ij}, b_i from the method: Butcher tableau, $A = [a_{ij}]_{i=1,\ldots,s; j=1,\ldots,s}$, $b = (b_1 \; b_2 \; \ldots \; b_s)$
- In general, $R(z) = 1 + zb^T(I_s - zA)^{-1}1_s$, yet for $p = s$, $R(z) = 1 + \sum_{p=1}^{s} \frac{1}{p!} z^p$



Computation of eigenbounds

- Need to compute the eigenbounds of $F(\mathrm{u}_s) = D - C(\mathrm{u}_s)$
- If D and $C(\mathrm{u}_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem

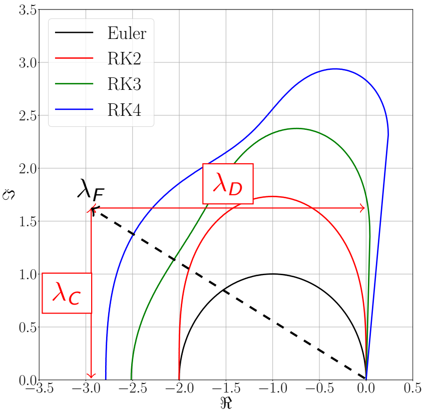
[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Computation of eigenbounds

- Need to compute the eigenbounds of $F(u_s) = D - C(u_s)$
- If D and $C(u_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem



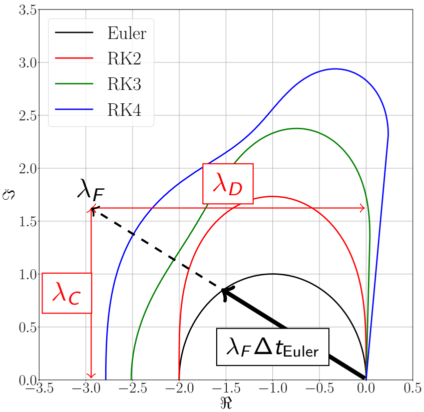
[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Computation of eigenbounds

- Need to compute the eigenbounds of $F(\mathrm{u}_s) = D - C(\mathrm{u}_s)$
- If D and $C(\mathrm{u}_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem



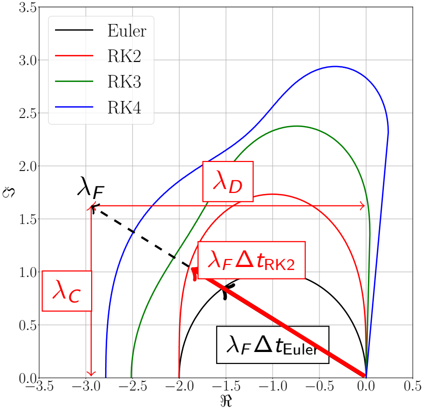
[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Computation of eigenbounds

- Need to compute the eigenbounds of $F(\mathrm{u}_s) = D - C(\mathrm{u}_s)$
- If D and $C(\mathrm{u}_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem



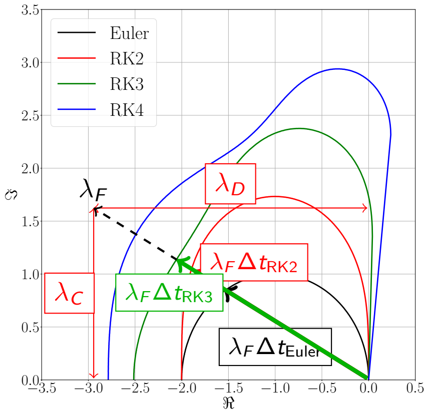
[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Computation of eigenbounds

- Need to compute the eigenbounds of $F(\mathrm{u}_s) = D - C(\mathrm{u}_s)$
- If D and $C(\mathrm{u}_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem



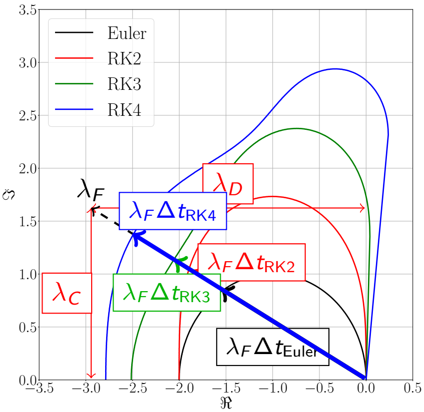
[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

Computation of eigenbounds

- Need to compute the eigenbounds of $F(\mathrm{u}_s) = D - C(\mathrm{u}_s)$
- If D and $C(\mathrm{u}_s)$ are discretized with a symmetry-preserving scheme[2],

$$\lambda_F \leq -|\lambda_D| + i\lambda_C$$

- λ_D and λ_C can be computed independently with Gershgorin circle theorem



[2] Trias, F.X, Lehmkuhl, O. (2011), "A self-adaptive strategy for the time integration of Navier-Stokes equations". Numerical Heat Transfer, Part B: Fundamentals 60 (2), pp. 116-134

From stability to efficiency regions

Do stability-region bounded Δt offer the most efficient integration?

Not always. It provides the most efficient integration given a integration scheme.

From stability to efficiency regions

Do stability-region bounded Δt offer the most efficient integration?

Not always. It provides the most efficient integration given a integration scheme.
What if... instead of using a single scheme, a **pool** of schemes is used?

From stability to efficiency regions

Do stability-region bounded Δt offer the most efficient integration?

Not always. It provides the most efficient integration given a integration scheme.
What if... instead of using a single scheme, a **pool** of schemes is used?

Computational efficiency region

Combination of a set of stability regions normalized by the ratio of wall clock times per iteration between the current scheme and Euler scheme, τ_{Method}.

With the generalized implementation in our in-house CFD&HT code...

$$T_{\Delta t}(s) = sT_{SLAE} + 33sT_{SpMV} + s\left(24 + 3\frac{s-1}{2}\right)T_{axpy} + 10sT_{axty}$$

From stability to efficiency regions

Do stability-region bounded Δt offer the most efficient integration?

Not always. It provides the most efficient integration given a integration scheme.
What if... instead of using a single scheme, a **pool** of schemes is used?

Computational efficiency region

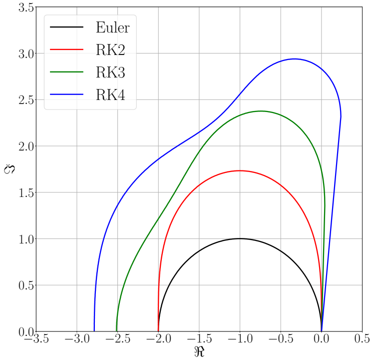
Combination of a set of stability regions normalized by the ratio of wall clock times per iteration between the current scheme and Euler scheme, τ_{Method}.

With the generalized implementation in our in-house CFD&HT code...

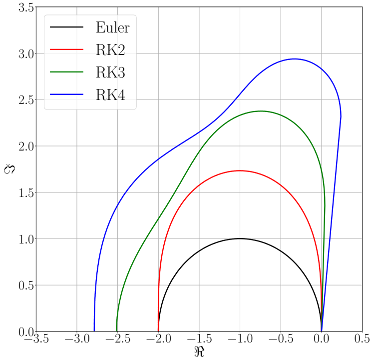
$$T_{\Delta t}(s) = sT_{SLAE} + 33sT_{SpMV} + s\left(24 + 3\frac{s-1}{2}\right)T_{axpy} + 10sT_{axty}$$

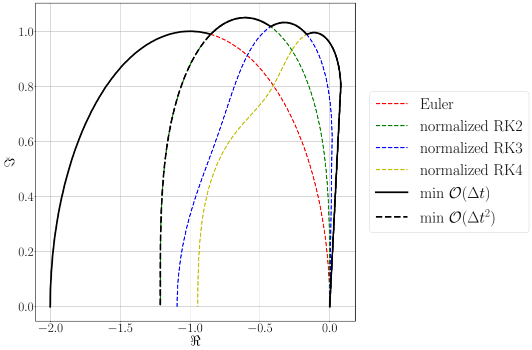
$$\tau_{\mathrm{Method}}(s) = \frac{T_{\Delta t}(s)}{T_{\Delta t}(1)} = s\left(1 + \frac{3}{2}(s-1)\frac{T_{axpy}}{T_{\Delta t}(1)}\right) \approx s$$

From stability to efficiency regions

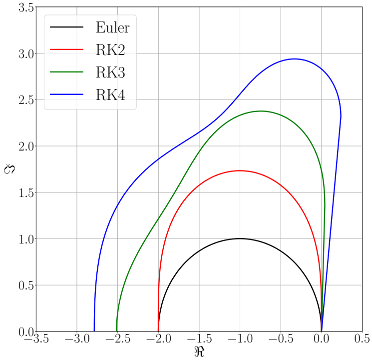


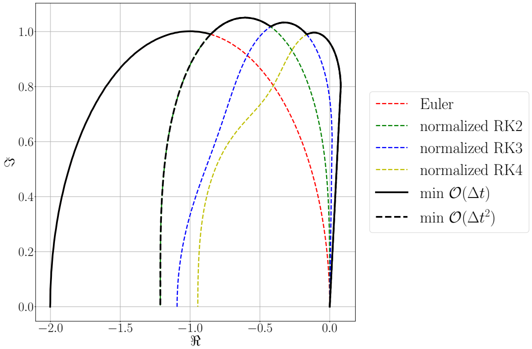
From stability to efficiency regions





From stability to efficiency regions





Additional conditions

Other possible conditions can be set to limit or handicap the proposed schemes in order to filter according to different categories.

From stability to efficiency regions

Algorithm 1: Self-adaptive efficiency time-step scheme

Data: *Pool* of schemes, (Scheme 1,Scheme 2,. . . ,Scheme n),Eigenbounds (λ_C, λ_D), Scale factor $f_{\Delta t}$

Result: Time-step, Δt

$i := 1;$

$i_s := i;$

$\Delta t_s := 0.0;$

while $i \leq n$ **do**

  $\Delta t_{s,i} = \Delta t_s(\text{Scheme}_i, \lambda_C, \lambda_D);$

  if $\Delta t_{s,i} / \tau_{Scheme_i} \geq \Delta t_s$ **then**

    $\Delta t_s = \Delta t_{s,i} / \tau_{\text{Scheme}_i};$

    $i_s := i;$

  end

end

return $\Delta t := f_{\Delta t} \Delta t_s(Scheme_{i_s}, \lambda_C, \lambda_D);$

Numerical experiments

Three-dimensional Taylor-Green vortex

- Initial conditions:

$$u_{x,0} = U_0 \frac{2}{\sqrt{3}} \sin(x)\cos(y)\cos(z)$$

$$u_{y,0} = U_0 \frac{2}{\sqrt{3}} \cos(x)\sin(y)\cos(z)$$

$$u_{z,0} = U_0 \frac{2}{\sqrt{3}} \cos(x)\cos(y)\sin(z)$$

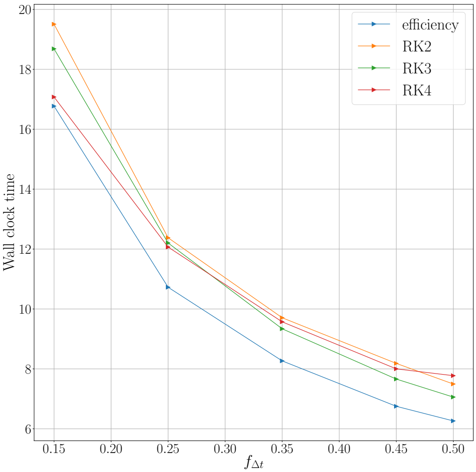
- Run until $t = 20\frac{2\pi}{U_0}$[4]
- $f_{\Delta t} = [0.15, 0.5]$, $\mathrm{Re} = 1500$
- $(32, 64)^3$ grid, cube with a side length of 2π

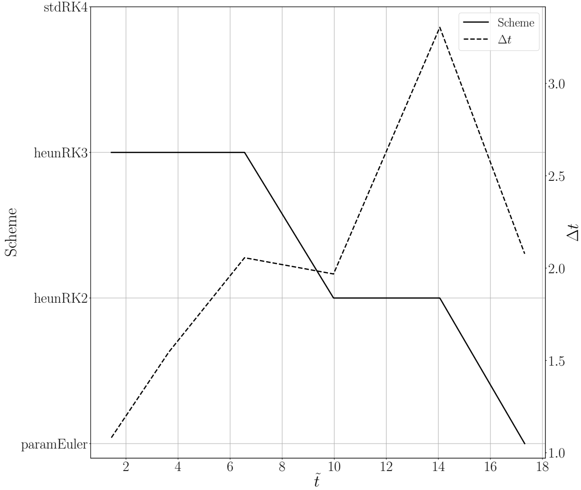
Method	Num. stag. s	Ord. acc. p
Euler	1	1
Heun RK2	2	2
Heun RK3	3	3
Standard RK4	4	4

[4] Capuano, F., Coppola, G., and de Luca, L. (2015), "An efficient time advancing strategy for energy-preserving simulations". Journal of Computational Physics 295, pp. 209-229

Numerical experiments

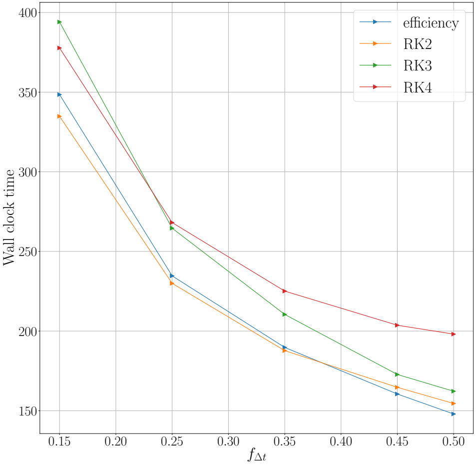
32^3 grid

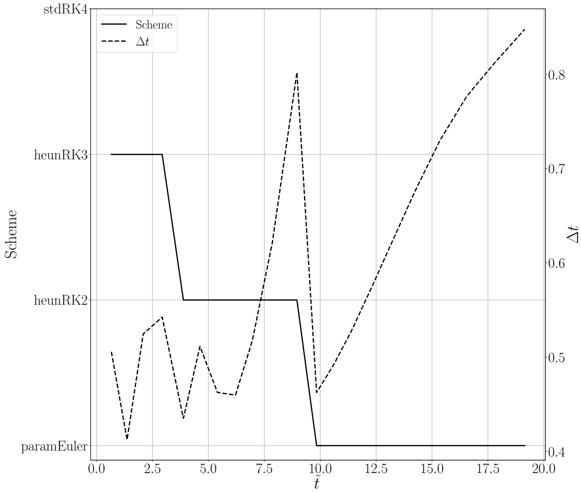




Numerical experiments

64^3 grid





Conclusion

Method

- Method is born as an evolution of self-adaptive strategies:
  - Pool of schemes chosen previous to the start of the simulation
  - Computational efficiency region allows addition of constraints or handicaps to methods to set other characteristics to just performance
  - Computation of eigenbounds might be swapped for a more efficient method

Conclusion

Numerical tests

- Numerical tests have provided improved performance vs the schemes presented on the pool.
- Starts simulation with higher order methods (RK4, RK3) to move to lower order with the evolution of the simulation (Euler).
- Could be tested with higher Re to check if the high order methods are more relevant in the solution.
- Method performance improves when cost per iteration is reduced, or for bigger $f_{\Delta t}$

Conclusion

Future perspectives

Can other properties from Runge-Kutta integration be exploited?

- Computation of Ψ only at the end of the stage, according to Le and Moin[5]:
  - Reduction of accuracy to second order
  - Stability region of the method is preserved

$$\tau_{\text{Method}^*} = 1 + (s-1)(1 - f_{SLAE})$$

[5] Le, H. and Moin, P. (1991), "An improvement of fractional step methods for the incompressible Navier-Stokes equations". Journal of Computational Physics 92, pp. 369-379

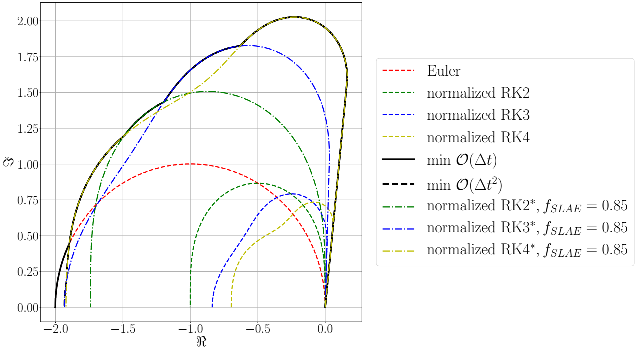
Conclusion

Future perspectives

Can other properties from Runge-Kutta integration be exploited?

- Computation of Ψ only at the end of the stage, according to Le and Moin[5]:
  - Reduction of accuracy to second order
  - Stability region of the method is preserved

$$\tau_{\text{Method}^*} = 1 + (s-1)(1 - f_{SLAE})$$



[5] Le, H. and Moin, P. (1991), "An improvement of fractional step methods for the incompressible Navier-Stokes equations". Journal of Computational Physics 92, pp. 369-379