Efficiency of the high-order Spectral Difference method in combustion

T. Marchal[a,b,c], A. De Brauer[b], H. Deniau[c], J-F Boussuge[b], B. Cuenot[b], R. Mercier[a]

[a] **Safran Tech**, Digital Sciences and Technologies Department, Rue des Jeunes Bois, Châteaufort, Magny-Les-Hameaux 78114, France

[b] **CERFACS**, 42 avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France

[c] **ONERA/DMPE**, Université de Toulouse, F-31055 Toulouse, France

39th INTERNATIONAL SYMPOSIUM ON COMBUSTION

24-29 JULY 2022
VANCOUVER, CANADA

Outline

1. Some context about high-order methods and the Spectral Difference (SD) method in combustion.
2. Influence of polynomial degree on accuracy of the results using the SD method.
3. Influence of polynomial degree on computational cost using the SD method.
4. Use of local polynomial adaptation to reduce computational cost with the SD method.
5. General conclusion.

A little bit of context

Context

- Large Eddy Simulations (LES) become essential to simulate correctly flame/turbulence interactions.
- LES need an accurate spatial discretization with low dissipation and dispersion errors.
- High-order methods (HOM) such as Discontinuous Galerkin[1] (DG), Flux Reconstruction[2] (FR) or Spectral Differences[3] (SD) methods have these properties.
- HOM can manage both mesh refinement and the order of the local representation of the solution.
- However, very few combustion applications were considered with HOM for now.
- Most of them were done using the DG method [4,5,6,7] and not using FR or SD...

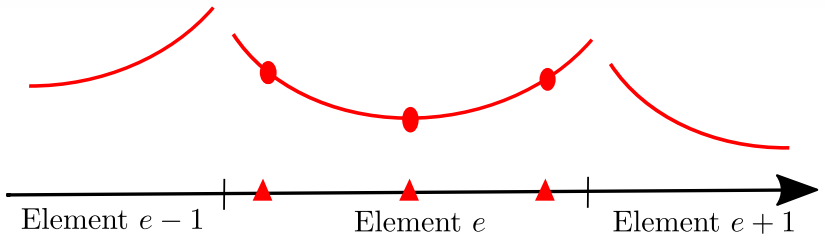
Objectives

★ Evaluate the SD capability in terms of accuracy and performance to simulate combustion.

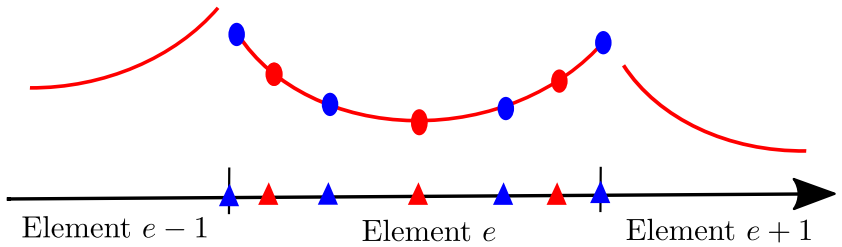
★ Use local p-adaptation on a combustion case with the SD method.

1. Reed et *al*, Triangular mesh methods for the neutron equation, Tech. rep., Los Alamos Scientific Lab., N. Mex.(USA) (1973).
2. HUYNH, Hung T. A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods. In : 18th AIAA computational fluid dynamics conference. 2007. p. 4079.
3. D. A. Kopriva, A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method, J. Comput. Phys. 128 (2) (1996) 475-488.
4. Lv et *al*, Discontinuous Galerkin method for multicomponent chemically reacting flows and combustion, J.Comput. Phys. 270 (2014) 105-137.
5. Lv et *al*, High-order discontinuous Galerkin method for applications to multicomponent and chemically reacting flows, Acta Mech. Sin. 33 (3) (2017) 486-499.
6. Billet et *al*, A Runge Kutta discontinuous Galerkin approach to solve reactive flows on conforming hybrid grids: the parabolic and source operators, Comput. Fluids 95 (2014) 98-115.
7. Johnson et *al*, A conservative discontinuous Galerkin discretization for the chemically reacting Navier-Stokes equations, J. Comput. Phys. 423 (2020) 109826.

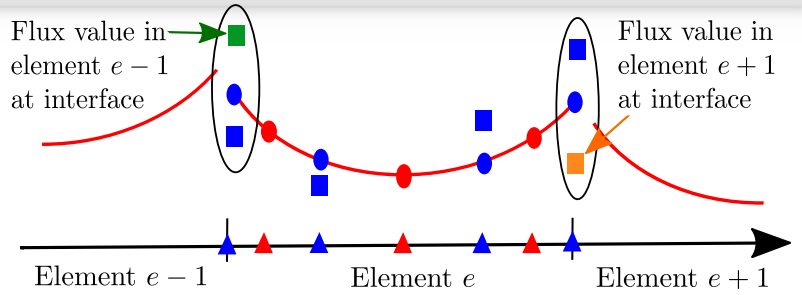
The SD method on a 1D example with p=2 (order 3)



Step 1: Build a continuous polynomial of degree $p = 2$ for $\mathbf{U}$ in element e based on the 3 values of $\mathbf{U}$ at solution points (SP).



Step 2: Evaluation of $\mathbf{U}$ at the 4 flux points (FP) in element e.



Step 3: Evaluation of $\mathbf{E}$ at the 4 flux points (FP) in element e. There are 2 flux values at interfaces between elements...

SP FP

U_{SP} U_{FP}

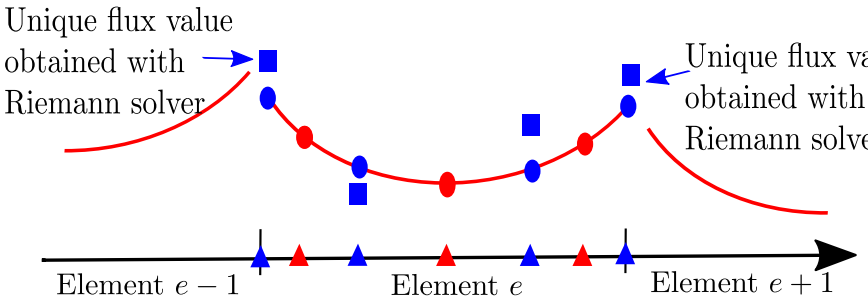
$\left(\frac{\partial \mathbf{E}}{\partial x}\right)_{SP}$ E_{FP}

Solve for $p = 2$:

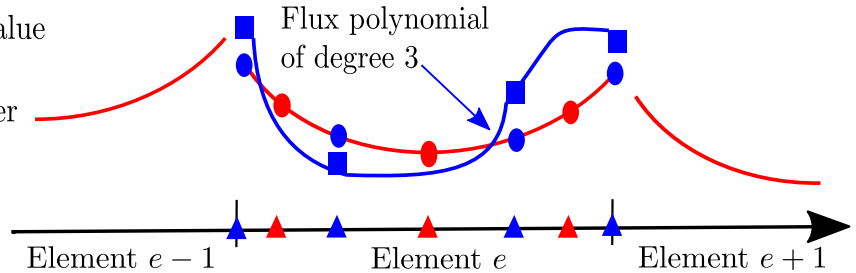
$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} = \mathbf{0}$$

$$\mathbf{E} = \mathbf{E}(\mathbf{U})$$

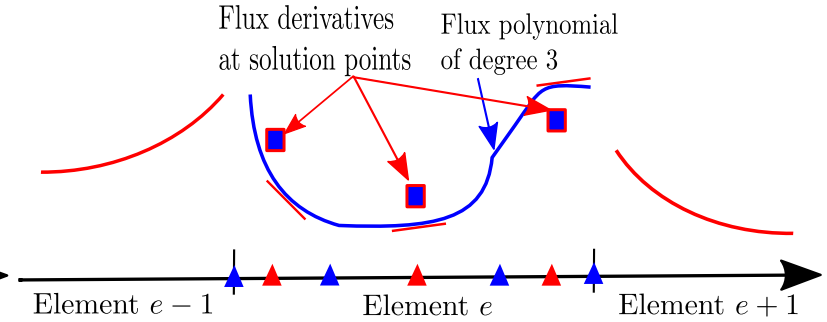
After step 6, $\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} = \mathbf{0}$ can be marched in time using any temporal scheme



Step 4: Use a Riemann solver to have an unique flux value at element interfaces.



Step 5: Build a flux polynomial of degree $p+1 = 3$ based on values of $\mathbf{E}$ at the 4 flux points (FP).



Step 6: Derivation of flux polynomial at the 3 solution points (SP) to get $\left(\frac{\partial \mathbf{E}}{\partial x}\right)_{SP}$.

SD method in combustion

- Very few combustion applications were done using the SD method:
  1. Gupta et *al*[8] and Tofaili et *al*[9] show 1D detonations.
  2. Marchal et *al*[10] succeeded in simulating 1D and 2D laminar flames.
- **Question:** Interest of using such method in combustion in terms of accuracy and computational cost?
- **To answer**

★ Compare JAGUAR, SD code validated in ref [10] in combustion, and AVBP[11] in terms of accuracy and computational cost.

JAGUAR

- **Species transport:** Hirschfelder and Curtiss with a constant Schmidt number per species.
- Species **source terms** computed with **Arrhenius's law**.
- **JANAF** enthalpy tables used for **thermodynamic**.
- Spatial schemes are of order $p+1$, p = polynomial degree

AVBP

- **Species transport:** Hirschfelder and Curtiss with a constant Schmidt number per species.
- Species **source terms** computed with **Arrhenius's law**.
- **JANAF** enthalpy tables used for **thermodynamic**.
- Spatial schemes: Lax-Wendroff (2nd order in space) (LW) and Two-step Taylor Galerkin (3rd order in space) (TTGC).

8. Gupta et *al*, Numerical investigation of sustained planar detonation waves in a periodic domain, in: 2018 Fluid Dynamics Conference, 2018, p. 3240.
9. Tofaili et *al*, One-dimensional dynamics of gaseous detonations revisited, Combustion and Flame 232 (2021) 111535.
10. Marchal et *al*, Extension of the Spectral Difference method to combustion, Submitted to Journal of Computational Physics. https://arxiv.org/pdf/2112.09636.pdf
11. Schonfeld et *al*, Steady and unsteady flow simulations using the hybrid flow solver AVBP, AIAA journal 37 (11) (1999) 1378–1385.

Concept of degrees of freedom and case presentation

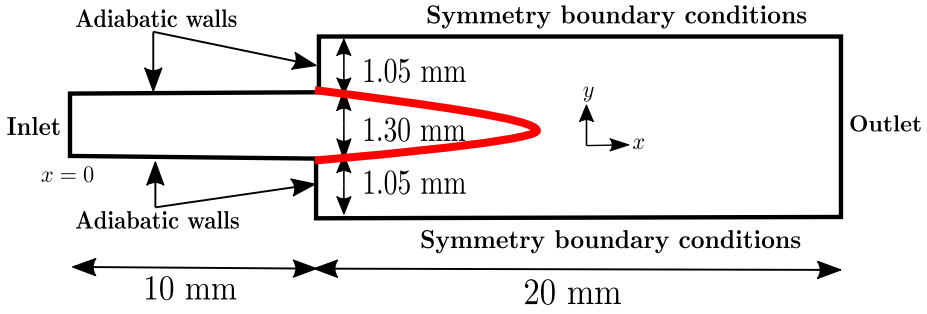
- Degrees of freedom (DOF) = points where the numerical solution is advanced in time.

JAGUAR	**AVBP**
DOF = **Solution points** in the whole domain = $N_e\,(p+1)^2$	DOF = **cell nodes** in the whole domain
N_e = number of quadrilateral elements	
$(p+1)^2$ = Solution points in a quadrilateral element	

- **Fair comparison** done at **same DOF value**. 2 choices available to keep DOF $\approx$ cst in SD:

★ **Choice 1** = large N_e with small degree p OR **Choice 2** = small N_e with large degree p.

Question: Which choice is better in terms of accuracy and computational time?



p	2	3	4	5	6
N_e	2230	1256	804	558	410

p	2	3	4	5	6
N_e	3437	1930	1236	860	632

- 2D burner case run from $p = 2$ to $p = 6$.
- For each p value, N_e was adjusted to keep respectively 20100 and 30900 DOF.
- These DOF values correspond respectively to 6 and 8 points in the flame front.

Influence of polynomial order on the solution accuracy

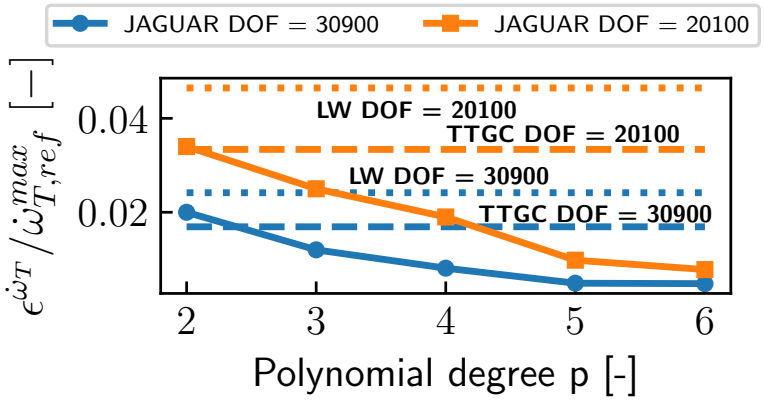


Figure: Evolution of $\epsilon^{\dot{\omega}_T}/\dot{\omega}^{max}_{T,ref}$ with respect to p for two numbers of DOF. AVBP schemes (LW and TTGC) at these DOF values are also shown with horizontal lines.

L2-error on the 2D heat release rate field:

$$\epsilon^{\dot{\omega}_T} = \left[\sum_{i=1}^{\text{DOF}} \left(\dot{\omega}_{T,i} - \dot{\omega}^{interp}_{T,i,ref}\right)^2 / \text{DOF}\right]^{1/2}$$

Heat release at point i in the considered solution

Heat release at point i of the reference solution

- Reference solution is at $p = 6$ with 23 points in the flame front.

Conclusion on accuracy results

- Error ↘ when DOF ↗ for both JAGUAR and AVBP simulations.
- At a given DOF, error ↘ when scheme order ↗ also for both codes.
- In JAGUAR case, error at high DOF with low p is recovered at low DOF with high p.
- **Large elements with high p values is better for accuracy. What about computational cost?**

Influence of polynomial order on computational cost

2 quantities are of interest:

- **Iteration cost per DOF** $\kappa\,[\mu s/\text{ite/DOF}]$ AND **real time taken to simulate 1 ms of physical time** τ[s].

Results for both AVBP and JAGUAR on 36 Intel cores

JAGUAR					
p	2	3	4	5	6
N_e	3437	1930	1236	860	632
κ	8.5	7.7	7.4	7.3	7.1
τ	361	371	407	471	558

AVBP				
	κ_{LW}	2.8	κ_{TTGC}	6.3
	τ_{LW}	106	τ_{TTGC}	239

Values of κ and τ in the **30900 DOF case.**

JAGUAR					
p	2	3	4	5	6
N_e	2230	1256	804	558	410
κ	8.2	7.5	7.4	7.2	7.2
τ	182	190	216	247	292

AVBP				
	κ_{LW}	2.8	κ_{TTGC}	6.4
	τ_{LW}	57	τ_{TTGC}	128

Values of κ and τ in the **20100 DOF case.**

JAGUAR

- κ almost constant with p and close to κ_{TTGC}.
- τ is higher and increases with p due to CFL restrictions.

$$\text{CFL}_{lim} \sim (p+1)^{-1}$$

AVBP

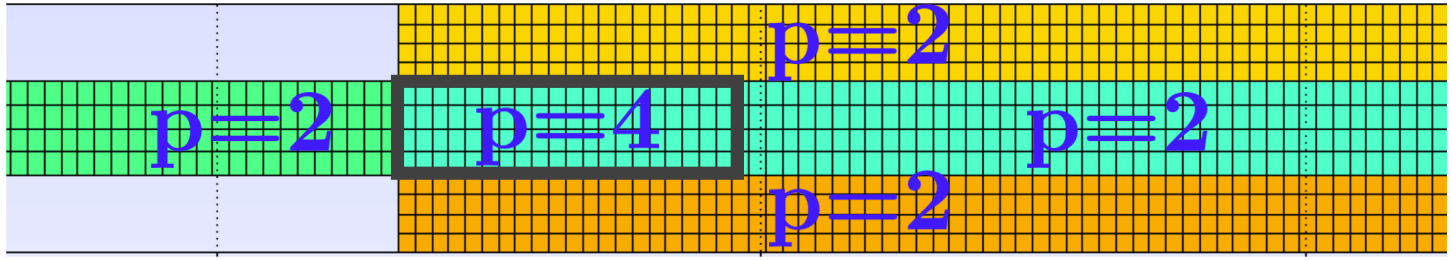
- κ increases with scheme order: $\kappa_{\text{TTGC}} > \kappa_{\text{LW}}$.
- τ increases mostly because of κ but same CFL value (≈ 0.7) for LW and TTGC.

The use of p-adaptation to reduce computational cost

- **Local *p*-adaptation** puts high values of p only in regions of interest.
- Thus, less DOF but accuracy is conserved $\Rightarrow$ **gain in computational time**.
- However, it does not solve time steps issues.

Illustration of local *p*-adaptation on the 2D burner case

Mesh firstly designed for 30900 DOF case with $p = 4$ in all elements:



★ In near flame elements: $p = p_{max} = 4$ ★ $p = p_{min} = 2$ elsewhere

- Computation done in **3 steps:**
  1. Compute **sensor values** $\widetilde{\theta}_e \in [0, 1]$ for each element with a **first simulation** at $p = 2$.
  2. Set **Polynomial order per element**: $p_e = p_{min} + \text{INT}\left[\tanh\left(\alpha\widetilde{\theta}_e\right)(p_{max} - p_{min})\right]$, $\alpha = 100$ here.
  3. A **new computation** with the p_e distribution is runned.
- **Sensor** is based on the **norm of density gradient** here.

The use of p-adaptation to reduce computational cost

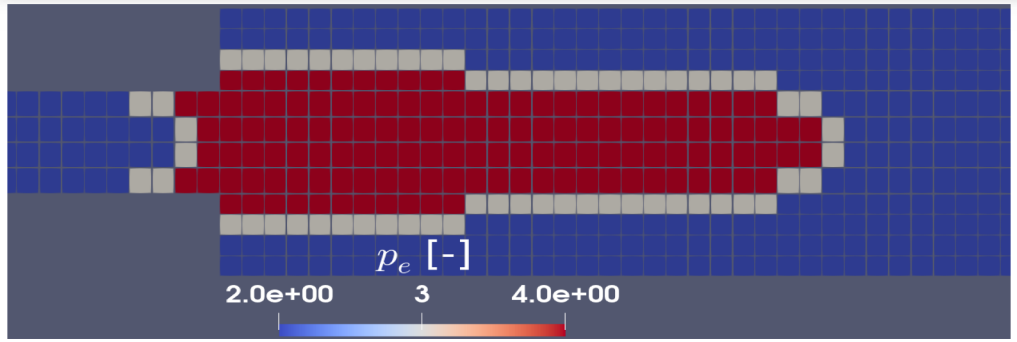


Figure: Polynomial repartition per element when $p_{min} = 2$ and $p_{max} = 4$ on the same mesh.

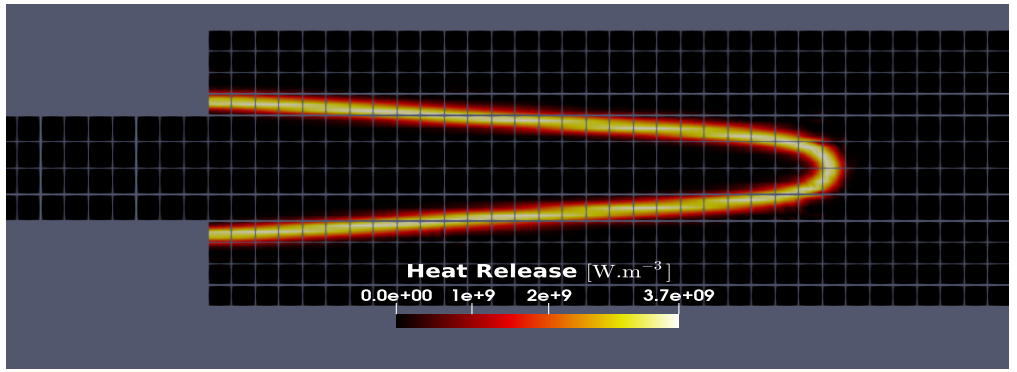


Figure: Steady 2D heat release rate field when $2 \leq p \leq 4$ on the same mesh.

	$p = 2$	$2 \leq p \leq 4$	$p = 4$
DOF	10944	13444	30900
$\epsilon^{\dot{\omega}_T}/\dot{\omega}^{max}_{T,ref}$	4.41×10^{-2}	8.7×10^{-3}	8.1×10^{-3}
κ	8.3	7.9	7.4
τ	96	199	407

Table: Values of DOF, $\epsilon^{\dot{\omega}_T}/\dot{\omega}^{max}_{T,ref}$, $\kappa[\mu s/\text{ite/DOF}]$ and $\tau[s]$ between a computation at $p = 2$, $2 \leq p \leq 4$ and $p = 4$ on the same mesh.

Compared to the full $p = 4$ case

- With $2 \leq p \leq 4$: **56% less DOF.**
- **Without** sensor evaluation:
  $\tau\,(2 \leq p \leq 4) = 295 < \tau\,(p = 4) = 407$: **51% gain in τ.**
- **With** sensor evaluation:
  $\tau\,(p = 2) + \tau\,(2 \leq p \leq 4) = 295 < \tau\,(p = 4) = 407$: **28% gain in τ.**

- **Load balancing** employed to put more processors on zones where p is higher.

General conclusion

The SD method:

- Performs better with **large elements** coupled with **high p** values.
- Has iteration cost **almost constant with p**.
- Has a **computational time increasing with p** because of CFL restrictions.
- Can use **local p-adaptation to keep accuracy and gain in computational time** without re-meshing.

Next step

- Compare Adaptative Mesh Refinement[12] coupled with finite volume method and local p-adaptation in combustion.

12. J.Bell, AMR for low mach number reacting flow, in: Adaptive Mesh Refinement-Theory and Applications, Springer, 2005, pp. 203–221.

Thank you for your attention! Questions?

Influence of polynomial order on computational cost

Comments about κ almost constant with p

- Competition between 2 processes: extrapolation process at FP and interface treatments at FP.
- At constant DOF: **extrapolation process** $\sim \boldsymbol{O(p)}$ and **interface treatment** $\sim \boldsymbol{O(1/p)}$.
- **Low p values:** interface treatment dominates (more interface FP and also more MPI communications).
- **High p values:** interface treatment is less costly and compensates the $O(p)$ increases as long as $p < 8$.
- With the DG method, an increase of p results in an increase of κ even if $p < 8$ because of quadrature rules[13].

Comments about τ well higher for the SD method

- τ is linked to the timestep/CFL used and in SD the CFL limit scales as $(p+1)^{-1}$.
- In this work, for $p = \{2, 3, 4, 5, 6\}$, CFL values were set to 0.36, 0.32, 0.28, 0.24 and 0.20 with a RK3-TVD scheme[14].
- On the other hand, AVBP is usually at CFL = 0.7 for both LW and TTGC schemes.
- However, this CFL difference is even higher for DG schemes where the CFL limit scales as[15] $(2p+1)^{-1}$.

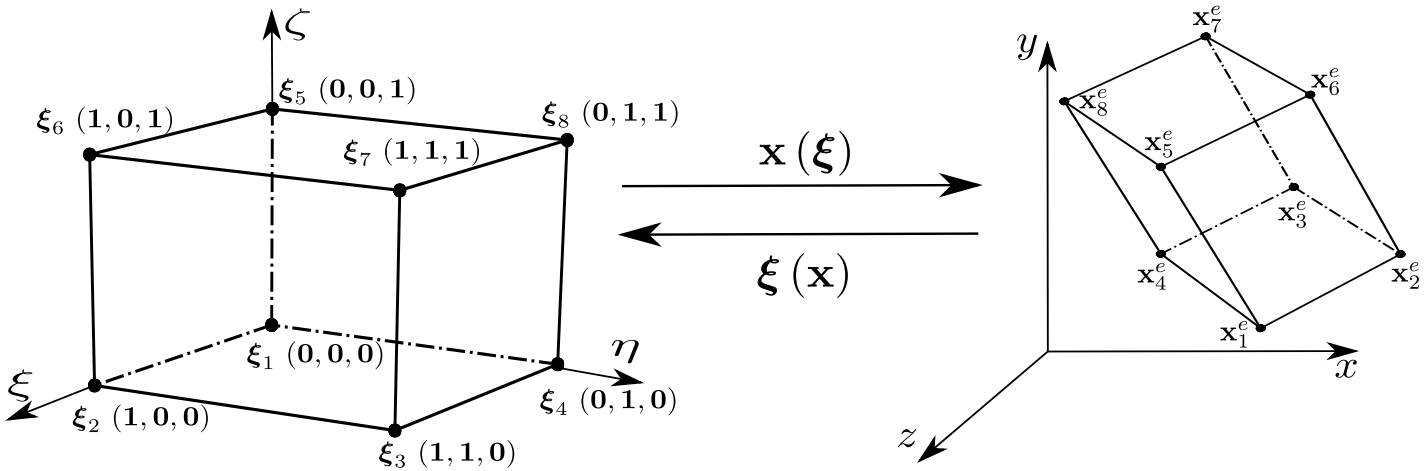
13. Wu et *al*, On the accuracy and efficiency of discontinuous Galerkin, Spectral Difference and Correction Procedure via Reconstruction methods, J. Comput. Phys. 259 (2014) 70–95.
14. Gottlieb et *al*, Total Variation diminishing Runge-Kutta schemes, Math. Comput. 67 (221) (1998) 73–85.
15. Cockburn et *al*, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, SIAM J. Sci. Comput. 16 (3) (2001) 173–261.

Isoparametric transformation for hexahedral elements

- Any hexahedral mesh element $[a,b] \times [c,d] \times [e,f]$ is transformed into a cube $[0,1]^3$:

$$\underset{x}{[a,b]} \times \underset{y}{[c,d]} \times \underset{z}{[e,f]} \xrightarrow[|J|]{} \underset{\xi}{[0,1]} \times \underset{\eta}{[0,1]} \times \underset{\zeta}{[0,1]}$$

with $|J|$ the determinant of the Jacobian of the transformation.



- Cube $[0,1]^3$ is called the **isoparametric domain**.
- Equations are also transformed and solved in the isoparametric domain.
- Such transformation is called **isoparametric**.

Position of SP and FP in 1D

- SP and FP positions are not arbitrary.
- Actually, in 1D-SD any segment $[a, b]$ is transformed into segment $[0, 1]$ using:

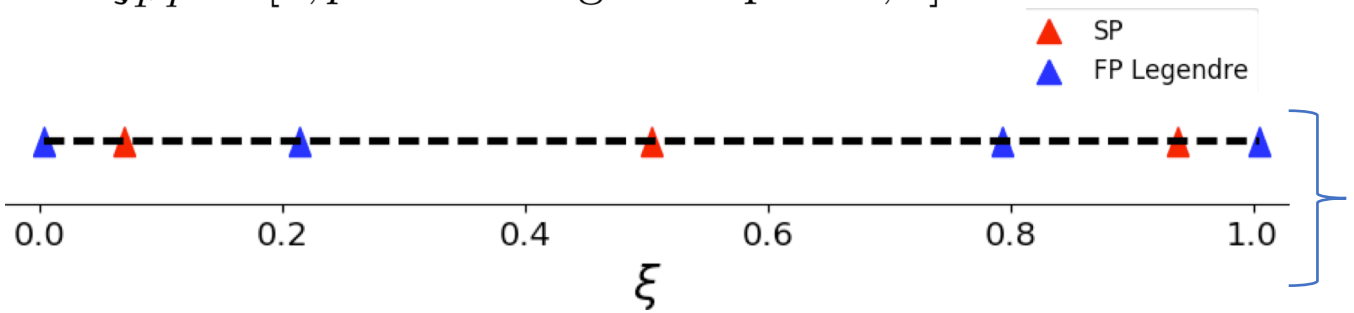
$$\underset{x}{[a,b]} \xrightarrow[|J|]{} \underset{\xi}{[0,1]} \qquad \textbf{isoparametric transformation in 1D}$$

with $|J|$ the determinant of the Jacobian of the transformation.

- Then, each element is treated in the same way using SP and FP located at:

$$\boldsymbol{\xi}_{SP} = \left[\frac{1}{2}\left(1 - \cos\left(\frac{2i-1}{2\,(p+1)}\pi\right)\right)\right]_{1\leq i\leq p+1} \quad \text{Gauss-Chebyshev points} \qquad (1)$$

$$\boldsymbol{\xi}_{FP} = [0, p \text{ Gauss-Legendre points}, 1] \qquad (2)$$



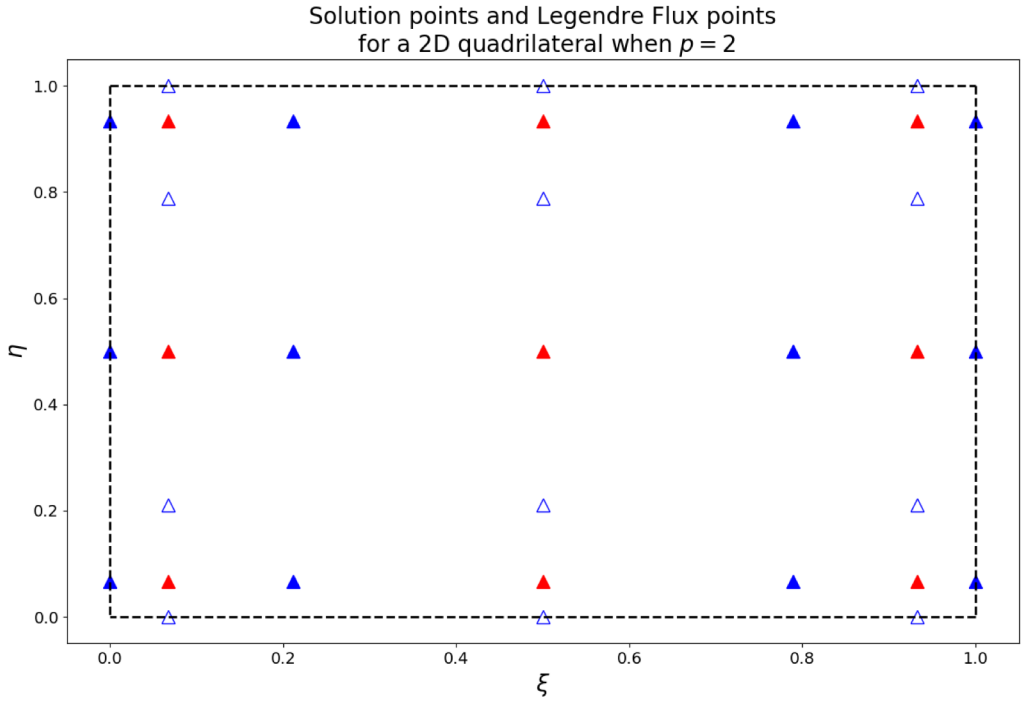
SP and FP for $p = 2$ in 1D in the isoparametric domain

Position of SP and FP: extension in 2D and 3D

- In 2D and 3D, SP and FP are set direction by direction as in 1D.
- For instance in 2D, quadrilateral $[a,b] \times [c,d]$ is transformed into $[0,1] \times [0,1]$:

$$\underset{x}{[a,b]} \times \underset{y}{[c,d]} \xrightarrow[|J|]{} \underset{\xi}{[0,1]} \times \underset{\eta}{[0,1]} \qquad \textbf{isoparametric transformation in 2D}$$

- **In dimension *d*:** $N_{SP} = (p+1)^d$, $N_{FP} = d \times (p+2)\,(p+1)^{d-1}$.



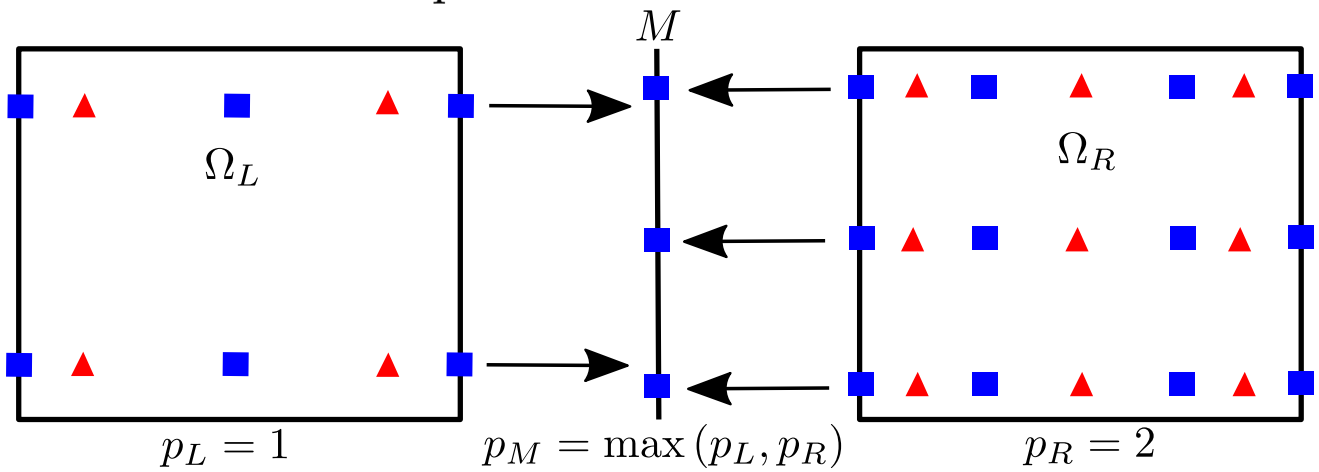
SP FP Legendre xi-direction FP Legendre eta-direction

Notes

- These positions are for **quadrilaterals/hexahedral** elements.
- Positions are **different for triangle/tetrahedral** elements.

Local polynomial adaptation: Mortar Element Method

- When using local p-adaptation, interfaces between elements with different degrees occur.
- Interface FP are not located at same positions between the 2 elements:



- A fictive interface, called a Mortar, is introduced where interface treatment is done.
- For convective fluxes for instance it is done in 3 steps:

1. $\mathbf{Q}^L$ projected on M to get $\mathbf{Q}^{M,L}$ and $\mathbf{Q}^{M,R} = \mathbf{Q}^R$ since $p_M = p_R$.
2. Riemann problem solved between $\mathbf{Q}^{M,L}$ and $\mathbf{Q}^{M,R}$ to get fluxes on Mortar FP noted $\mathbf{F}^M$.
3. $\mathbf{F}^M$ projected back on Ω_L to get $\mathbf{F}^L$ and $\mathbf{F}^R = \mathbf{F}^M$ since $p_M = p_R$.

Local polynomial adaptation: setting p distribution

Polynomial distribution within each element can be:

- Set by hand but it is not really practical.
- Set using a sensor based on physical quantities. For instance with norm of density gradient:

Sensor value in element Ω_e

Averaging duration

$$\theta_e = \frac{1}{t_{\text{avg}}|\Omega_e|}\int_t^{t+t_{\text{avg}}}\int_{\Omega_e}|\nabla\rho|\,d\Omega_e dt \quad \text{and} \quad \widetilde{\theta}_e = \frac{\theta_e}{\max\limits_{\Omega}(\theta_e)}$$

Volume of Ω_e

Normalized sensor value in Ω_e

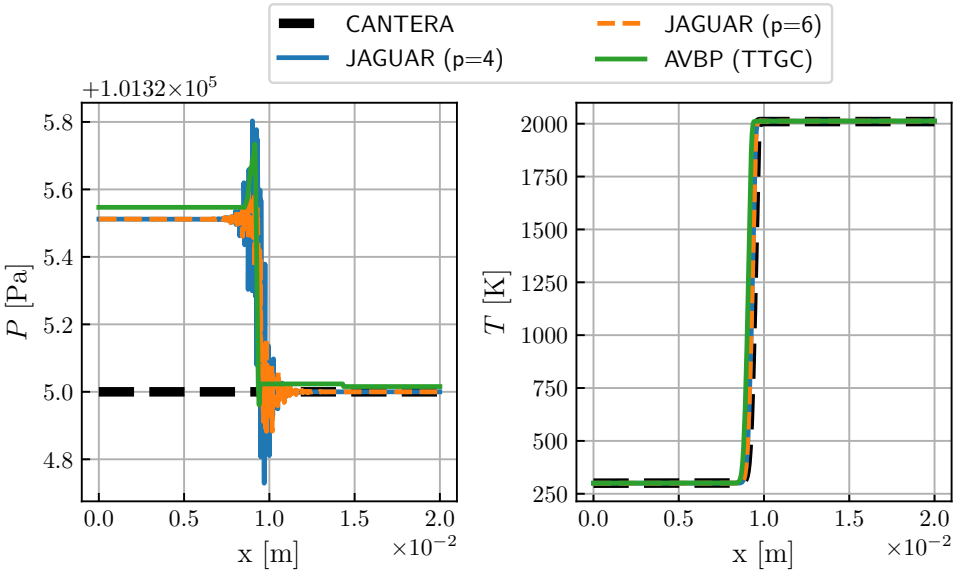
- Polynomial distribution is then set using:

$$p_e = p_{min} + \text{INT}\left[\tanh\left(\alpha\widetilde{\theta}_e\right)(p_{max} - p_{min})\right]$$

p_{min}, p_{max} are set by the user and α is here to avoid large degree jumps between elements.

Recent combustion simulations with JAGUAR

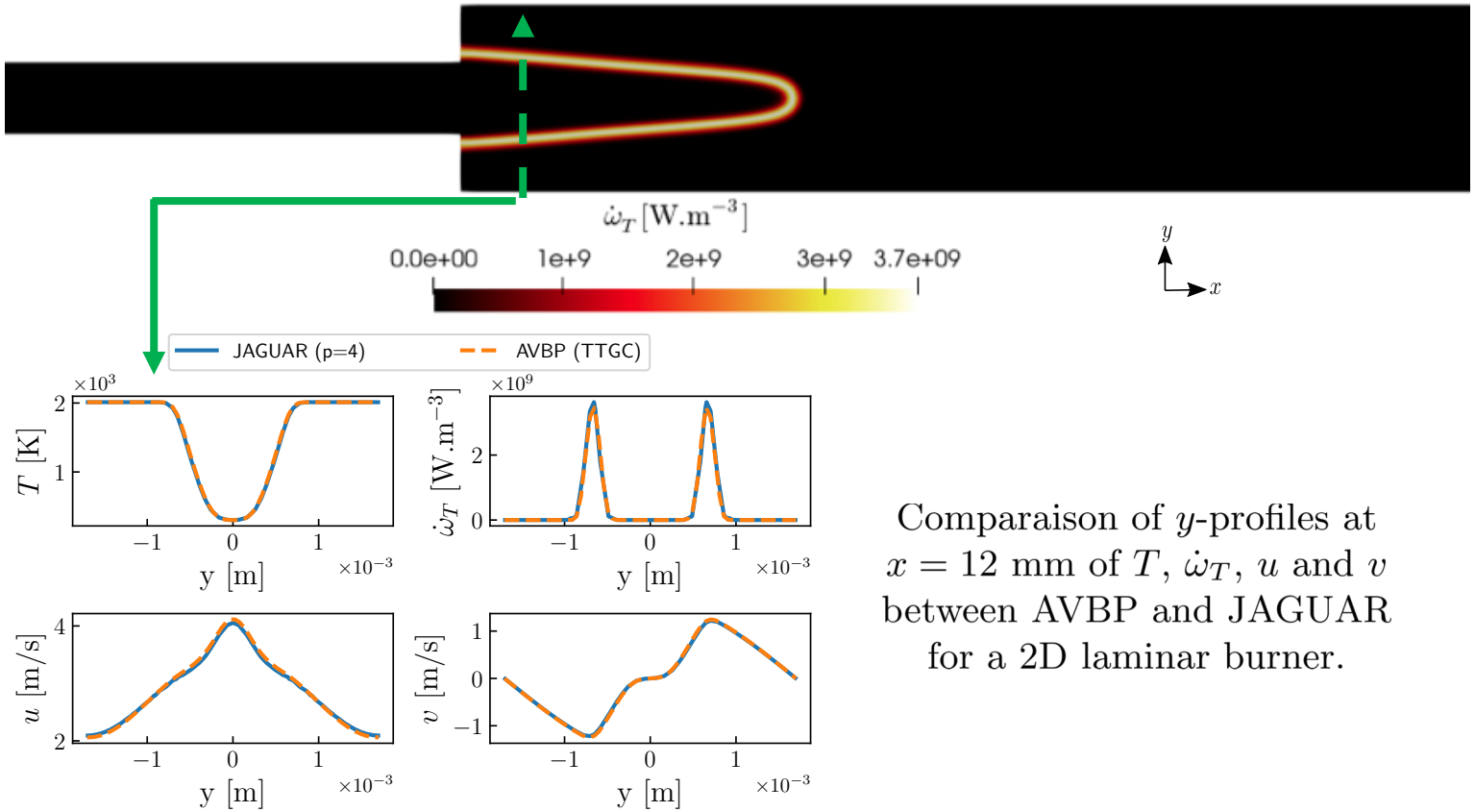
1D laminar flame



Comparaison of P and T between CANTERA, AVBP and JAGUAR for a 1D CH4/Air flame.

SD results were in good agreement with already existing methods

2D laminar burner



Comparaison of y-profiles at $x = 12$ mm of T, $\dot{\omega}_T$, u and v between AVBP and JAGUAR for a 2D laminar burner.

SD results were in good agreement with already existing methods

10. Marchal et al, Extension of the Spectral Difference method to combustion, Submitted to Journal of Computational Physics. https://arxiv.org/pdf/2112.09636.pdf

High-order methods: general principle

- At first glance, high-order space discretization **could mean an increase of the stencil.**
- But **large stencils** are **hard to use with unstructured meshes** which are essential for complex geometries.
- Thus, high-order methods will have to use a **compact stencil.** Then, they will increase the number of degrees of freedom (DOF) inside each mesh element and most of them follow the same process:

  1. Define a **high-order representation of the variables** (mostly polynomial) inside each mesh element using values at the DOF and a **high-order interpolation procedure.**
  2. At element boundaries, reconstructed data are not equal: **Riemann solvers** (for convective fluxes) and **Diffusion schemes** (for diffusive fluxes) are used to **handle these discontinuities ensuring conservation.**

Best interest of these methods: **manage both mesh refinement** noted h and the **degree of the local representation** of the solution noted p.

Conclusion on polynomial order influence

- Combining accuracy and computational time analysis, for the same level of error, it is faster to use less DOF with high p rather than more DOF with low p.
- For instance, $p = 2$ in 30900 DOF case and $p = 4$ in 20100 case have same error but the first one has a turnaroud time 67% higher than the second one.
- Cost per iteration is almost constant with p and not so far than the one of already existing methods.
- Main differences in computational cost come from time step restrictions.