Saving computational costs with efficient iterative ADER methods: p-adaptivity, accuracy results and structure preserving limiters

Davide Torlo*, **Lorenzo Micalizzi, Maria Han Veiga, Walter Boscheri**

*MathLab, Mathematics Area, SISSA International School for Advanced Studies, Trieste, Italy
davidetorlo.it

Numhyp23
27th June 2023

Table of contents

Table of contents

Introduction to explicit ADER-DG

History

- Algorithm based on Cauchy-Kovaleskaya theorem (Titarev, Toro 2002)
- High order space-time discretization of the PDE (Dumbser et al. 2008)

- Conservation Laws

$$\partial_t u + \partial_{x_d} F^d(u) = 0$$

Introduction to explicit ADER-DG

History

- Algorithm based on Cauchy-Kovaleskaya theorem (Titarev, Toro 2002)
- High order space-time discretization of the PDE (Dumbser et al. 2008)

- Conservation Laws

$$\partial_t u + \partial_{x_d} F^d(u) = 0$$

- Space–time high order discretization

$$\int_{t^n}^{t^{n+1}} \int_K \theta_i(x,t) \left(\partial_t \theta_j(x,t) q^j + \partial_{x_d} F^d(\theta_j(x,t) q^j) \right) = 0$$

Introduction to explicit ADER-DG

History

- Algorithm based on Cauchy-Kovaleskaya theorem (Titarev, Toro 2002)
- High order space-time discretization of the PDE (Dumbser et al. 2008)

- Conservation Laws

$$\partial_t u + \partial_{x_d} F^d(u) = 0$$

- Space–time high order discretization

$$\int_{t^n}^{t^{n+1}} \int_K \theta_i(x,t) \left(\partial_t \theta_j(x,t) q^j + \partial_{x_d} F^d(\theta_j(x,t) q^j) \right) = 0$$

- **Predictor** (integration by parts in time)

$$\int_K \theta_i(x,t^{n+1}) \cdot \theta_j(x,t^{n+1}) q^j - \int_K \theta_i(x,t^n) \psi_\ell(x) u^{n,\ell} - \int_{t^n}^{t^{n+1}} \int_K \partial_t \theta_i(x,t) \cdot \theta_j(x,t) q^j + \int_{t^n}^{t^{n+1}} \int_K \theta_i(x,t) \partial_{x_d} F^d(\theta_j(x,t) q^j) = 0$$

Introduction to explicit ADER-DG

History

- Algorithm based on Cauchy-Kovaleskaya theorem (Titarev, Toro 2002)
- High order space-time discretization of the PDE (Dumbser et al. 2008)

- Conservation Laws

$$\partial_t u + \partial_{x_d} F^d(u) = 0$$

- Space–time high order discretization

$$\int_{t^n}^{t^{n+1}} \int_K \theta_i(x,t) \left(\partial_t \theta_j(x,t) q^j + \partial_{x_d} F^d(\theta_j(x,t) q^j) \right) = 0$$

- **Predictor** (integration by parts in time)

$$\int_K \theta_i(x,t^{n+1}) \cdot \theta_j(x,t^{n+1}) q^j - \int_K \theta_i(x,t^n) \psi_\ell(x) u^{n,\ell} - \int_{t^n}^{t^{n+1}} \int_K \partial_t \theta_i(x,t) \cdot \theta_j(x,t) q^j + \int_{t^n}^{t^{n+1}} \int_K \theta_i(x,t) \partial_{x_d} F^d(\theta_j(x,t) q^j) = 0$$

- **Corrector** (integration by parts in space and test function constant in time)

$$\int_K \psi_i(x)\psi_\ell(x) u^{n+1,\ell} - \psi_i(x)\psi_\ell(x) u^{n,\ell} + \int_{t^n}^{t^{n+1}} \int_{\partial K} \psi_i(x) \hat{F}^d(\theta_j(x,t) q^j, q_{K^+}(x,t))\, n_d - \int_{t^n}^{t^{n+1}} \int_K \partial_{x_d} \psi_i(x) F^d(\theta_j(x,t) q^j) = 0$$

Corrector

$$\int_K \psi_i(x)\psi_\ell(x)u^{n+1,\ell} - \psi_i(x)\psi_\ell(x)u^{n,\ell} + \int_{t^n}^{t^{n+1}}\int_{\partial K} \psi_i(x)\hat{F}^d(\theta_j(x,t)q^j, q_{K^+}(x,t))\, n_d - \int_{t^n}^{t^{n+1}}\int_K \partial_{x_d}\psi_i(x,t)F^d(\theta_j(x,t)q^j) = 0$$

Properties

- Communication between cells (numerical flux)
- Possibly different degree of basis ψ and θ $(\mathbb{P}^N\mathbb{P}^M)$
- **Explicit** step

Need of extra stabilization?

- Various limiters:
  - WENO, C-WENO
  - A posteriori limiters, e.g. MOOD

Computational cost: scaling factors

- Number of basis functions in space
- Less dependency on time basis functions
- No parallelization
- Needed step to guarantee stability, no room for improvement (?)

Predictor

$$\int_K \theta_i(x,t^{n+1})\cdot\theta_j(x,t^{n+1})q^j - \int_K \theta_i(x,t^n)\psi_\ell(x)u^{n,\ell} - \int_{t^n}^{t^{n+1}}\int_K \partial_t\theta_i(x,t)\cdot\theta_j(x,t)q^j + \int_{t^n}^{t^{n+1}}\int_K \theta_i(x,t)\partial_{x_d}F^d(\theta_j(x,t)q^j) = 0$$

Properties

- **Local** nonlinear system
- Space–time basis/test functions
  - Tensor product space and time basis functions
  - Maximum degree equal to something
  - Lagrange, Taylor
- Possibly many equations in time

How to get the solution

- **Fixed–point** problem and iterations

Questions

- Will fixed–point converge?
- How many **iterations** do we need?
- What is the **order** of accuracy?

Computational cost: scaling factors

- Iterations
- Number of basis functions in time
- Number of basis functions in space
- Space parallelization, domain decomposition

Table of contents

Explicit ADER Predictor for ODE [1]

ODE

$$\partial_t u + F(u) = 0$$

Fixed point iterations to solve

$$\begin{cases} \theta_i(t^{n+1}) \cdot \theta_j(t^{n+1}) q^j - \theta_i(t^n)\psi_\ell(x) u^{n,\ell} - \int_{t^n}^{t^{n+1}} \partial_t \theta_i(t) \cdot \theta_j(t) q^j + \int_{t^n}^{t^{n+1}} \theta_i(t) F(\theta_j(t) q^j) = 0 \\ u^{n+1} = \theta_j(t^{n+1}) q^j \end{cases}$$

Parameters

- Number of fixed–point **iterations**
- Choice of **basis** functions
- Choice of **quadrature** rules
- Where to evaluate F

Runge–Kutta

- Use q^j at each fixed–point iterations as stages
- Stages $=$ iterations $\otimes$ basis functions
- Order of accuracy?
  - Parameters

[1] Han Veiga, Öffner, Torlo. (2021)

ADER-IWF[2]

ADER Integral Weak Form (ADER-IWF)

$$\begin{cases} \theta_i(t^{n+1}) \cdot \theta_j(t^{n+1}) q^j - \theta_i(t^n)\psi_\ell(x)u^{n,\ell} - \int_{t^n}^{t^{n+1}} \partial_t \theta_i(t) \cdot \theta_j(t) q^j + \int_{t^n}^{t^{n+1}} \theta_i(t) F(\theta_j(t) q^j) = 0 \\ u^{n+1} = \theta_j(t^{n+1}) q^j \end{cases}$$

ADER-IWF definition

- Consider **NO iterations**
- $M + 1$ basis functions
- **Implicit** (non linear) method
- Iterations converge to this solution
- Can be written as an **implicit Runge–Kutta**
- Order?

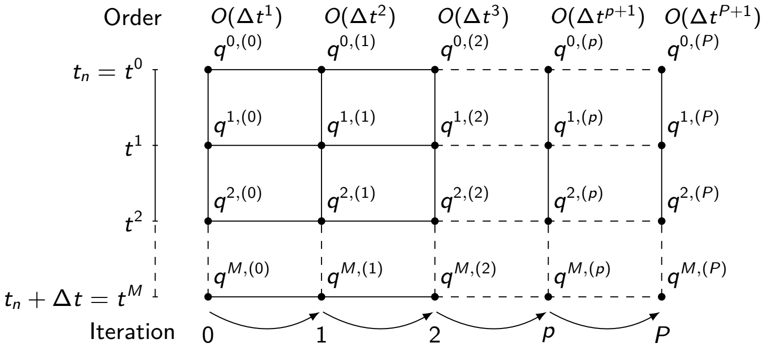
ADER-IWF properties

- Order dictated by quadrature rule and F evaluation
- Equispaced evaluations of $F \Longrightarrow$ Order $M + 1$
- If quadrature and points of evaluation of F coincide, the type of basis functions does not matter
- **Gauss–Lobatto** quad and eval of $F \Longrightarrow$ Order $2M$ (Lobatto IIIC methods)
- **Gauss–Legendre** quad and eval of $F \Longrightarrow$ Order $2M + 1$ (NO Gauss methods)

[2] Han Veiga, Micalizzi, Torlo. (2023)

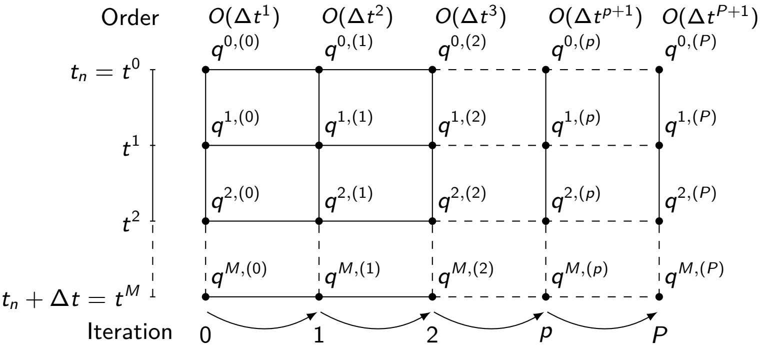
Explicit ADER for ODE: iterations

$$\frac{d}{dt}u(t) = G(t, u(t)), \qquad u^n \approx u(t^n), \qquad q^m : \theta_m(t)q^m \approx u(t)$$



Explicit ADER for ODE: iterations

$$\frac{d}{dt}u(t) = G(t, u(t)), \qquad u^n \approx u(t^n), \qquad q^m : \theta_m(t)q^m \approx u(t)$$



Order is **minimum** between **iterations** P and **order of ADER-IWF**

Costs

Large costs!

Number of Runge–Kutta stages

Equispaced

P	M	ADER
2	1	2
3	2	6
4	3	12
5	4	20
6	5	30
7	6	42
8	7	56
9	8	73
10	9	90

Gauss–Lobatto

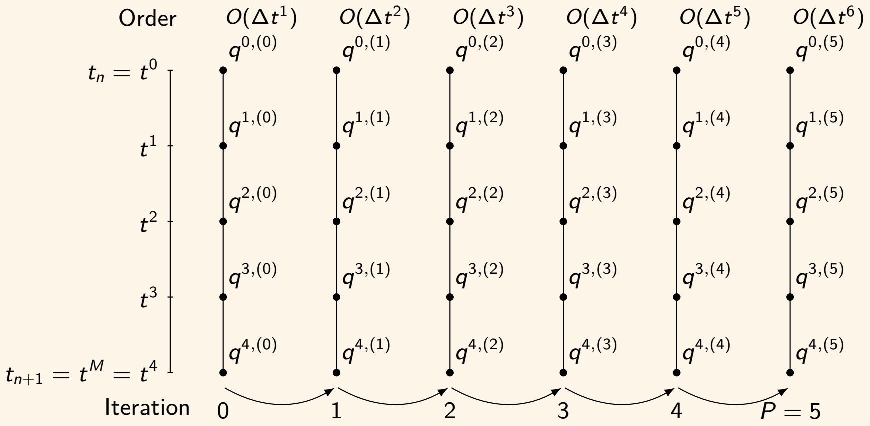
P	M	ADER
2	1	2
3	2	6
4	2	9
5	3	16
6	3	20
7	4	30
8	4	35
9	5	48
10	5	54

Gauss–Legendre

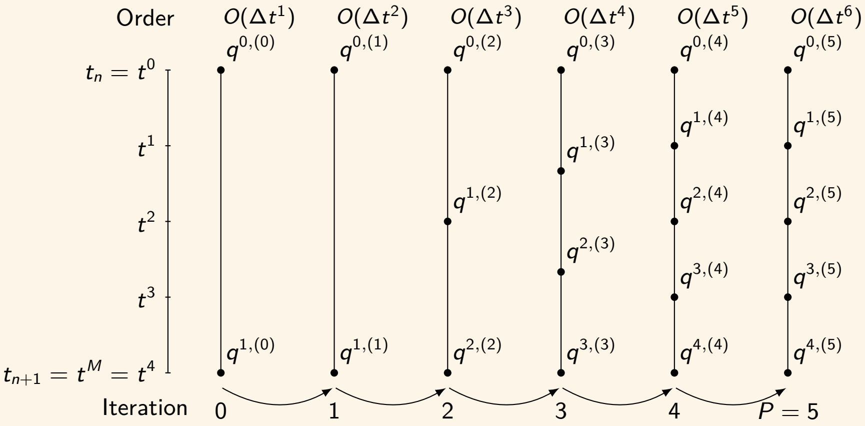
P	M	ADER
2	1	3
3	1	5
4	2	10
5	2	13
6	3	21
7	3	25
8	4	36
9	4	41
10	5	55

How can we save computational time?

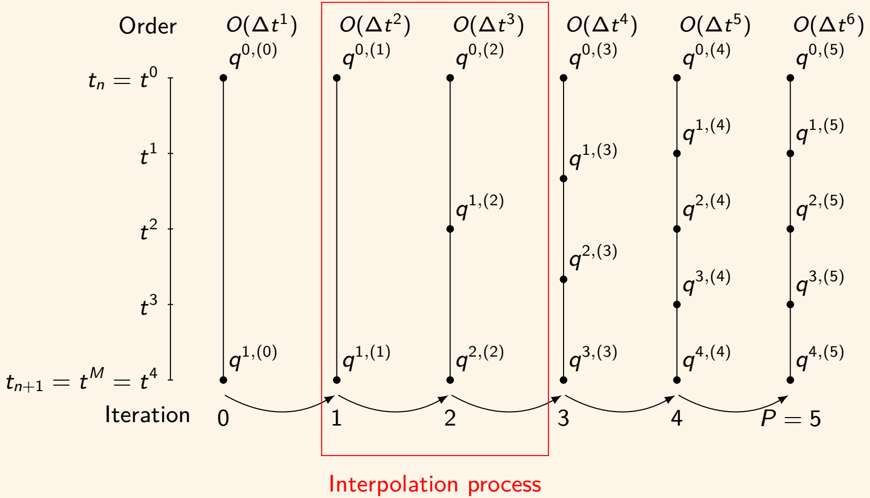
Idea for reduction of stages



Idea for reduction of stages



Idea for reduction of stages



Computational costs reduction: RK stages

Equispaced

Param		RK Stages		
P	M	ADER	ADERu	ADERdu
2	1	2	2	2
3	2	6	6	4
4	3	12	11	7
5	4	20	17	11
6	5	30	24	16
7	6	42	32	22
8	7	56	41	29
9	8	73	51	37
10	9	90	62	46
11	10	111	74	56
12	11	133	87	67
13	12	156	101	79
14	13	183	116	92

Gauss-Lobatto

Param		RK Stages		
P	M	ADER	ADERu	ADERdu
2	1	2	2	2
3	2	6	6	4
4	2	9	9	7
5	3	16	15	11
6	3	20	19	15
7	4	30	27	21
8	4	35	32	26
9	5	48	42	34
10	5	54	48	40
11	6	71	60	50
12	6	78	67	57
13	7	96	81	69
14	7	104	89	77

Gauss-Legendre

Param		RK Stages		
P	M	ADER	ADERu	ADERdu
2	1	3	3	3
3	1	5	5	5
4	2	10	10	9
5	2	13	13	12
6	3	21	20	18
7	3	25	24	22
8	4	36	33	30
9	4	41	38	35
10	5	55	49	45
11	5	61	55	51
12	6	78	68	63
13	6	85	75	70
14	7	105	90	84

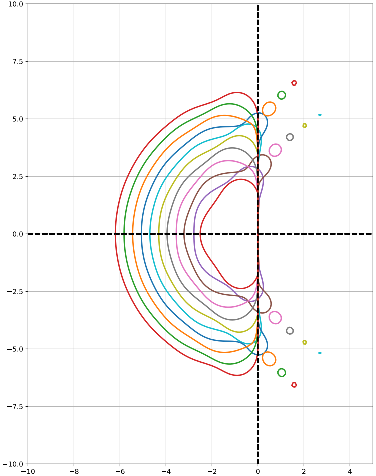
Stability Properties

Stability of ADER-ADERu-ADERdu

The **stability function** of ADER, ADERu, ADERdu of order P for any basis function and quadrature distribution is

$$R(z) = 1 + z + \frac{z^2}{2!} + \cdots + \frac{z^P}{P!}.$$

ADER, ADERu, ADERdu



Adaptive DeC

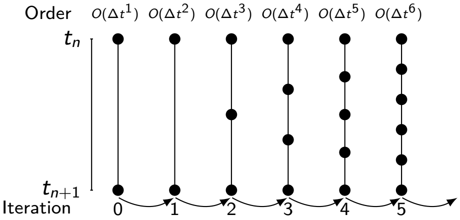
How can we exploit the increasing order of accuracy?

Adaptive DeC

How can we exploit the increasing order of accuracy?

Adaptive order DeC

- Set tolerance ε
- Check at each iteration if $\left\|\underline{\boldsymbol{u}}^{(p)} - \underline{\boldsymbol{u}}^{(p-1)}\right\| < \varepsilon$
- Stop at a certain order when tolerance is reached



Adaptive DeC

How can we exploit the increasing order of accuracy?

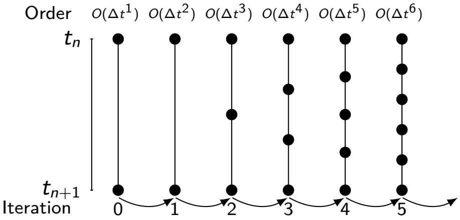
Adaptive order DeC

- Set tolerance ε
- Check at each iteration if $\left\| \underline{\boldsymbol{u}}^{(p)} - \underline{\boldsymbol{u}}^{(p-1)} \right\| < \varepsilon$
- Stop at a certain order when tolerance is reached

☺ Saving on useless iterations

☺ Reach the needed order for tolerance

☹ Sub-optimal (waste of few stages)



ODE test: C5 five body problem in 3D

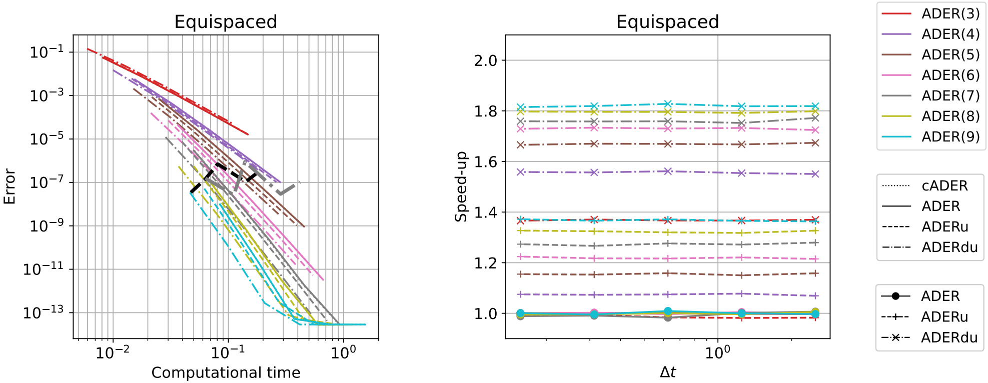


Figure: C5: Error with respect to computational time (left) and speed-up with respect to classical ADER (with $M+1=P$) (right).

ODE test: C5 five body problem in 3D

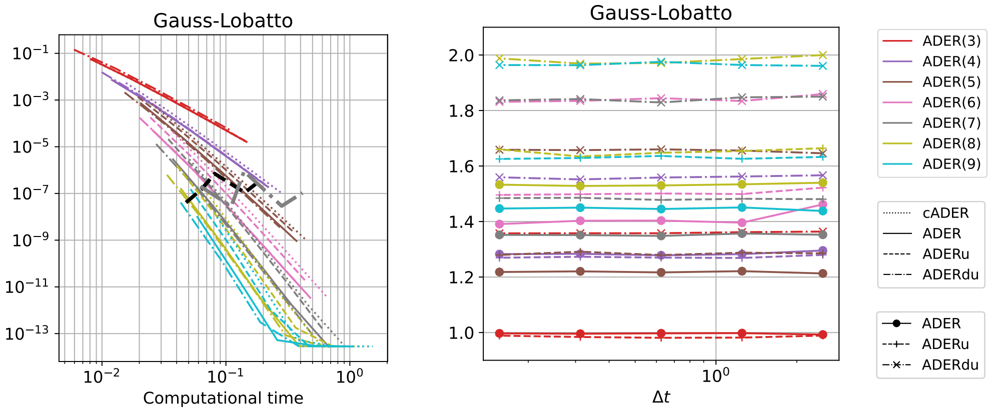


Figure: C5: Error with respect to computational time (left) and speed-up with respect to classical ADER (with $M+1=P$) (right).

ODE test: C5 five body problem in 3D

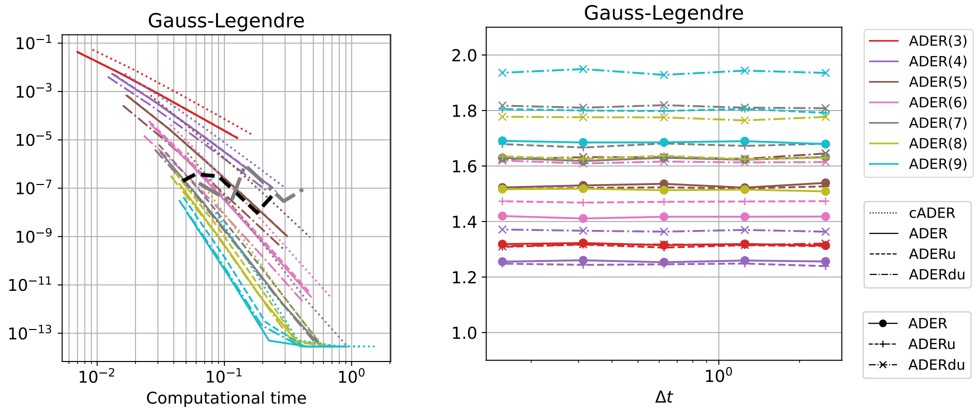


Figure: C5: Error with respect to computational time (left) and speed-up with respect to classical ADER (with $M + 1 = P$) (right).

Table of contents

Efficient ADERdu for PDEs

- Predictor (integration by parts in time)

$$\int_K \theta_i(x,t^{n+1})q^{(p)}(x,t) - \int_K \theta_i(x,t^n)u^n(x) - \int_{T^n\times K} \partial_t\theta_i(x,t)q^{(p)}(x,t) + \int_{T^n\times K} \theta_i(x,t)\partial_{x_d}F^d(q^{(p-1)}(x,t)) = 0$$

- Corrector (integration by parts in space and test function constant in time)

$$\int_K \psi_i(x)u^{n+1}(x) - \psi_i(x)u^n(x) + \int_{t^n}^{t^{n+1}}\int_{\partial K} \psi_i(x)\hat{F}^d(q_K^{(P)}(x,t), q_{K^+}^{(P)}(x,t))\, n_d - \int_{t^n}^{t^{n+1}}\int_K \partial_{x_d}\theta_i(x,t)F^d(q^{(P)}(x,t)) = 0$$

Predictor setting

- Polytopal meshes (hexagonal)
- θ_i Taylor basis in space–time $\mathbb{P}^N(T^n \times K)$
- $q|_K \in \mathbb{P}^N(T^n \times K)$

Corrector setting

- Polytopal meshes (hexagonal)
- ψ_i Taylor basis in space $\mathbb{P}^M(K)$ with $M \leq N$
  - $M = N$ DG, limiter for non smooth
  - $M = 0$ DG-FV, with C-WENO reconstruction

Efficient ADERdu for PDEs

- Predictor (integration by parts in time)

$$\int_K \theta_i^{(p)}(x,t^{n+1}) q^{(p)}(x,t) - \int_K \theta_i^{(p)}(x,t^n) u^n(x) - \int_{T^n\times K} \partial_t \theta_i^{(p)}(x,t) q^{(p)}(x,t) + \int_{T^n\times K} \theta_i^{(p)}(x,t) \partial_{x_d} F^d(q^{(p-1)}(x,t)) = 0$$

- Corrector (integration by parts in space and test function constant in time)

$$\int_K \psi_i(x) u^{n+1}(x) - \psi_i(x) u^n(x) + \int_{t^n}^{t^{n+1}} \int_{\partial K} \psi_i(x) \hat{F}^d(q_K^{(P)}(x,t), q_{K^+}^{(P)}(x,t))\, n_d - \int_{t^n}^{t^{n+1}} \int_K \partial_{x_d} \theta_i(x,t) F^d(q^{(P)}(x,t)) = 0$$

Predictor setting

- Polytopal meshes (hexagonal)
- $\theta_i^{(p)}$ Taylor basis in space–time $\mathbb{P}^{\min(p,N)}(T^n \times K)$
- $q^{(p)}|_K \in \mathbb{P}^{\min(p,N)}(T^n \times K)$
- $q^{(p-1)}|_K \in \mathbb{P}^{\min(p-1,N)}(T^n \times K)$
- Hierarchical matrices

Corrector setting

- Polytopal meshes (hexagonal)
- ψ_i Taylor basis in space $\mathbb{P}^M(K)$ with $M \leq N$
  - $M = N$ DG, limiter for non smooth
  - $M = 0$ DG-FV, with C-WENO reconstruction

DOOM: a posteriori limiter in the predictor

Goal

Detect “bad” behaviors and avoid them using low order reconstructions

- NaNs
- Negative density/pressure
- Discrete maximum principle violations

DOOM: a posteriori limiter in the predictor

Goal

Detect "bad" behaviors and avoid them using low order reconstructions

- NaNs
- Negative density/pressure
- Discrete maximum principle violations

MOOD (Clain et al. (2011))

- Run the whole time step with high order method
- Detect **troubled** cells
- **Run lower order** scheme
- Until criteria are met
- Lowest order scheme is parachute, should always work

DOOM: a posteriori limiter in the predictor

Goal

Detect “bad” behaviors and avoid them using low order reconstructions

- NaNs
- Negative density/pressure
- Discrete maximum principle violations

MOOD (Clain et al. (2011))

- Run the whole time step with high order method
- Detect **troubled** cells
- **Run lower order** scheme
- Until criteria are met
- Lowest order scheme is parachute, should always work

DOOM (Micalizzi, Torlo, Boscheri. (2023))

- Discrete Optimally increasing Order Method
- At each iteration checks criteria
- As soon as they are not met, we fall to the previous order
- **No need to recompute** anything
- **Guarantees** that the previous order met the criteria

DOOM: a posteriori limiter in the predictor

Goal

Detect "bad" behaviors and avoid them using low order reconstructions

- NaNs
- Negative density/pressure
- Discrete maximum principle violations

Test Setting

- $\mathbb{P}^N\mathbb{P}^0$
- Parachute is FV
- Checks on NaNs and positivity of ρ and p
- C-WENO for reconstructions and oscillations

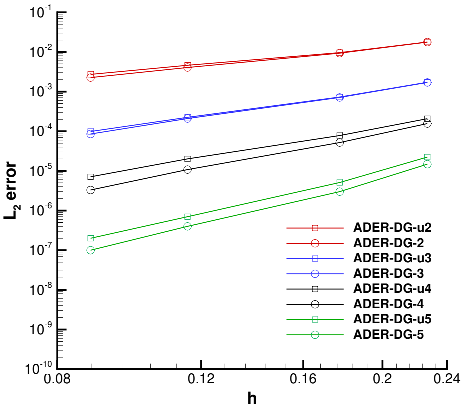
MOOD (Clain et al. (2011))

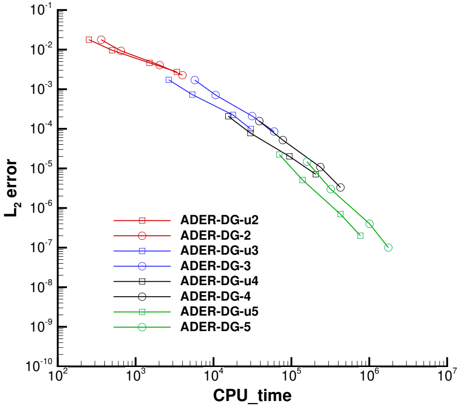
- Run the whole time step with high order method
- Detect **troubled** cells
- **Run lower order** scheme
- Until criteria are met
- Lowest order scheme is parachute, should always work

DOOM (Micalizzi, Torlo, Boscheri. (2023))

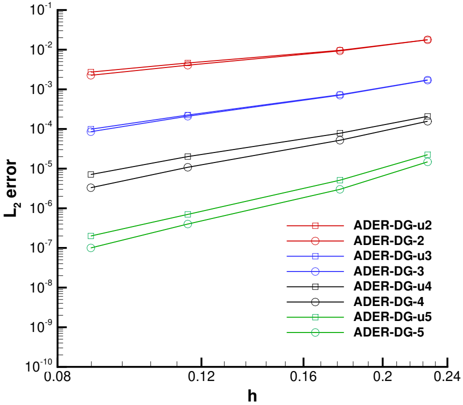
- Discrete Optimally increasing Order Method
- At each iteration checks criteria
- As soon as they are not met, we fall to the previous order
- **No need to recompute** anything
- **Guarantees** that the previous order met the criteria

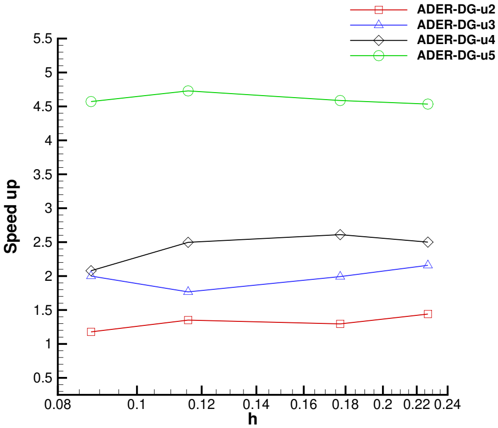
Test: isotropic vortex for compressible Euler equations





Test: isotropic vortex for compressible Euler equations





Test: Riemann Problems – High Jump in Pressure

$$\rho_L = \rho_R = 1, \ u_L = u_R = 0, \ p_L = 1000, \ p_R = 1$$

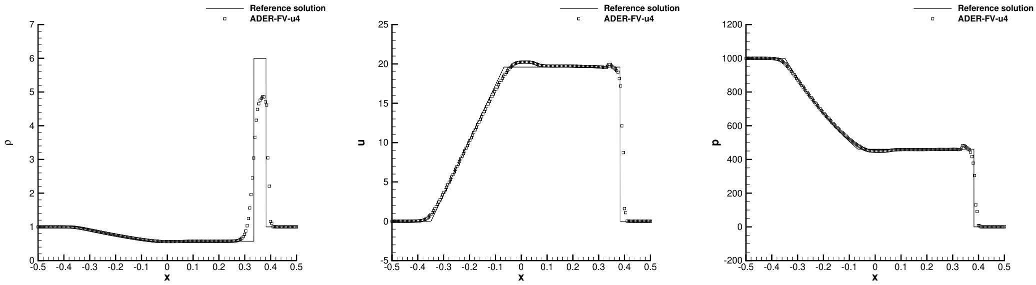


Figure: Double Rarefaction: Density (left), pressure (center) and internal energy (right)

Test: Riemann Problems – Double rarefaction

$$\rho_L = \rho_R = 1, \; u_L = -2, \; u_R = 2, \; p_L = p_R = 0.4$$

Very difficult to keep positivity of density and pressure!

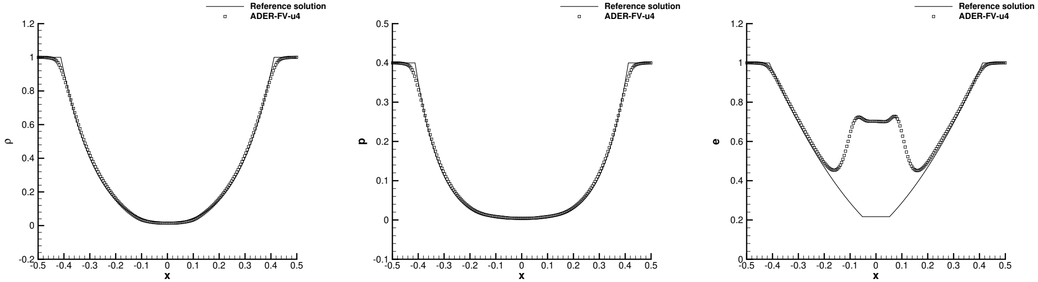


Figure: Double Rarefaction: Density (left), pressure (center) and internal energy (right)

Table of contents

Summary and perspectives

Summary

- ADER
- **Efficient** ADER
- Careful choice of iterations, basis functions
- **Increasing order** of reconstruction with iterations
- Easy to change in an ADER code
- Adaptivity
- **DOOM** a posteriori limiter
- Speed up (up to factor 4 in an ADER parallel code)

Summary

- ADER
- **Efficient** ADER
- Careful choice of iterations, basis functions
- **Increasing order** of reconstruction with iterations
- Easy to change in an ADER code
- Adaptivity
- **DOOM** a posteriori limiter
- Speed up (up to factor 4 in an ADER parallel code)

Perspectives

- Other criteria in DOOM procedure
- DOOM for plateaux $\Longrightarrow$ low order
- DOOM for shocks $\Longrightarrow$ low order

Summary and perspectives

Summary

- ADER
- **Efficient** ADER
- Careful choice of iterations, basis functions
- **Increasing order** of reconstruction with iterations
- Easy to change in an ADER code
- Adaptivity
- **DOOM** a posteriori limiter
- Speed up (up to factor 4 in an ADER parallel code)

Perspectives

- Other criteria in DOOM procedure
- DOOM for plateaux $\Longrightarrow$ low order
- DOOM for shocks $\Longrightarrow$ low order

THANK YOU!

Preprints

- M. Han Veiga, L. Micalizzi and D. Torlo. "On improving the efficiency of ADER methods." (2023) arXiv:2305.13065
- L. Micalizzi, D. Torlo and W. Boscheri. "Efficient iterative arbitrary high order methods: an adaptive bridge between low and high order." (2022) arXiv:2212.07783.