Many-Stage Optimal Stabilized Runge-Kutta Methods for Hyperbolic Partial Differential Equations

Springer Journal of Scientific Computing, 2024

Daniel Doehring[1]

in collaboration with

M. Torrilhon[1], G. Gassner[2]

[1]Applied & Computational Mathematics
RWTH Aachen University

[2]Division of Mathematics: Numerical Simulation
University of Cologne

15.05.2024

Applied and Computational Mathematics

RWTH AACHEN UNIVERSITY

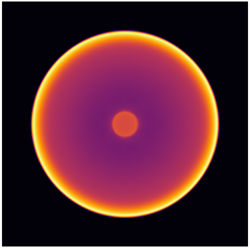
Contents

Contents

Motivation

Many interesting physical phenomena are localized in space, for instance

- Shocks
- Flamefronts
- Fluid-Structure Interactions
- Interference of Waves
- ...

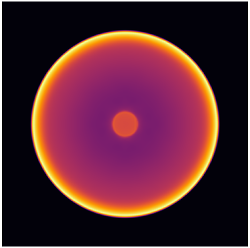


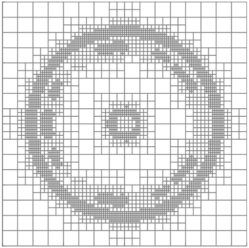
Motivation

Many interesting physical phenomena are localized in space, for instance

- Shocks
- Flamefronts
- Fluid-Structure Interactions
- Interference of Waves
- ...

and typically require a fine resolution in these regions of interest.





Motivation: CFL Condition

- Computational costs of uniformly refined grids are usually too high, especially in 2D/3D

Motivation: CFL Condition

- Computational costs of uniformly refined grids are usually too high, especially in 2D/3D
- Meshes are conventionally only refined in relevant regions

Motivation: CFL Condition

- Computational costs of uniformly refined grids are usually too high, especially in 2D/3D
- Meshes are conventionally only refined in relevant regions

For the 1D system of conservation laws

$$\partial_t \mathbf{u}(t,x) + \partial_x \mathbf{f}\big(\mathbf{u}(t,x)\big) = \mathbf{0} \tag{1}$$

Motivation: CFL Condition

- Computational costs of uniformly refined grids are usually too high, especially in 2D/3D
- Meshes are conventionally only refined in relevant regions

For the 1D system of conservation laws

$$\partial_t \mathbf{u}(t,x) + \partial_x \mathbf{f}\big(\mathbf{u}(t,x)\big) = \mathbf{0} \tag{1}$$

the *CFL Condition*

$$\Delta t \overset{!}{\leq} C_t \cdot C_x \min_{\substack{i=1,\dots,N \\ l=1,\dots,L}} \frac{\Delta x_i}{|\mu_l(\mathbf{u}_i)|} \tag{2}$$

demands a reduction in timestep Δt for reduced mesh-width Δx_i. Here, μ_l are the eigenvalues of the Jacobian $\partial_{\mathbf{u}} \mathbf{f}(\mathbf{u})$.

Motivation: CFL Condition

For an <u>efficient</u> treatment of locally varying mesh widths Δx_i there are now two options:

Motivation: CFL Condition

For an efficient treatment of locally varying mesh widths Δx_i there are now two options:

- Take multiple smaller timesteps where required.
  $\rightarrow$ *Local Time Stepping*[1]

[1] *For instance:* Grote et. al. SISC 2015

Motivation: CFL Condition

For an efficient treatment of locally varying mesh widths Δx_i there are now two options:

- Take multiple smaller timesteps where required.
  $\rightarrow$ *Local Time Stepping*[1]
- Increase the temporal solver stability constant C_t locally.

$$\Delta t \overset{!}{\leq} C_t \cdot C_x \min_{\substack{i=1,\dots,N \\ l=1,\dots,L}} \frac{\Delta x_i}{|\mu_l(\mathbf{u}_i)|} \tag{2}$$

$\rightarrow$ *Multirate*[2]

[1] *For instance:* Grote et. al. SISC 2015

Motivation: CFL Condition

For an <u>efficient</u> treatment of locally varying mesh widths Δx_i there are now two options:

- Take multiple smaller timesteps where required.
  $\rightarrow$ *Local Time Stepping*[1]
- **Increase the temporal solver stability constant C_t locally.**

$$\Delta t \overset{!}{\leq} C_t \cdot C_x \min_{\substack{i=1,\dots,N \\ l=1,\dots,L}} \frac{\Delta x_i}{|\mu_l(\mathbf{u}_i)|} \tag{2}$$

  $\rightarrow$ *Multirate*[2]
- Use different schemes in different partitions of the domain
  $\Rightarrow$ **Partitioned Runge-Kutta Methods**

[1] *For instance:* Grote et. al. SISC 2015

[2] *For instance:* Constantinescu & Sandu J. Sci. Comp. 2007

Contents

Stabilized Explicit Runge-Kutta Methods: Motivation

The conventional treatment of time-dependent PDEs

$$\partial_t \mathbf{u}(t, \mathbf{x}) + \partial_i \mathbf{f}_i\big(\mathbf{u}(t, \mathbf{x}), \nabla \mathbf{u}(t, \mathbf{x})\big) = \mathbf{0} \tag{3}$$

Stabilized Explicit Runge-Kutta Methods: Motivation

The conventional treatment of time-dependent PDEs

$$\partial_t \mathbf{u}(t, \mathbf{x}) + \partial_i \mathbf{f}_i\big(\mathbf{u}(t, \mathbf{x}), \nabla \mathbf{u}(t, \mathbf{x})\big) = \mathbf{0} \tag{3}$$

follows a *Method of Lines* like approach of constructing a *Semi-Discretization*

$$\mathbf{U}(t_0) = \mathbf{U}_0 \tag{4a}$$
$$\mathbf{U}'(t) = \mathbf{F}\big(\mathbf{U}(t)\big) \tag{4b}$$

which is then solved with an ODE integrator $\rightarrow$Runge-Kutta.

Stabilized Explicit Runge-Kutta Methods: RKM Recap

For the Initial Value Problem (4) a Runge-Kutta Method (RKM) computes approximations $\mathbf{U}_n$ through (*Butcher form*)

$$\mathbf{U}_0 = \mathbf{U}(t_0) \tag{5a}$$

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \sum_{i=1}^{S} b_i \mathbf{k}_i \tag{5b}$$

$$\mathbf{k}_i = \mathbf{F}\left(t_n + c_i \Delta t, \mathbf{U}_n + \Delta t \sum_{j=1}^{S} a_{i,j} \mathbf{k}_j\right), \quad i = 1, \dots, S. \tag{5c}$$

where S denotes the number of *stages* $\mathbf{k}_i$.

Stabilized Explicit Runge-Kutta Methods: RKM Recap

For the Initial Value Problem (4) a Runge-Kutta Method (RKM) computes approximations $\mathbf{U}_n$ through (*Butcher form*)

$$\mathbf{U}_0 = \mathbf{U}(t_0) \tag{5a}$$

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \sum_{i=1}^{S} b_i \mathbf{k}_i \tag{5b}$$

$$\mathbf{k}_i = \mathbf{F}\left(t_n + c_i \Delta t, \mathbf{U}_n + \Delta t \sum_{j=1}^{S} a_{i,j} \mathbf{k}_j\right), \quad i = 1, \dots, S. \tag{5c}$$

where S denotes the number of *stages* $\mathbf{k}_i$.

Butcher Tableau:

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

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY

Stabilized Explicit Runge-Kutta Methods: RKM Recap

For the Initial Value Problem (4) a Runge-Kutta Method (RKM) computes approximations $\mathbf{U}_n$ through (*Butcher form*)

$$\mathbf{U}_0 = \mathbf{U}(t_0) \tag{5a}$$

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \sum_{i=1}^{S} b_i \mathbf{k}_i \tag{5b}$$

$$\mathbf{k}_i = \mathbf{F}\left(t_n + c_i \Delta t, \mathbf{U}_n + \Delta t \sum_{j=1}^{S} a_{i,j} \mathbf{k}_j\right), \quad i = 1, \dots, S. \tag{5c}$$

where S denotes the number of *stages* $\mathbf{k}_i$.

Butcher Tableau:

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

For $a_{i,j} = 0, j \geq i$ the method is *explicit*.

Stabilized Explicit Runge-Kutta Methods: Implicit vs. Explicit

- Runge-Kutta Methods come in two flavours: Implicit & Explicit.

Applied and Computational Mathematics

Stabilized Explicit Runge-Kutta Methods: Implicit vs. Explicit

- Runge-Kutta Methods come in two flavours: Implicit & Explicit.
- Implicit methods are usually *A-stable*, i.e., for the test equation

$$u'(t) = \lambda u(t), \quad \mathrm{Re}(\lambda) < 0 \tag{6}$$

  they produce iterates u_n that decrease in magnitude for all timesteps Δt:

$$|u_{n+1}| \leq |u_n| \quad \forall \, \Delta t. \tag{7}$$

Stabilized Explicit Runge-Kutta Methods: Implicit vs. Explicit

- Runge-Kutta Methods come in two flavours: Implicit & Explicit.
- Implicit methods are usually *A-stable*, i.e., for the test equation
$$u'(t) = \lambda u(t), \quad \mathrm{Re}(\lambda) < 0 \tag{6}$$
they produce iterates u_n that decrease in magnitude for all timesteps Δt:
$$|u_{n+1}| \leq |u_n| \quad \forall \, \Delta t. \tag{7}$$
- This comes at the expense that at each timestep a <u>system</u> of (non)linear equations has to be <u>solved</u> (!)

Stabilized Explicit Runge-Kutta Methods: Implicit vs. Explicit

- Runge-Kutta Methods come in two flavours: Implicit & Explicit.
- Implicit methods are usually *A-stable*, i.e., for the test equation

$$u'(t) = \lambda u(t), \quad \mathrm{Re}(\lambda) < 0 \tag{6}$$

  they produce iterates u_n that decrease in magnitude for all timesteps Δt:

$$|u_{n+1}| \leq |u_n| \quad \forall\, \Delta t. \tag{7}$$

- This comes at the expense that at each timestep a system of (non)linear equations has to be solved (!)
- In contrast, explicit methods require only evaluations of $\mathbf{F}$. Drawback: Restriction on timestep (CFL).

Stabilized Explicit Runge-Kutta Methods: Idea

- The central idea of **Stabilized** Explicit Runge-Kutta methods is to use additional stages S to increase the maximum admissible timestep Δt.

Stabilized Explicit Runge-Kutta Methods: Idea

- The central idea of **Stabilized** Explicit Runge-Kutta methods is to use additional stages S to increase the maximum admissible timestep Δt.
- In particular: One settles for a moderate order of accuracy $p = 2, 3, 4$ and uses the additional stages for stability improvement!

Stabilized Explicit Runge-Kutta Methods: Idea

- The central idea of **Stabilized** Explicit Runge-Kutta methods is to use additional stages S to increase the maximum admissible timestep Δt.
- In particular: One settles for a moderate order of accuracy $p = 2, 3, 4$ and uses the additional stages for stability improvement!
- First successful applications for parabolic PDEs already in the 1960s which led to the development of Runge-Kutta Chebyshev (RKC).[3]

[3] *Review:* Abdulle, Explicit stabilized Runge-Kutta methods, Tech. Rep. 2011.

Stabilized Explicit Runge-Kutta Methods: Idea

- The central idea of **Stabilized** Explicit Runge-Kutta methods is to use additional stages S to increase the maximum admissible timestep Δt.
- In particular: One settles for a moderate order of accuracy $p = 2, 3, 4$ and uses the additional stages for stability improvement!
- First successful applications for parabolic PDEs already in the 1960s which led to the development of Runge-Kutta Chebyshev (RKC).[3]
  Key to success: The eigenvalues λ_m of the Jacobian $\partial_{\mathbf{U}} \mathbf{F}(\mathbf{U})$ lie on the negative real axis.

[3] *Review:* Abdulle, Explicit stabilized Runge-Kutta methods, Tech. Rep. 2011.

Stabilized Explicit Runge-Kutta Methods: Idea

- The central idea of **Stabilized** Explicit Runge-Kutta methods is to use additional stages S to increase the maximum admissible timestep Δt.
- In particular: One settles for a moderate order of accuracy $p = 2, 3, 4$ and uses the additional stages for stability improvement!
- First successful applications for parabolic PDEs already in the 1960s which led to the development of Runge-Kutta Chebyshev (RKC).[3]
  Key to success: The eigenvalues λ_m of the Jacobian $\partial_{\mathbf{U}} \mathbf{F}(\mathbf{U})$ lie on the negative real axis.
- This is, however, not the case for right-hand-sides $\mathbf{F}$ corresponding to semi-discretizations of hyperbolic PDEs.

[3] *Review:* Abdulle, Explicit stabilized Runge-Kutta methods, Tech. Rep. 2011.

Stabilized Explicit RKM: Optimization Objective

- One can subsume the application of the RKM (5) to the test equation $u'(t) = \lambda u(t)$ as

$$u_{n+1} = P(\underbrace{\lambda \Delta t}_{=:z}) u_n \tag{8}$$

Stabilized Explicit RKM: Optimization Objective

- One can subsume the application of the RKM (5) to the test equation $u'(t) = \lambda u(t)$ as

$$u_{n+1} = P(\underbrace{\lambda \Delta t}_{=:z})u_n \tag{8}$$

where the *amplification function* $P(z)$ is a
- Rational function for implicit RKMs
- Polynomial for explicit RKMs.

Stabilized Explicit RKM: Optimization Objective

- One can subsume the application of the RKM (5) to the test equation $u'(t) = \lambda u(t)$ as

$$u_{n+1} = P(\underbrace{\lambda \Delta t}_{=:z}) u_n \tag{8}$$

where the *amplification function* $P(z)$ is a
  - Rational function for implicit RKMs
  - Polynomial for explicit RKMs.

- Absolute/linear stability (7) is now asserted when

$$|P(z)| \leq 1 \quad \forall\, z \in \mathcal{S} \subset \mathbb{C}. \tag{9}$$

Stabilized Explicit RKM: Optimization Objective

For the ODE system $\mathbf{U}'(t) = \mathbf{F}\big(\mathbf{U}(t)\big)$ with spectrum $\lambda_m \in \sigma\left(\partial_{\mathbf{U}} \mathbf{F}(\mathbf{U})\right)$ the (linear) stability requirement (9) corresponds to

$$\big|P_S(\Delta t \lambda_m)\big| \leq 1, \quad m = 1, \dots, M \tag{10}$$

Stabilized Explicit RKM: Optimization Objective

For the ODE system $\mathbf{U}'(t) = \mathbf{F}\big(\mathbf{U}(t)\big)$ with spectrum $\lambda_m \in \sigma\left(\partial_{\mathbf{U}} \mathbf{F}(\mathbf{U})\right)$ the (linear) stability requirement (9) corresponds to

$$\big|P_S(\Delta t \lambda_m)\big| \leq 1, \quad m = 1, \dots, M \tag{10}$$

and we search now the optimal stability polynomial P_S of degree S in the sense that the admissible timestep Δt is maximal:

$$\max_{P_S \in \mathcal{P}_S} \Delta t \text{ such that } \big|P_S(\Delta t \lambda_m)\big| \leq 1, \quad m = 1, \dots, M. \tag{11}$$

Stabilized Explicit RKM: Optimal Timestep

Practical results indicate that for a general spectrum σ the maximum stable timesteps scales (asymptotically) linear in the polynomial degree S:

$$\Delta t_{\text{Exp}} := \Delta t_{\text{Ref}} \frac{S}{S_{\text{Ref}}}. \tag{12}$$

Stabilized Explicit RKM: Optimal Timestep

Practical results indicate that for a general spectrum σ the maximum stable timesteps scales (asymptotically) linear in the polynomial degree S:

$$\Delta t_{\text{Exp}} := \Delta t_{\text{Ref}} \frac{S}{S_{\text{Ref}}}. \tag{12}$$

This is not proven, but

- Observed in practice
- Theoretical results indicating this available[4][5][6]

[4] Vichnevetsky Math. Comp. Simul. 1983

[5] Kinnmark & Gray Math. Comput. Simul. 1984

[6] Kinnmark Math. Comput. Simul. 1987

Contents

Optimization of the Stability Polynomial: Convex, but Numerically Unstable

- Write $P_S(z)$ as

$$P_S(z; \boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j z^j, \quad \boldsymbol{\alpha} \in \mathbb{R}^{S+1}, z \in \mathbb{C} \tag{13}$$

Optimization of the Stability Polynomial: Convex, but Numerically Unstable

- Write $P_S(z)$ as

$$P_S(z;\boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j z^j, \quad \boldsymbol{\alpha} \in \mathbb{R}^{S+1}, z \in \mathbb{C} \tag{13}$$

- For fixed Δt

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^{S+1}} \left| P_S(\Delta t \lambda_m;\boldsymbol{\alpha}) \right|, \quad m = 1, \dots, \widetilde{M}. \tag{14}$$

is a **convex** optimization problem[7] $\rightarrow$Unique solution for $\boldsymbol{\alpha}$, many solvers available.

[7] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Convex, but Numerically Unstable

- Write $P_S(z)$ as

$$P_S(z;\boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j z^j, \quad \boldsymbol{\alpha} \in \mathbb{R}^{S+1}, z \in \mathbb{C} \tag{13}$$

- For fixed Δt

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{S+1}} \left| P_S(\Delta t \lambda_m; \boldsymbol{\alpha}) \right|, \quad m = 1, \dots, \widetilde{M}. \tag{14}$$

is a **convex** optimization problem[7] →Unique solution for $\boldsymbol{\alpha}$, many solvers available.
Compute Δt through bisection (outer optimization)

[7] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Convex, but Numerically Unstable

- Write $P_S(z)$ as

$$P_S(z;\boldsymbol{\alpha}) = \sum_{j=0}^{S} \alpha_j z^j, \quad \boldsymbol{\alpha} \in \mathbb{R}^{S+1}, z \in \mathbb{C} \tag{13}$$

- For fixed Δt

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{S+1}} \left| P_S(\Delta t \lambda_m; \boldsymbol{\alpha}) \right|, \quad m = 1, \dots, \widetilde{M}. \tag{14}$$

  is a **convex** optimization problem[7] $\rightarrow$Unique solution for $\boldsymbol{\alpha}$, many solvers available.
  Compute Δt through bisection (outer optimization)

- Issue: $\alpha_j \sim \frac{1}{j!}$ $\rightarrow$Limited to 16-20 stages.

[7] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Nonconvex, but Numerically Stable

For an order p (linearly) consistent RKM the coefficients α_j of the corresponding stability polynomial need to match

$$\alpha_j \overset{!}{=} \frac{1}{j!}, \quad j = 0, \dots, p. \tag{15}$$

Optimization of the Stability Polynomial: Nonconvex, but Numerically Stable

For an order p (linearly) consistent RKM the coefficients α_j of the corresponding stability polynomial need to match

$$\alpha_j \overset{!}{=} \frac{1}{j!}, \quad j = 0, \dots, p. \tag{15}$$

Consequently, any first order accurate method has a stability polynomial of form

$$P_S(z) = 1 + z\widetilde{P}_{S-1}(z) \tag{16}$$

with *Lower Degree Polynomial*

$$\widetilde{P}_{S-1}(z; \widetilde{\mathbf{r}}) = \prod_{j=1}^{S-1} \left(1 - \frac{z}{\widetilde{r}_j}\right). \tag{17}$$

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY

Optimization of the Stability Polynomial: Nonconvex, but Numerically Stable

$$P_S(z;\widetilde{\mathbf{r}}) = 1 + z\widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}), \quad \widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}) = \prod_{j=1}^{S-1}\left(1 - \frac{z}{\widetilde{r}_j}\right)$$

- The roots $\widetilde{\mathbf{r}}$ of the Lower Degree Polynomial $\widetilde{P}_{S-1}$ are called *Pseudo Extrema* and will be the optimization variables.

Optimization of the Stability Polynomial: Nonconvex, but Numerically Stable

$$P_S(z;\widetilde{\mathbf{r}}) = 1 + z\widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}), \quad \widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}) = \prod_{j=1}^{S-1}\left(1 - \frac{z}{\widetilde{r}_j}\right)$$

- The roots $\widetilde{\mathbf{r}}$ of the Lower Degree Polynomial $\widetilde{P}_{S-1}$ are called *Pseudo Extrema* and will be the optimization variables.
- Advantage of the parametrization of the polynomial in roots: Roots $\widetilde{\mathbf{r}}$ are of **same order as the eigenvalues, regardless of the stage count S!**

Optimization of the Stability Polynomial: Nonconvex, but Numerically Stable

$$P_S(z;\widetilde{\mathbf{r}}) = 1 + z\widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}), \quad \widetilde{P}_{S-1}(z;\widetilde{\mathbf{r}}) = \prod_{j=1}^{S-1}\left(1 - \frac{z}{\widetilde{r}_j}\right)$$

- The roots $\widetilde{\mathbf{r}}$ of the Lower Degree Polynomial $\widetilde{P}_{S-1}$ are called *Pseudo Extrema* and will be the optimization variables.
- Advantage of the parametrization of the polynomial in roots: Roots $\widetilde{\mathbf{r}}$ are of **same order as the eigenvalues, regardless of the stage count S!**
- Issue: Highly nonlinear optimization problem

$$\max_{\widetilde{\mathbf{r}} \in \mathbb{C}^{S-1}} \Delta t \text{ such that } \left|P_S(\Delta t \lambda_m;\widetilde{\mathbf{r}})\right| \leq 1, \quad m = 1, \dots, \widetilde{M}. \tag{18}$$

Optimization of the Stability Polynomial: Initial Guess

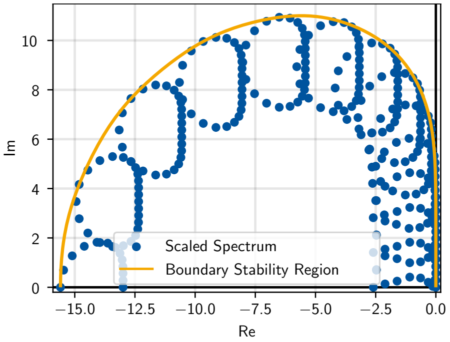
As a consequence, we need a **good initial guess** to have reasonable hope to find the global optimum.

Optimization of the Stability Polynomial: Initial Guess

As a consequence, we need a **good initial guess** to have reasonable hope to find the global optimum.

Central observation: The Pseudo Extrema $\widetilde{\mathbf{r}}$ lie necessarily on the stability boundary:

$$P_S(\widetilde{r}_j; \widetilde{\mathbf{r}}) = 1 + \widetilde{r}_j \underbrace{\widetilde{P}_{S-1}(\widetilde{r}_j; \widetilde{\mathbf{r}})}_{=0} = 1 \quad \forall\, j = 1, \dots, S-1 \tag{19}$$

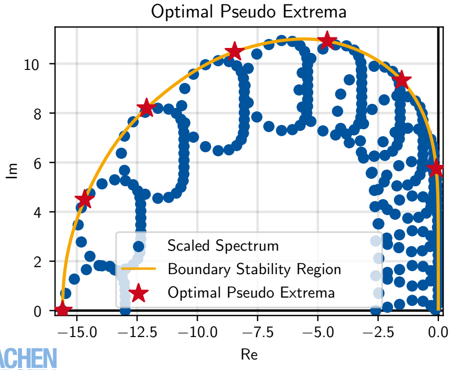


Optimization of the Stability Polynomial: Initial Guess

As a consequence, we need a **good initial guess** to have reasonable hope to find the global optimum.

Central observation: The Pseudo Extrema $\widetilde{\mathbf{r}}$ lie necessarily on the stability boundary:

$$P_S(\widetilde{r}_j; \widetilde{\mathbf{r}}) = 1 + \widetilde{r}_j \underbrace{\widetilde{P}_{S-1}(\widetilde{r}_j; \widetilde{\mathbf{r}})}_{=0} = 1 \quad \forall\, j = 1, \dots, S-1 \tag{19}$$



Optimization of the Stability Polynomial: Initial Guess

Question: Where do we place the pseudo extrema initially on the stability boundary?

Optimization of the Stability Polynomial: Initial Guess

Question: Where do we place the pseudo extrema initially on the stability boundary?

Consider proven optimal stability polynomials:

- Parabolic Spectra: Shifted Chebyshev polynomials

$$P_{S,1}(z) = T_S\left(1 + \frac{z}{S^2}\right) \tag{20}$$

Optimization of the Stability Polynomial: Initial Guess

Question: Where do we place the pseudo extrema initially on the stability boundary?

Consider proven optimal stability polynomials:

- Parabolic Spectra: Shifted Chebyshev polynomials

$$P_{S,1}(z) = T_S\left(1 + \frac{z}{S^2}\right) \tag{20}$$

- Disks/Circular Spectra:
  1. First order:

$$P_{S,1} = \left(1 + \frac{z}{S}\right)^S \tag{21}$$

Optimization of the Stability Polynomial: Initial Guess

Question: Where do we place the pseudo extrema initially on the stability boundary?

Consider proven optimal stability polynomials:

- Parabolic Spectra: Shifted Chebyshev polynomials

$$P_{S,1}(z) = T_S\left(1 + \frac{z}{S^2}\right) \tag{20}$$

- Disks/Circular Spectra:
  1. First order:

$$P_{S,1} = \left(1 + \frac{z}{S}\right)^S \tag{21}$$

  2. Second order:

$$P_{S,2} = \frac{S-1}{S}\left(1 + \frac{z}{S-1}\right)^S + \frac{1}{S} \tag{22}$$

Optimization of the Stability Polynomial: Pseudo Extrema for Optimal Polynomials

For all the previous cases one can immediately show that the pseudo extrema $\widetilde{r}_j$ are given by the positive *Chebyshev Extreme Points*, i.e., the points

$$x_j = \cos\left(\frac{2\pi j}{S} - 1\right), j = 0, \dots, \begin{cases} S/2, & S \text{ even} \\ (S-1)/2 & S \text{ odd} \end{cases} \tag{23}$$

such that

$$T_S(x_j) = 1 \tag{24}$$

when projected onto the spectrum.

Optimization of the Stability Polynomial: Pseudo Extrema for Optimal Polynomials

For all the previous cases one can immediately show that the pseudo extrema $\widetilde{r}_j$ are given by the positive *Chebyshev Extreme Points*, i.e., the points

$$x_j = \cos\left(\frac{2\pi j}{S} - 1\right), j = 0, \dots, \begin{cases} S/2, & S \text{ even} \\ (S-1)/2 & S \text{ odd} \end{cases} \tag{23}$$

such that

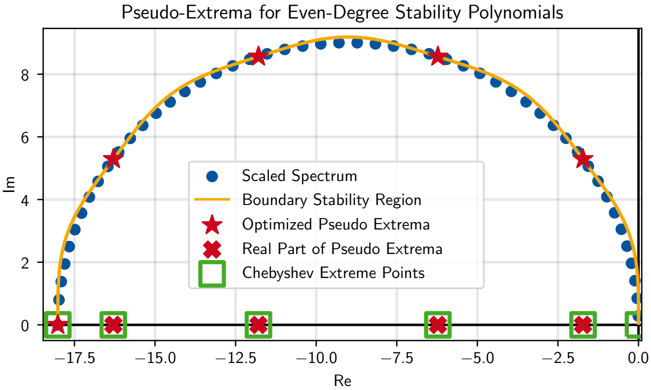
$$T_S(x_j) = 1 \tag{24}$$

when projected onto the spectrum. For disks with radius R:

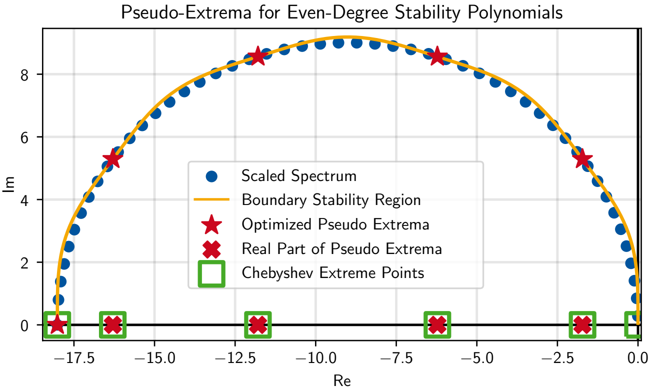
$$\widetilde{r}_j = x_j \pm \sqrt{R^2 - (x_j + R)^2} \tag{25}$$

$$= R\left[\cos\left(\frac{2\pi j}{S} - 1\right) \pm i \sin\left(\frac{2\pi j}{S}\right)\right] \tag{26}$$

Optimization of the Stability Polynomial: Pseudo Extrema for Optimal Polynomials



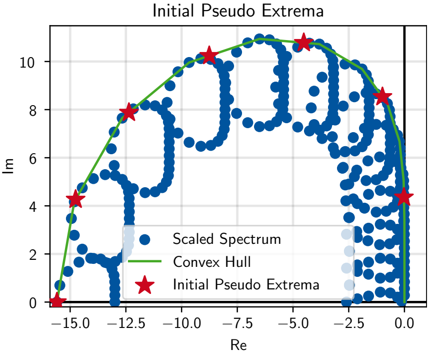
Optimization of the Stability Polynomial: Pseudo Extrema for Optimal Polynomials



- Significance of the Chebyshev Extreme Points: Partition the circle into segments with equal arc length

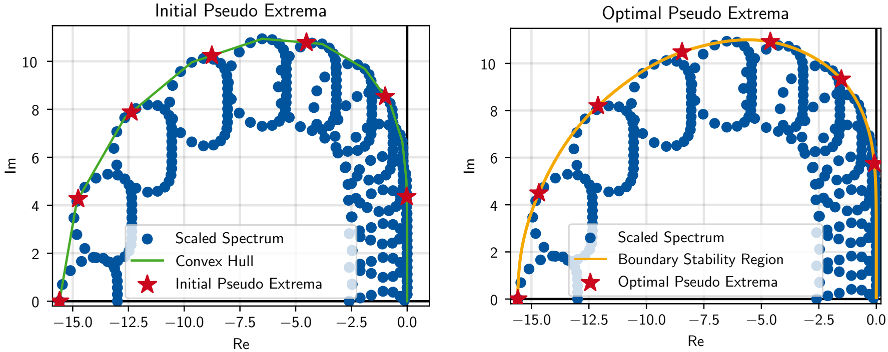
Optimization of the Stability Polynomial: Initial Guess

- Generalize this idea to other spectra: Place pseudo extrema initially with equal arc length on the convex hull of the spectrum



Optimization of the Stability Polynomial: Initial Guess

- Generalize this idea to other spectra: Place pseudo extrema initially with equal arc length on the convex hull of the spectrum



- Deviation from initial value: Courtesy of the optimizer!

Optimization of the Stability Polynomial: Algorithm & Implementation

Algorithm:

1. Find reference timestep for small number of stages S[8]

[8]Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Algorithm & Implementation

Algorithm:

1. Find reference timestep for small number of stages S[8]
2. Distribute Pseudo Extrema with equal arc length on spectrum enclosing, scaled hull

[8] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Algorithm & Implementation

Algorithm:

1. Find reference timestep for small number of stages S[8]
2. Distribute Pseudo Extrema with equal arc length on spectrum enclosing, scaled hull
3. Optimize real part of Pseudo Extrema

[8] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Algorithm & Implementation

Algorithm:

1. Find reference timestep for small number of stages S[8]
2. Distribute Pseudo Extrema with equal arc length on spectrum enclosing, scaled hull
3. Optimize real part of Pseudo Extrema
4. Correct imaginary part of Pseudo Extrema

[8] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Algorithm & Implementation

Algorithm:

1. Find reference timestep for small number of stages S[8]
2. Distribute Pseudo Extrema with equal arc length on spectrum enclosing, scaled hull
3. Optimize real part of Pseudo Extrema
4. Correct imaginary part of Pseudo Extrema

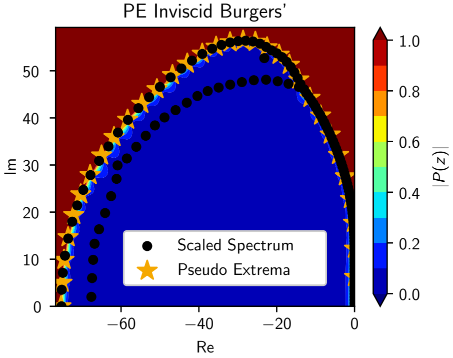
Software:

- Optimizer: `Ipopt: Interioir Point OPTimizer`
- Derivatives: Via `dco/c++`

[8] Ketcheson & Ahmadia, Comm. App. Math. Comp. Sci., 2013

Optimization of the Stability Polynomial: Results

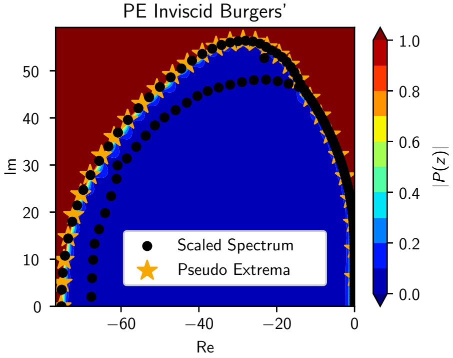
- Generate stability polynomials with degrees $S > 100$ for $p = 2, 3$



Optimization of the Stability Polynomial: Results

- Generate stability polynomials with degrees $S > 100$ for $p = 2, 3$
- For moderate number (100-300) of constraining eigenvalues λ very fast: $P_{128,2}$ for spectrum on the right generated in ~ 12.6s.

S	2^{nd} Order	3^{rd} Order
16	15.346	15.076
32	0.879	1.183
64	2.869	2.276
128	8.845	8.549



- $S = 16$: Generated with approach from Ketcheson & Ahmadia

Contents

Many Stage Runge-Kutta Methods: Construction

- Task: Stability Polynomial $\rightarrow$RKM

Many Stage Runge-Kutta Methods: Construction

- Task: Stability Polynomial $\to$RKM
- Inspired from many stage Runge-Kutta Chebyshev: Directly **Read-off**:

$$
\begin{aligned}
P_S(z) &= 1 + z \prod_{j=1}^{S-1} \left(1 - \frac{z}{\widetilde{r}_j}\right) \\
&= \underbrace{1 + z \left[\underbrace{\prod_{j=1}^{S_{\text{Real}}} \left(1 - \frac{z}{\widetilde{r}_j}\right)}_{} \cdot \underbrace{\prod_{j=1}^{S_{\text{Complex}}/2} \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right)}_{} \right]}_{}
\end{aligned}
$$

Many Stage Runge-Kutta Methods: Construction

- Task: Stability Polynomial $\rightarrow$RKM
- Inspired from many stage Runge-Kutta Chebyshev: Directly **Read-off**:

$$
\begin{aligned}
P_S(z) &= 1 + z \prod_{j=1}^{S-1} \left(1 - \frac{z}{\widetilde{r}_j}\right) \\
&= \underbrace{1 + z \left[\underbrace{\prod_{j=1}^{S_{\text{Real}}} \left(1 - \frac{z}{\widetilde{r}_j}\right)}_{\text{Forward Euler}} \cdot \underbrace{\prod_{j=1}^{S_{\text{Complex}}/2} \left(1 - \frac{z}{\widetilde{r}_j}\right) \left(1 - \frac{z}{\widetilde{r}_j^\star}\right)} \right]}
\end{aligned}
$$

Many Stage Runge-Kutta Methods: Construction

- Task: Stability Polynomial $\to$RKM
- Inspired from many stage Runge-Kutta Chebyshev: Directly **Read-off**:

$$P_S(z) = 1 + z \prod_{j=1}^{S-1} \left(1 - \frac{z}{\widetilde{r}_j}\right)$$

$$= \underbrace{1 + z \left[\underbrace{\prod_{j=1}^{S_{\text{Real}}} \left(1 - \frac{z}{\widetilde{r}_j}\right)}_{\text{Forward Euler}} \cdot \underbrace{\prod_{j=1}^{S_{\text{Complex}}/2} \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right)}_{\text{2-Stage Submethod}} \right]}$$

Many Stage Runge-Kutta Methods: Construction

- Task: Stability Polynomial →RKM
- Inspired from many stage Runge-Kutta Chebyshev: Directly **Read-off**:

$$
\begin{aligned}
P_S(z) &= 1 + z \prod_{j=1}^{S-1} \left(1 - \frac{z}{\widetilde{r}_j}\right) \\
&= \underbrace{1 + z \left[\underbrace{\prod_{j=1}^{S_{\text{Real}}} \left(1 - \frac{z}{\widetilde{r}_j}\right)}_{\text{Forward Euler}} \cdot \underbrace{\prod_{j=1}^{S_{\text{Complex}}/2} \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^{\star}}\right)}_{\text{2-Stage Submethod}} \right]}_{\text{Forward Euler}}
\end{aligned}
$$

Many Stage Runge-Kutta Methods: Internal Stability

Only overall, unperturbed update step $\mathbf{U}_n \to \mathbf{U}_{n+1}$ is stable.

Many Stage Runge-Kutta Methods: Internal Stability

Only overall, unperturbed update step $\mathbf{U}_n \to \mathbf{U}_{n+1}$ is stable.

- Employ for analysis of internal stability *Modified Shu-Osher Form*[9] of the RKM:

$$\mathbf{Y}_j \coloneqq v_j \mathbf{U}_n + \sum_{k=1}^{j-1} \Big(\alpha_{j,k} \mathbf{Y}_k + \Delta t \beta_{j,k} \mathbf{F}(\mathbf{Y}_k) \Big) \tag{27a}$$

$$\mathbf{U}_{n+1} \coloneqq \mathbf{Y}_{S+1} \tag{27b}$$

[9] Ferracina & Spijker, Math. Comp., 2005

Many Stage Runge-Kutta Methods: Internal Stability

Only overall, unperturbed update step $\mathbf{U}_n \to \mathbf{U}_{n+1}$ is stable.

- Employ for analysis of internal stability *Modified Shu-Osher Form*[9] of the RKM:

$$\mathbf{Y}_j := v_j \mathbf{U}_n + \sum_{k=1}^{j-1} \left(\alpha_{j,k} \mathbf{Y}_k + \Delta t \beta_{j,k} \mathbf{F}(\mathbf{Y}_k) \right) \tag{27a}$$

$$\mathbf{U}_{n+1} := \mathbf{Y}_{S+1} \tag{27b}$$

with Round-off errors $\mathbf{e}$

$$\mathbf{Y}_j^{(e)} := v_j \mathbf{U}_n^{(e)} + \sum_{k=1}^{j-1} \left(\alpha_{j,k} \mathbf{Y}_k^{(e)} + \Delta t \beta_{j,k} \mathbf{F}\left(\mathbf{Y}_k^{(e)} \right) \right) + \mathbf{e}_j \tag{28a}$$

$$\mathbf{U}_{n+1}^{(e)} := \mathbf{Y}_{S+1}^{(e)} \tag{28b}$$

[9] Ferracina & Spijker, Math. Comp., 2005

Many Stage Runge-Kutta Methods: Internal Stability

- Round-off errors may get significantly amplified:[10]

$$
\begin{aligned}
\left\| \mathbf{U}^{(e)}_{n+1} - \mathbf{U}(t_{n+1}) \right\| \leq & \underbrace{\left\| \mathbf{U}^{(e)}_{n} - \mathbf{U}(t_{n}) \right\| + \mathcal{O}(\Delta t^{p+1})}_{\text{Usual estimate}} \\
& + \underbrace{\mathcal{M}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \|\mathbf{e}_i\|_\infty}_{\text{Round-off Error Amplification}}
\end{aligned}
\tag{29}
$$

[10] Ketcheson et al., SINUM, 2014

Many Stage Runge-Kutta Methods: Internal Stability

- Round-off errors may get significantly amplified:[10]

$$
\begin{aligned}
\left\| \mathbf{U}^{(e)}_{n+1} - \mathbf{U}(t_{n+1}) \right\| \leq & \underbrace{\left\| \mathbf{U}^{(e)}_{n} - \mathbf{U}(t_{n}) \right\| + \mathcal{O}(\Delta t^{p+1})}_{\text{Usual estimate}} \\
+ & \underbrace{\mathcal{M}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \|\mathbf{e}_i\|_\infty}_{\text{Round-off Error Amplification}}
\end{aligned} \tag{29}
$$

- Design RKM to minimize error amplification factor $\mathcal{M}$:

[10] Ketcheson et al., SINUM, 2014

Many Stage Runge-Kutta Methods: Internal Stability

- Round-off errors may get significantly amplified:[10]

$$
\begin{aligned}
\left\| \mathbf{U}^{(e)}_{n+1} - \mathbf{U}(t_{n+1}) \right\| \leq & \underbrace{\left\| \mathbf{U}^{(e)}_{n} - \mathbf{U}(t_{n}) \right\| + \mathcal{O}(\Delta t^{p+1})}_{\text{Usual estimate}} \\
+ & \underbrace{\mathcal{M}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \|\mathbf{e}_i\|_\infty}_{\text{Round-off Error Amplification}}
\end{aligned} \tag{29}
$$

- Design RKM to minimize error amplification factor $\mathcal{M}$:
- Employ generalization of *Lebedev's Idea*:[11]
  Group pseudo extrema with $\text{Re}\,(\widetilde{r}_j) \leq -0.5$ together with the ones with largest $\text{Re}\,(\widetilde{r}_k)$

[10] Ketcheson et al., SINUM, 2014

[11] Lebedev, Russ. J. Numer. Anal. Math. Model., 1989

Many Stage Runge-Kutta Methods: Internal Stability

- Round-off errors may get significantly amplified:[10]

$$\begin{aligned}\left\|\mathbf{U}^{(e)}_{n+1} - \mathbf{U}(t_{n+1})\right\| \leq & \underbrace{\left\|\mathbf{U}^{(e)}_{n} - \mathbf{U}(t_{n})\right\| + \mathcal{O}(\Delta t^{p+1})}_{\text{Usual estimate}} \\ & + \underbrace{\mathcal{M}(\boldsymbol{\alpha}, \boldsymbol{\beta}) \|\mathbf{e}_i\|_\infty}_{\text{Round-off Error Amplification}}\end{aligned} \tag{29}$$

- Design RKM to minimize error amplification factor $\mathcal{M}$:
- Employ generalization of *Lebedev's Idea*:[11]
  Group pseudo extrema with $\mathrm{Re}\left(\widetilde{r}_j\right) \leq -0.5$ together with the ones with largest $\mathrm{Re}\left(\widetilde{r}_k\right)$
- This avoids large $\beta_{j,k}$

[10] Ketcheson et al., SINUM, 2014

[11] Lebedev, Russ. J. Numer. Anal. Math. Model., 1989

Many Stage Runge-Kutta Methods: Internal Stability

- This $j \leftrightarrow k$ grouping leads to intermediate 4'th degree polynomials

$$Q_4^{j,k}(z) = \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right) \cdot \left(1 - \frac{z}{\widetilde{r}_k}\right)\left(1 - \frac{z}{\widetilde{r}_k^\star}\right) \quad (30)$$

which are represented using four instances of the $2N$ storage scheme

$$\begin{aligned} \mathbf{U}_{l+1} :=& \alpha_{l+1,l-1}\mathbf{U}_{l-1} + \alpha_{l+1,l}\mathbf{U}_l \\ &+ \Delta t \left[\beta_{l+1,l-1}\mathbf{F}(\mathbf{U}_{l-1}) + \beta_{l+1,l}\mathbf{F}(\mathbf{U}_l)\right] \end{aligned} \quad (31)$$

Many Stage Runge-Kutta Methods: Internal Stability

- This $j \leftrightarrow k$ grouping leads to intermediate 4'th degree polynomials

$$Q_4^{j,k}(z) = \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right) \cdot \left(1 - \frac{z}{\widetilde{r}_k}\right)\left(1 - \frac{z}{\widetilde{r}_k^\star}\right) \quad (30)$$

  which are represented using four instances of the $2N$ storage scheme

$$\begin{aligned}\mathbf{U}_{l+1} :=& \alpha_{l+1,l-1}\mathbf{U}_{l-1} + \alpha_{l+1,l}\mathbf{U}_l \\ &+ \Delta t \left[\beta_{l+1,l-1}\mathbf{F}(\mathbf{U}_{l-1}) + \beta_{l+1,l}\mathbf{F}(\mathbf{U}_l)\right]\end{aligned} \quad (31)$$

- 12 parameters subject to 4 constraints
  $\Rightarrow$ Use additional freedom to minimize $|\boldsymbol{\beta}|$

Applied and Computational Mathematics | RWTH AACHEN UNIVERSITY

Many Stage Runge-Kutta Methods: Internal Stability

- This $j \leftrightarrow k$ grouping leads to intermediate 4'th degree polynomials

$$Q_4^{j,k}(z) = \left(1 - \frac{z}{\widetilde{r}_j}\right)\left(1 - \frac{z}{\widetilde{r}_j^\star}\right) \cdot \left(1 - \frac{z}{\widetilde{r}_k}\right)\left(1 - \frac{z}{\widetilde{r}_k^\star}\right) \quad (30)$$

which are represented using four instances of the $2N$ storage scheme

$$\begin{aligned}\mathbf{U}_{l+1} :=& \alpha_{l+1,l-1}\mathbf{U}_{l-1} + \alpha_{l+1,l}\mathbf{U}_l \\ &+ \Delta t \left[\beta_{l+1,l-1}\mathbf{F}(\mathbf{U}_{l-1}) + \beta_{l+1,l}\mathbf{F}(\mathbf{U}_l)\right]\end{aligned} \quad (31)$$

- 12 parameters subject to 4 constraints
  $\Rightarrow$ Use additional freedom to minimize $|\boldsymbol{\beta}|$
- Reduction of $\mathcal{M}$ from $\mathcal{O}(10^7)$ to $\mathcal{O}(10^{-6})$!

Many Stage Runge-Kutta Methods:
Results for linear systems: 1D Advection Equation

- 1D Advection Equation

$$\partial_t u + a\partial_x u = 0 \tag{32a}$$

$$u_0(x) = e^{-10x^2} \tag{32b}$$

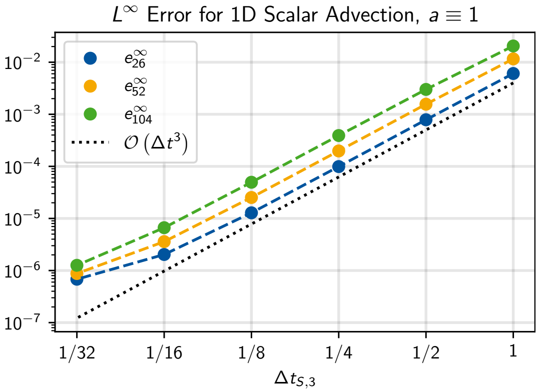
- Construct (linear) third–order schemes

S	Δt	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^4
26	$5.86 \cdot 10^{-2}$	1708	$3.94 \cdot 10^{-12}$	$1.28 \cdot 10^{-5}$
52	$1.17 \cdot 10^{-1}$	854	$1.95 \cdot 10^{-10}$	$1.87 \cdot 10^{-4}$
104	$2.34 \cdot 10^{-1}$	427	$1.63 \cdot 10^{-6}$	$3.00 \cdot 10^{-3}$

Many Stage Runge-Kutta Methods: Results for linear systems: 1D Advection Equation

- Report error in terms of L^∞ error

$$e_S^\infty := \| u^{(h,S)}(t_f, x) - u(t_f, x) \|_\infty \tag{33}$$



Many Stage Runge-Kutta Methods: Results for linear systems: 2D Linearized Euler Equations

- 2D Linearized Euler Equations

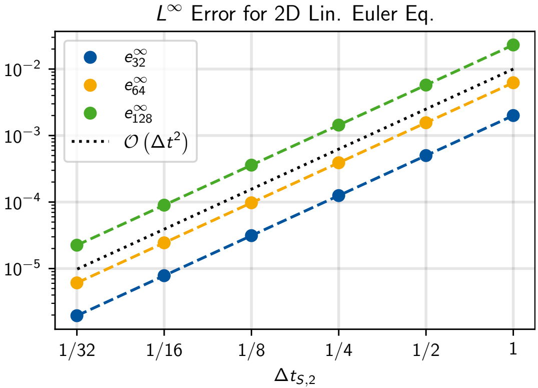
$$\partial_t \begin{pmatrix} \rho' \\ u' \\ v' \\ p' \end{pmatrix} + \partial_x \begin{pmatrix} \bar\rho u' + \bar u \rho' \\ \bar u u' + \frac{p'}{\bar\rho} \\ \bar u v' \\ \bar u p' + c^2 \bar\rho u' \end{pmatrix} + \partial_y \begin{pmatrix} \bar\rho v' + \bar v \rho' \\ \bar v u' \\ \bar v v' + \frac{p'}{\bar\rho} \\ \bar v p' + c^2 \bar\rho v' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

S	Δt	CFL	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^3
32	$1.13 \cdot 10^{-2}$	1.0	928	$1.91 \cdot 10^{-11}$	$1.45 \cdot 10^{-6}$
64	$2.17 \cdot 10^{-2}$	0.96	484	$6.30 \cdot 10^{-9}$	$1.03 \cdot 10^{-5}$
128	$3.67 \cdot 10^{-2}$	0.81	287	$5.63 \cdot 10^{-5}$	$4.94 \cdot 10^{-5}$

Many Stage Runge-Kutta Methods:
Results for linear systems: 2D Linearized Euler Equations

- Report error in terms of density L^∞ error

$$e_S^\infty := \|\rho^{(h,S)}(t_f, x) - \rho(t_f, x)\|_\infty \qquad (34)$$



Many Stage Runge-Kutta Methods:
Results for nonlinear systems: Burgers' Equation

- Inviscid Burgers' Equation

$$\partial_t + \frac{1}{2}\partial_x u^2 = s(x,t) \tag{35a}$$

$$u(x,t) = 2 + \sin\big(2\pi(x-t)\big) \tag{35b}$$

- Consider also error (increase) in *Total Variation*

$$e_{\text{TV}} := \|\mathbf{u}^{(h,S)}(t_f)\|_{\text{TV}} - \|\mathbf{u}^{(h,S)}(t_0)\|_{\text{TV}} \tag{36}$$

$$\|\mathbf{u}(t)\|_{\text{TV}} := \sum_j |u_{j+1}(t) - u_j(t)| \tag{37}$$

Applied and Computational Mathematics

RWTH AACHEN UNIVERSITY

Many Stage Runge-Kutta Methods:
Results for nonlinear systems: Burgers' Equation

S	CFL	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^3	e_{TV}
28	0.72	1363	$1.41 \cdot 10^{-12}$	$4.95 \cdot 10^{-8}$	$1.30 \cdot 10^{-2}$
56	0.69	711	$1.39 \cdot 10^{-10}$	$3.48 \cdot 10^{-7}$	$5.18 \cdot 10^{-2}$
112	0.28	876	$4.79 \cdot 10^{-10}$	$1.86 \cdot 10^{-7}$	$6.09 \cdot 10^{-3}$

Many Stage Runge-Kutta Methods: Results for nonlinear systems: Burgers' Equation

S	CFL	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^3	e_{TV}
28	0.72	1363	$1.41 \cdot 10^{-12}$	$4.95 \cdot 10^{-8}$	$1.30 \cdot 10^{-2}$
56	0.69	711	$1.39 \cdot 10^{-10}$	$3.48 \cdot 10^{-7}$	$5.18 \cdot 10^{-2}$
112	0.28	876	$4.79 \cdot 10^{-10}$	$1.86 \cdot 10^{-7}$	$6.09 \cdot 10^{-3}$

- Attribute reduction in CFL due to lack of SSP property

Many Stage Runge-Kutta Methods: Results for nonlinear systems: Burgers' Equation

S	CFL	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^3	e_{TV}
28	0.72	1363	$1.41 \cdot 10^{-12}$	$4.95 \cdot 10^{-8}$	$1.30 \cdot 10^{-2}$
56	0.69	711	$1.39 \cdot 10^{-10}$	$3.48 \cdot 10^{-7}$	$5.18 \cdot 10^{-2}$
112	0.28	876	$4.79 \cdot 10^{-10}$	$1.86 \cdot 10^{-7}$	$6.09 \cdot 10^{-3}$

- Attribute reduction in CFL due to lack of SSP property
- Reason: Compare to very many stage SSP methods[12] with higher $\mathcal{M}$: For these, no reduction in timestep is needed!

[12] Ketcheson, SISC, 2013

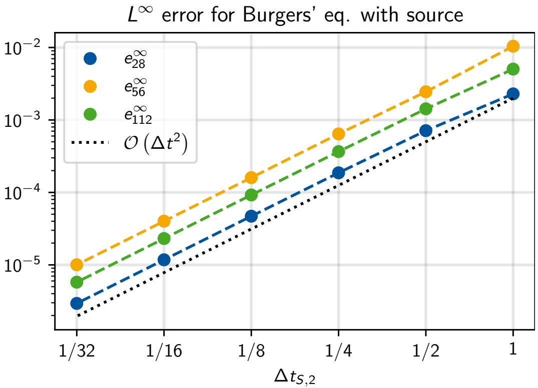
Many Stage Runge-Kutta Methods: Results for nonlinear systems: Burgers' Equation

S	CFL	N_t	$\mathcal{M} \cdot 10^{-15}$	Δt^3	e_{TV}
28	0.72	1363	$1.41 \cdot 10^{-12}$	$4.95 \cdot 10^{-8}$	$1.30 \cdot 10^{-2}$
56	0.69	711	$1.39 \cdot 10^{-10}$	$3.48 \cdot 10^{-7}$	$5.18 \cdot 10^{-2}$
112	0.28	876	$4.79 \cdot 10^{-10}$	$1.86 \cdot 10^{-7}$	$6.09 \cdot 10^{-3}$

- Attribute reduction in CFL due to lack of SSP property
- Reason: Compare to very many stage SSP methods[12] with higher $\mathcal{M}$: For these, no reduction in timestep is needed!
- Difficulty for nonlinear problems: Small changes in u change the spectrum σ

[12] Ketcheson, SISC, 2013

Many Stage Runge-Kutta Methods: Results for nonlinear systems: Burgers' Equation



Many Stage Runge-Kutta Methods: Results for nonlinear Systems: 2D Compressible Euler Eq.

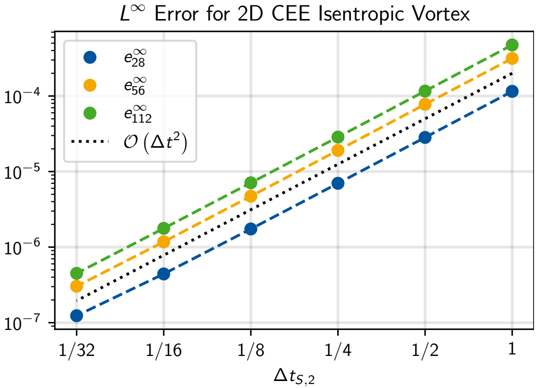
- Testcase: Isentropic Vortex Advection

S	Δt	CFL	N_t	$\widetilde{\mathcal{M}} \cdot 10^{-15}$	Δt^3
30	$5.92 \cdot 10^{-2}$	0.79	338	$1.80 \cdot 10^{-12}$	$2.08 \cdot 10^{-4}$
60	$9.75 \cdot 10^{-2}$	0.65	206	$3.21 \cdot 10^{-11}$	$9.26 \cdot 10^{-4}$
120	$1.20 \cdot 10^{-1}$	0.40	167	$6.50 \cdot 10^{-10}$	$1.72 \cdot 10^{-3}$

Many Stage Runge-Kutta Methods: Results for nonlinear Systems: 2D Compressible Euler Eq.

- Report error in terms of density L^∞ error

$$e_S^\infty := \|\rho^{(h,S)}(t_f, x) - \rho(t_f, x)\|_\infty \tag{38}$$



Contents

Applied and Computational Mathematics

RWTH AACHEN UNIVERSITY

Summary & Outlook

Summary:

- Different representation of stability polynomials
  - Motivated from theory
  - Enable very efficient optimization
- $p = 2$ Internal-stability optimized RKMs

Outlook/Future Work:

- Optimize $p = 4$ stability polynomials
- Construction of higher order ($p = 3, 4$) RKMs
- Improving internal stability

Thank you for your attention!

Questions?

Applied and Computational Mathematics

RWTH AACHEN UNIVERSITY