A comprehensive framework for high-order solver verification

5^{th} High-order CFD workshop

Farshad Navah
PhD Candidate

Siva Nadarajah
Associate Professor

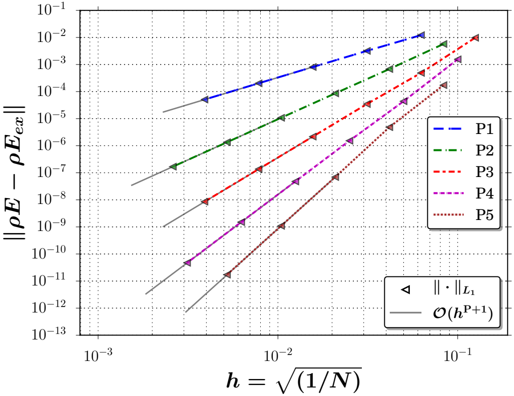
McGill University
Computational Aerodynamics Group

January 6^{th}, 2018

Outline

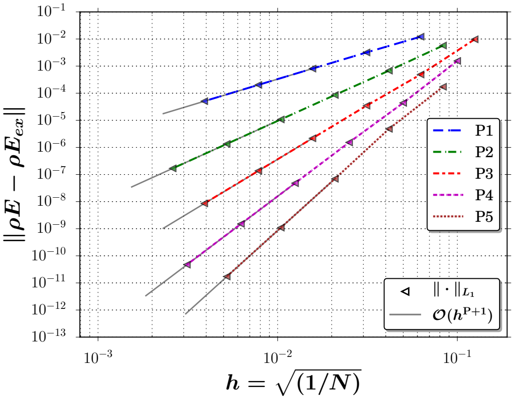
Motivation for higher orders of accuracy:

Generating lower error for the same effort



Motivation for higher orders of accuracy:

Generating lower error for the same effort



But, are we getting what we expect for a given solver?

Code Verification : Gathering cumulative evidence that the implemented discrete equations recover the model PDEs as $\mathrm{DOFs} \to \infty$

Code Verification : Gathering cumulative evidence that the implemented discrete equations recover the model PDEs as $\mathrm{DOFs} \rightarrow \infty$

Importance:

- around 10 serious faults / 1000 lines [1]

Code Verification : Gathering cumulative evidence that the implemented discrete equations recover the model PDEs as $\mathrm{DOFs} \rightarrow \infty$

Importance:

- around 10 serious faults / 1000 lines [1]

Method : as $\mathrm{DOFs} \rightarrow \infty$

- Solution(DOFs) $\rightarrow$ Exact solution ?
- Observed order of accuracy $=$ Formal order of accuracy?

$\Rightarrow$ 2 ingredients: Exact solution & Orders of accuracy

Code verification via Solution verification[1]

Can we estimate output errors to verify high-order codes?

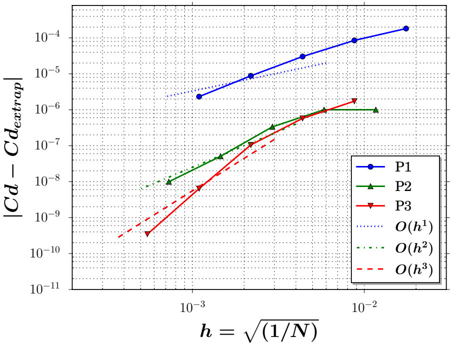


Figure: RANS-SA–CPR/DG results [2] for TMR-ZPG [3]

Sustained convergence, **but**

[1] Solution verification: Estimating output error and its observed order of accuracy for a given problem; See [2]

Code verification via Solution verification[1]

Can we estimate output errors to verify high-order codes?

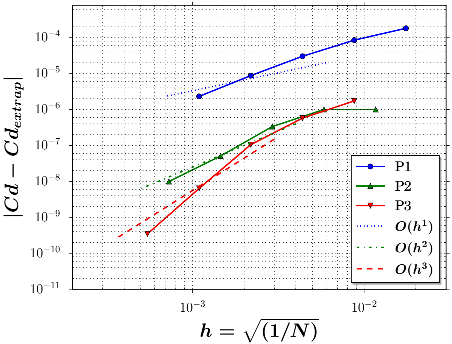


Figure: RANS-SA–CPR/DG results [2] for TMR-ZPG [3]

Sustained convergence, **but**

Observed order of accuracy $\neq$ Expected order of accuracy $\mathcal{O}(h^{\mathrm{P}})$

[1] Solution verification: Estimating output error and its observed order of accuracy for a given problem; See [2]

Code verification via solution verification

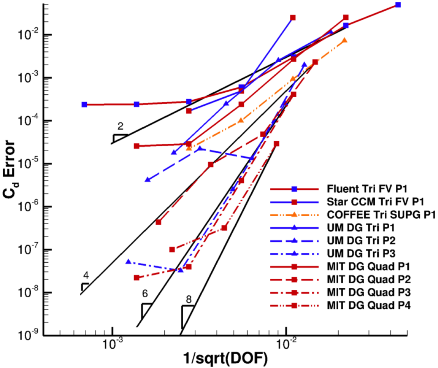


Figure: Joukowski RANS [4] HOW04

Code verification via solution verification

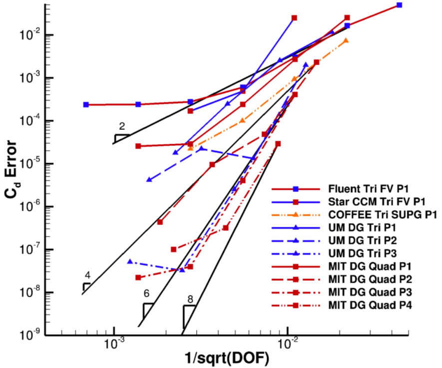


Figure: Joukowski RANS [4] HOW04

Challenges:

- Universal highly-accurate "truth value"?
- Grid set producing asymptotic range for all P & in all codes?

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set
- Irreducible high complexity (curved grids, multiple BCs, output computation, singularities, etc.)

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set
- Irreducible high complexity (curved grids, multiple BCs, output computation, singularities, etc.)
- Hardly allows for debugging via simplification/elimination

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set
- Irreducible high complexity (curved grids, multiple BCs, output computation, singularities, etc.)
- Hardly allows for debugging via simplification/elimination
- For RANS, depends on wall-distance computation

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set
- Irreducible high complexity (curved grids, multiple BCs, output computation, singularities, etc.)
- Hardly allows for debugging via simplification/elimination
- For RANS, depends on wall-distance computation
- No systematic guarantee on the verification of all model terms

[2] Estimating output error and its observed order of accuracy for a given problem

Code verification via solution verification

Shortcomings of solution verification[2] in verifying high-order codes:

- For RANS, it heavily depends on the grid set
- Irreducible high complexity (curved grids, multiple BCs, output computation, singularities, etc.)
- Hardly allows for debugging via simplification/elimination
- For RANS, depends on wall-distance computation
- No systematic guarantee on the verification of all model terms
- Local quantities (outputs), not global norms

[2] Estimating output error and its observed order of accuracy for a given problem

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

- Debugging via simplification/elimination : arbitrary PDEs, solutions & BCs

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

- Debugging via simplification/elimination : arbitrary PDEs, solutions & BCs
- Enables cross-solver comparisons via model standardization

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

- Debugging via simplification/elimination : arbitrary PDEs, solutions & BCs
- Enables cross-solver comparisons via model standardization
- Needs minimal changes to the code: a few calls to the same routine

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

- Debugging via simplification/elimination : arbitrary PDEs, solutions & BCs
- Enables cross-solver comparisons via model standardization
- Needs minimal changes to the code: a few calls to the same routine
- Manufactured solutions (MSs) tested for orders up to sixth and collocated schemes

Proposed high-order code verification framework via the method of manufactured solutions (MMS):

- Debugging via simplification/elimination : arbitrary PDEs, solutions & BCs
- Enables cross-solver comparisons via model standardization
- Needs minimal changes to the code: a few calls to the same routine
- Manufactured solutions (MSs) tested for orders up to sixth and collocated schemes
- Quantified sensitivity to model terms

The framework:

Repository: `https://github.com/fanav/Verification_MMS_Tools`

Branches:

- Free Flows: (fundamental cases) `FreeFlow_Euler/NS/RANS-SA`
- Wall Flows: (advanced cases) `WallFlow_Euler/NS/RANS-SA`

Files in each branch:

- Ipython notebook: generate forcing functions and manufactured solutions
- C file: pre-defined MSs and models, facilitated debugging via simplification/elimination
- arXiv manuscript: full description of code, BCs, cases and results

High-order codes

McGill-CPR/DG

- **Governing equations**: compressible RANS + original & negative Spalart-Allmaras (SA) model [5]
- **Discretization scheme**: high-order correction procedure via flux reconstruction (CPR) - DG correction functions
- **Numerical Flux**: Roe for inviscid terms and BR2[3] for viscous terms
- **Elements**: Quads
- **Nodes**: Gauss-Legendre-Lobatto (GLL)
- **BCs**: See [7] and [8]

MIT-SANS

- **Governing equations**: compressible RANS + original & negative SA model [5]
- **Discretization scheme**: high-order DG
- **Numerical Flux**: Roe for inviscid terms and BR2 for viscous terms
- **Elements**: Quads/Tris
- **Nodes**: Gauss-Legendre (GL) with/without[4] over-collocated-integration

[3] Tensor-product operators; see [6]

[4] No significant difference observed for the cases reported in here.

Conservation equations

The governing equations for compressible RANS-modelled flows:

Continuity:

$$\partial_t \rho + \partial_j(\rho u_j) = 0$$

Momentum:

$$\partial_t(\rho u_i) + \partial_j(\rho u_j u_i + p\delta_{ij}) - \partial_j \tau_{ij} = 0$$

Energy:

$$\partial_t(\rho E) + \partial_j(\rho u_j H) - \partial_j(u_i \tau_{ij} + q_j) = 0$$

Spalart-Allmaras turbulence model:

$$\partial_t(\rho\tilde{\nu}) + \partial_j(\rho u_j \tilde{\nu}) - \frac{1}{\sigma}\partial_j[(\mu + \rho\tilde{\nu} f_n)\partial_j \tilde{\nu}] = \frac{c_{b2}}{\sigma}\rho\ \partial_j\tilde{\nu}\ \partial_j\tilde{\nu} + \rho\mathcal{P} - \rho\mathcal{D} - \frac{1}{\sigma}(\nu + \tilde{\nu} f_n)\partial_j(\rho\partial_j\tilde{\nu})$$

Negative Spalart-Allmaras turbulence model definitions ICCFD7-1902

$$\mu_t = \rho \nu_t \begin{cases} \rho \tilde{\nu} f_{v1} & \tilde{\nu} \geq 0, \\ 0 & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \tilde{\nu}/\nu, \quad c_{v1} = 7.1$$

$$\mathcal{P} = \begin{cases} c_{b1} \tilde{s} \tilde{\nu} & \tilde{\nu} \geq 0, \\ c_{b1} s \tilde{\nu} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad c_{b1} = 0.1355, \quad f_n = \begin{cases} 1 & \tilde{\nu} \geq 0, \\ \dfrac{c_{n1} + \chi^3}{c_{n1} - \chi^3} & \tilde{\nu} < 0, \end{cases}$$

$$\tilde{s} = \begin{cases} s + \bar{s} & \bar{s} \geq -c_{v2} s, \\ s + \dfrac{s(c_{v2}^2 s + c_{v3} \bar{s})}{(c_{v3} - 2c_{v2}) s - \bar{s}} & \bar{s} < -c_{v2} s, \end{cases} \quad \text{where} \quad \begin{array}{l} \bar{s} = \dfrac{\tilde{\nu} f_{\nu 2}}{\kappa^2 d_w^2}, \ f_{v2} = 1 - \dfrac{\chi}{1 + \chi f_{v1}} \\ c_{v2} = 0.7, \ c_{v3} = 0.9, \ \kappa = 0.41 \end{array}$$

$$\mathcal{D} = \begin{cases} c_{w1} f_w \dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} \geq 0, \\ -c_{w1} \dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad \begin{array}{l} c_{w1} = \dfrac{c_{b1}}{\kappa^2} + \dfrac{1 + c_{b2}}{\sigma} \\ c_{b2} = 0.622, \quad \sigma = 2/3 \end{array}$$

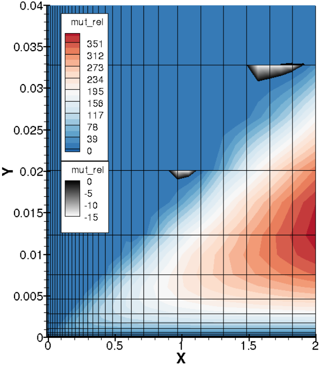
where d_w is the distance to the closest wall.

See [7] for the full set of definitions.

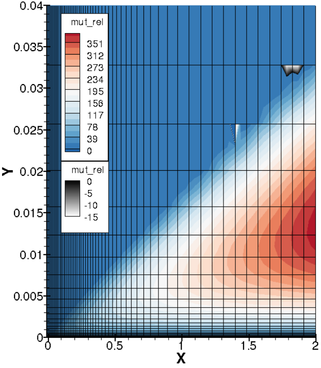
The role of the Negative SA model

The solution is stable despite the occurrence of negative $\tilde{\nu}$ values at the edge of the boundary layer on coarse grids

CPR-DG results [2] for ZPG from TMR [3]:



P2-L0



P2-L1

Part I: Free Flows (no walls)

Reference for theoretical details and extensive results:
"A comprehensive high-order solver verification methodology for free fluid flows", Navah F., Nadarajah, S. : [7], http://arxiv.org/abs/1712.09478

Reference for software framework:
"A framework for the verification of high-order CFD solvers via the method of manufactured solutions (MMS)", Navah F. : [9], `https://github.com/fanav/Verification_MMS_Tools`,
Branch: `FreeFlow_Euler/NS/RANS-SA`

Manufactured solutions for Free Flows (Part I)

$$\rho^{\mathrm{MS}} \equiv \rho_0 + \rho_x \sin(a_{\rho_x}\pi x/L) + \rho_y \cos(a_{\rho_y}\pi y/L) + \rho_{xy} \cos(a_{\rho_{xy}}\pi x/L)\cos(a_{\rho_{xy}}\pi y/L),$$

$$u^{\mathrm{MS}} \equiv u_0 + u_x \sin(a_{u_x}\pi x/L) + u_y \cos(a_{u_y}\pi y/L) + u_{xy} \cos(a_{u_{xy}}\pi x/L)\cos(a_{u_{xy}}\pi y/L),$$

$$v^{\mathrm{MS}} \equiv v_0 + v_x \cos(a_{v_x}\pi x/L) + v_y \sin(a_{v_y}\pi y/L) + v_{xy} \cos(a_{v_{xy}}\pi x/L)\cos(a_{v_{xy}}\pi y/L),$$

$$p^{\mathrm{MS}} \equiv p_0 + p_x \cos(a_{p_x}\pi x/L) + p_y \sin(a_{p_y}\pi y/L) + p_{xy} \cos(a_{p_{xy}}\pi x/L)\cos(a_{p_{xy}}\pi y/L),$$

$$\tilde{\nu}^{\mathrm{MS}} \equiv \tilde{\nu}_0 + \tilde{\nu}_x \cos(a_{\tilde{\nu}_x}\pi x/L) + \tilde{\nu}_y \cos(a_{\tilde{\nu}_y}\pi y/L) + \tilde{\nu}_{xy} \cos(a_{\tilde{\nu}_{xy}}\pi x/L)\cos(a_{\tilde{\nu}_{xy}}\pi y/L),$$

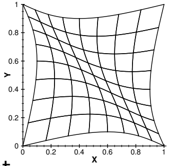
Features

- Domain $\Omega = [0,1]^2$
- BCs: in/out flow via numerical fluxes

MS-1: Inviscid subsonic flow
(See: Extra slides at the end)

MS-2: Inviscid supersonic flow on curved grids

$(\cdot)$	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a_{(\cdot)x}$	$a_{(\cdot)y}$	$a_{(\cdot)xy}$
ρ	2.7	0.9	−0.9	1.0	1.5	1.5	1.5
u	2.0	0.7	0.7	0.4	1.0	1.0	1.0
v	2.0	0.4	0.4	0.4	1.0	1.0	1.0
p	2.0	1.0	1.0	0.5	1.0	1.0	1.5
$\tilde{\nu}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

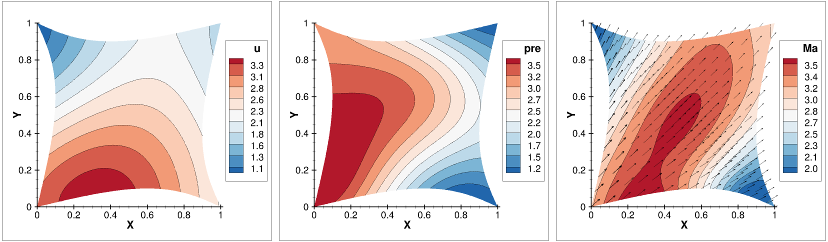


Verifying non-affine mapping treatment

Domain deformation:

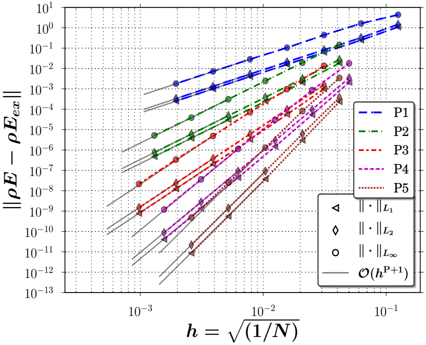
$$x = \mathcal{X} + 0.1\sin(\pi\mathcal{X} + \pi\mathcal{Y}); \;\; y = \mathcal{Y} + 0.1\sin(\pi\mathcal{X} + \pi\mathcal{Y}) \quad (\mathcal{X},\mathcal{Y}) \in \Omega_0 = [0,1]^2$$

Some solution fields:

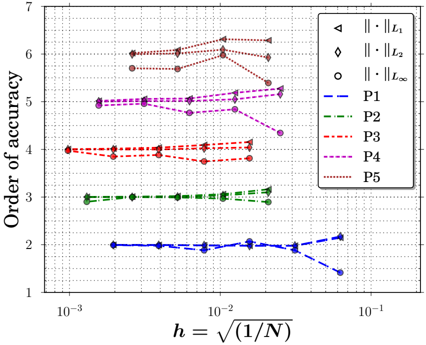


MS-2: Inviscid supersonic flow on curved grids

Some typical results:



ρE



ρE

MS-3: Laminar flows
(See: Extra slides at the end)

MS-4: Turbulent flows – RANS + **original** SA model

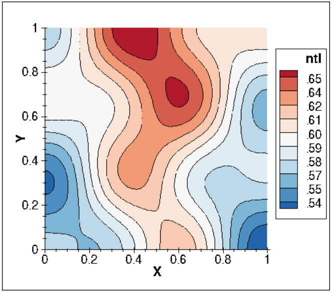
$(\cdot)$	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a_{(\cdot)x}$	$a_{(\cdot)y}$	$a_{(\cdot)xy}$
ρ	1.0	0.1	−0.2	0.1	1.0	1.0	1.0
u	2.0	0.3	0.3	0.3	3.0	1.0	1.0
v	2.0	0.3	0.3	0.3	1.0	1.0	1.0
p	10.0	1.0	1.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	0.6	−0.03	−0.02	0.02	2.0	1.0	3.0

Same solution fields as MS-3 + $\tilde{\nu}$

$$5.5 \times 10^{-1} \lessapprox \tilde{\nu} \lessapprox 6.5 \times 10^{-1};\ \mu = 1 \times 10^{-3}$$

$\mu_{\text{eff}} \approx 1 \times 10^{-1} \Rightarrow$ fair Inviscid/Viscous

(See extra slides at the end on MS-3 inviscid/viscous balance)



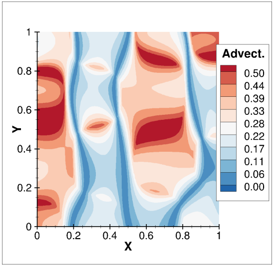
MS-4: Turbulent flows – RANS + **original** SA model

SA Forcing function sensitivity analysis $|\mathrm{S}_{\mathrm{term}}|/\mathrm{S}_{\mathrm{SA}}^{\mathrm{sum}}$:

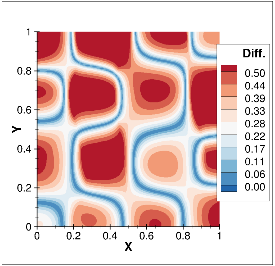
$$\mathrm{S}_{\mathrm{SA}}^{\mathrm{sum}} = \sum_{term} |\mathrm{S}_{\mathrm{term}}| = \Big|\underbrace{\partial_j(\rho u_j \tilde{\nu})}_{\text{Advection}}\Big| + \Big|\underbrace{-\,\partial_j\left(\frac{1}{\sigma}(\mu+\rho\tilde{\nu})\partial_j\tilde{\nu}\right)}_{\text{Diffusion}}\Big| + \Big|\underbrace{\rho\, c_{b1}(1-f_{t2})\,\tilde{s}\,\tilde{\nu}}_{\text{Production}}\Big| +$$

$$\Big|\underbrace{-\rho\left(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right)\frac{\tilde{\nu}^2}{d_w^2}}_{\text{Destruction}}\Big| + \Big|\underbrace{\frac{c_{b2}}{\sigma}\rho\,\partial_j\tilde{\nu}\,\partial_j\tilde{\nu}}_{\text{Distribution}}\Big| + \Big|\underbrace{\frac{1}{\sigma}(\nu+\tilde{\nu}f_n)\,\partial_j(\rho\,\partial_j\tilde{\nu})}_{\text{Conservation}}\Big|.$$

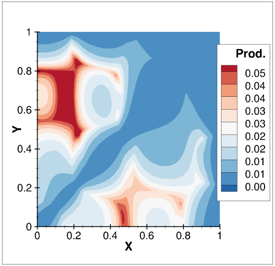
Some examples:



Advection



Diffusion

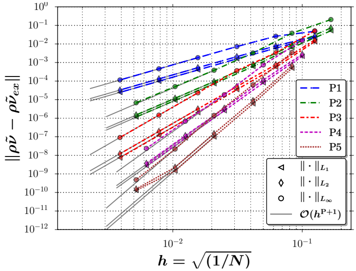


Production

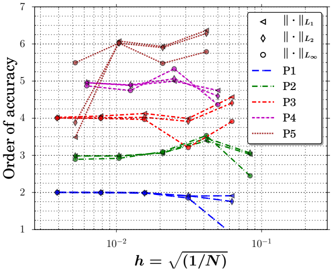
MS-4: Turbulent flows – RANS + **original** SA model

Sensitivity analysis: Can the MMS pick a tiny bug in the production term?

Production $= (1 + 10^{-7})\frac{c_{b2}}{\sigma}\rho\,\partial_j\tilde{\nu}\,\partial_j\tilde{\nu}$



Discretization error $\rho\tilde{\nu}$



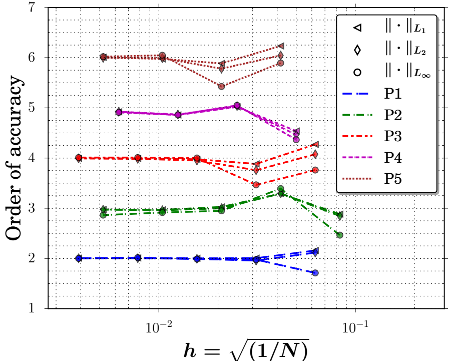
Order of accuracy $\rho\tilde{\nu}$

Yes, we can.

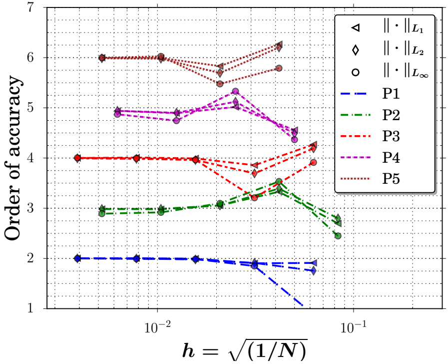
$\Rightarrow$ It is good to include a relatively high P in verification.

MS-4: Turbulent flows – RANS + **original** SA model

Some typical results in L_1, L_2 and L_∞



ρE



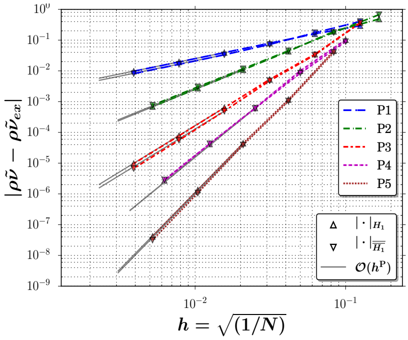
$\rho\tilde{\nu}$

MS-4: Turbulent flows – RANS + **original** SA model

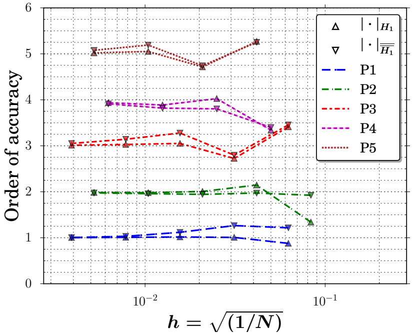
H_1: uncorrected derivatives $(\partial_q Q_k)$

$\overline{\overline{H_1}}$: fully corrected BR2 derivatives $(\overline{\overline{\partial_q Q_k}})$

Some typical results in H_1 & $\overline{\overline{H_1}}$



$\rho\tilde{\nu}$

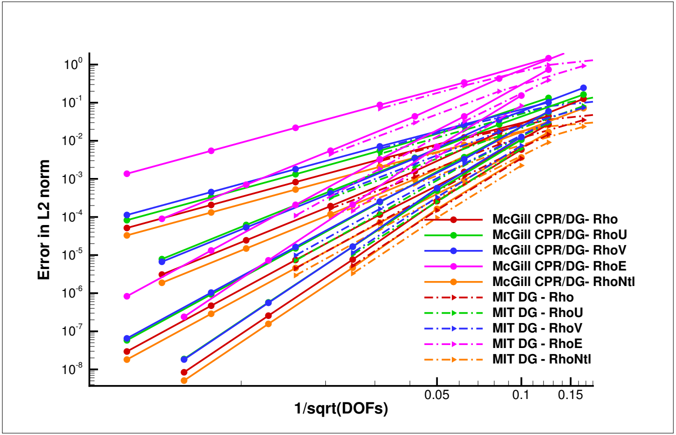


$\rho\tilde{\nu}$

MS-4: Turbulent flows – RANS + **original** SA model

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

P1-P4

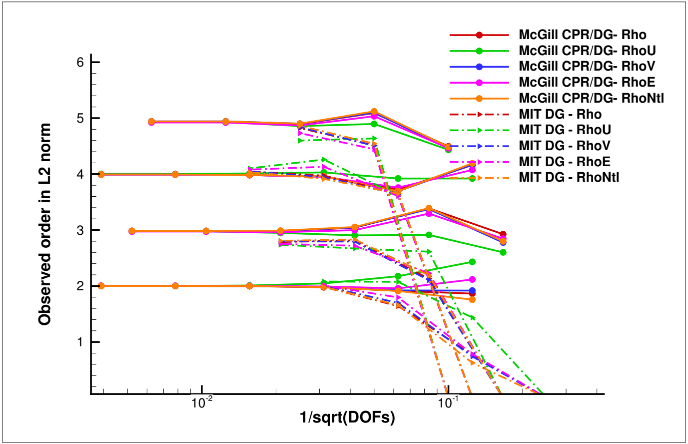


$\Rightarrow$ SANS results slightly more accurate.

MS-4: Turbulent flows – RANS + **original** SA model

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

P1-P4



⇒ CPR results converge faster.

MS-5: Turbulent flows – RANS + **negative** SA model
(See: Extra slides at the end)

Part I, Free Flows verification summary

Property	**Feature**	MS-1	MS-2	MS-3	MS-4	MS-5	**Cum.**
Re	Inviscid	✓	✓	✓	✓	✓	✓
	Viscous	✗	✗	✓	✓	✓	✓
	Turbulent	✗	✗	✗	✓	✓	✓
Ma	Supersonic	✗	✓	✗	✗	✗	✓
	Transonic	✗	✗	✗	✗	✗	✗
	Subsonic	✓	✗	✓	✓	✓	✓
Boundary Conditions	Riemann	✓	✓	✓	✓	✓	✓
	Viscous	✗	✗	✓	✓	✓	✓
	Slip Wall	✗	✗	✗	✗	✗	✗
	No-slip Wall	✗	✗	✗	✗	✗	✗
Mapping	Curved Elements	✗	✓	✗	✗	✗	✓

Part II: Wall flows (advanced cases)

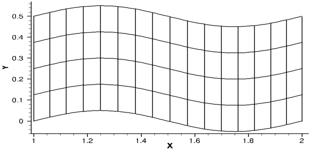
Reference for theoretical details and extensive results:
"On the verification of CFD solvers of all orders of accuracy on curved wall-bounded domains and for realistic RANS flows", Navah F., Nadarajah S. : [8],
http://arxiv.org/abs/1712.09478

Reference for software framework:
"A framework for the verification of high-order CFD solvers via the method of manufactured solutions (MMS)", Navah F. : [9],
`https://github.com/fanav/Verification_MMS_Tools`,
Branch: `WallFlow_Euler/NS/RANS-SA`

MS-1: Inviscid transonic flow on curved slip wall

$$\rho^{\mathrm{MS}} = \rho_0 + \mathcal{Y}^2, \qquad u^{\mathrm{MS}} = u_w + \mathcal{Y},$$
$$v^{\mathrm{MS}} = \frac{\partial y}{\partial x}\, u^{\mathrm{MS}}, \qquad p^{\mathrm{MS}} = p_0 + \mathcal{Y}^2,$$

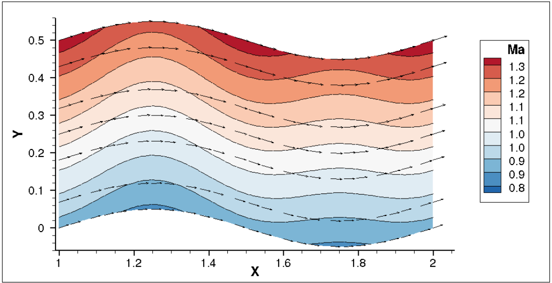
where $\rho_0 = 1.0$, $p_0 = 1.0$ and $u_w = 1.0$ is the horizontal velocity component at the wall ($\mathcal{Y} = 0$)



Domain deformation: $x = \frac{4}{3}\left(\mathcal{X}^2 - \frac{1}{4}\right) + 1$; $y = \mathcal{Y} + 0.05\sin(2\pi x)$; $(\mathcal{X}, \mathcal{Y}) \in \Omega_0 = [0.5, 1.0] \times [0.0, 0.5]$

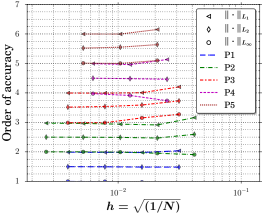
Analytical wall normal: $\boldsymbol{n}_w = \frac{1}{\sqrt{(\frac{\partial y}{\partial x})^2 + (-1)^2}} \left(\frac{\partial y}{\partial x}\,\mathbf{e}_1 - \mathbf{e}_2\right)$

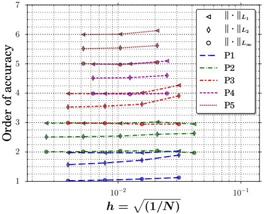
Some solution fields

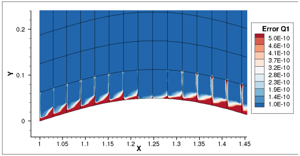


MS-1: Inviscid transonic flow on curved slip wall

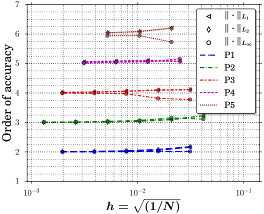
Using **inexact wall normals** (from isoparametric mapping) in reflecting (non-penetration) BC:



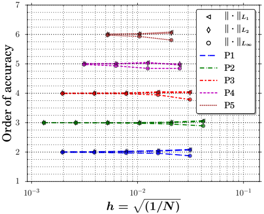




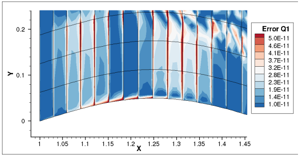
Using **exact wall normals** in reflecting (non-penetration) BC:



ρu



ρv



ρu Error at wall

MS-2: Laminar flow on curved no-slip wall

(See: Extra slides at the end)

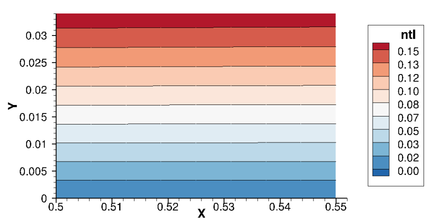
MS-3: Transonic turbulent boundary layer flow
(See: Extra slides at the end)

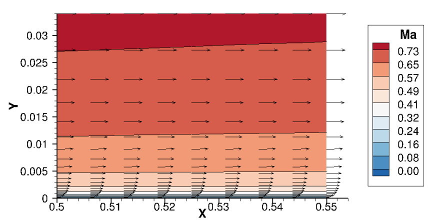
MS-4: Realistic turbulent boundary layer flow

Adopted from Oliver et al. (2012) [10] ; See [8] for full details

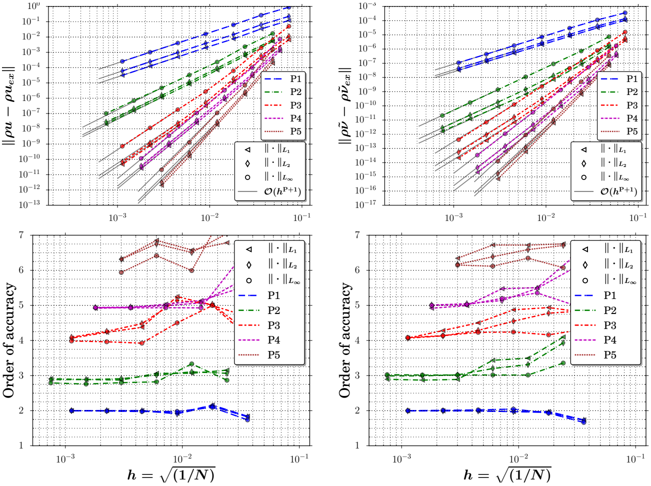
$$176,690 \leq \mathrm{Re}_x \leq 194,359$$

Some solution fields:





MS-4: Realistic turbulent boundary layer flow



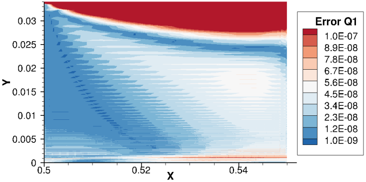
ρu $\rho\tilde{\nu}$

⇒ Converges to the formal order, but slower and less clean than cases in Part I.

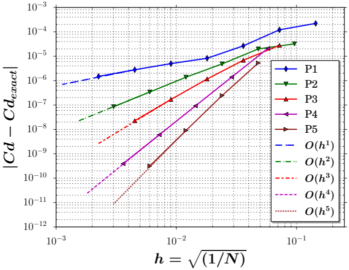
MS-4: Realistic turbulent boundary layer flow

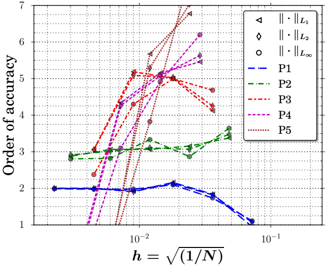
Code verification vs. Solution verification

A bug in the upper boundary:



Does not affect the output (C_d), but does affect the variable errors.

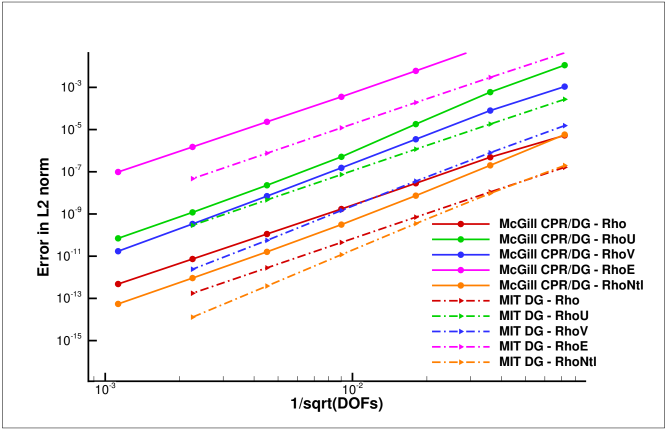




MS-4: Realistic turbulent boundary layer flow

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

P3



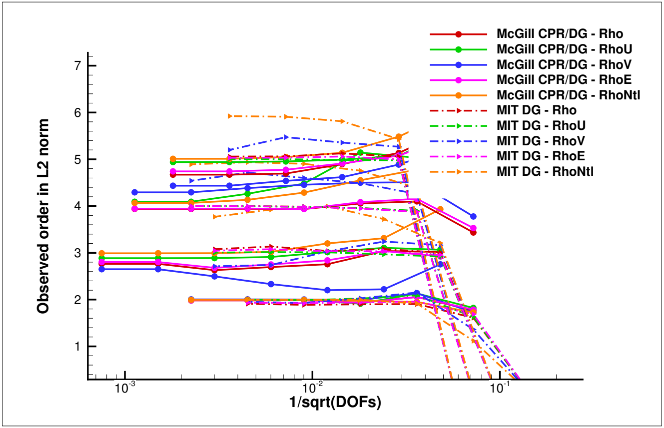
⇒ SANS results are more accurate (unexpected since results were close in Part I)

under investigation: post-processing, BCs, ...?

MS-4: Realistic turbulent boundary layer flow

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

P1-P4

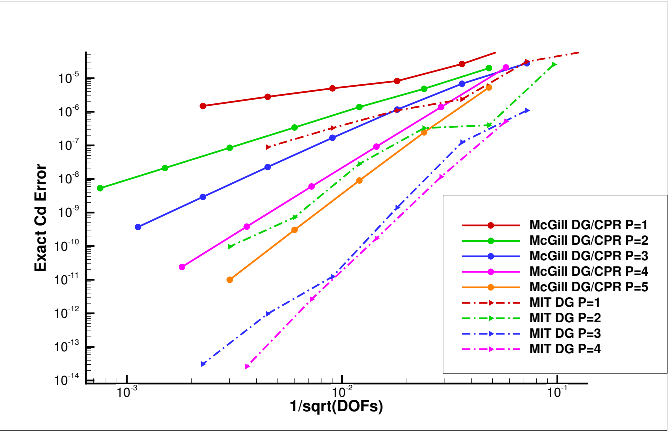


$\Rightarrow$ Convergence to $\approx$ formal orders but less cleanly than in Part I

MS-4: Realistic turbulent boundary layer flow

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

Exact C_d error, $C_d^{\mathrm{ex}} = 3.6013213414944 \times 10^{-03}$

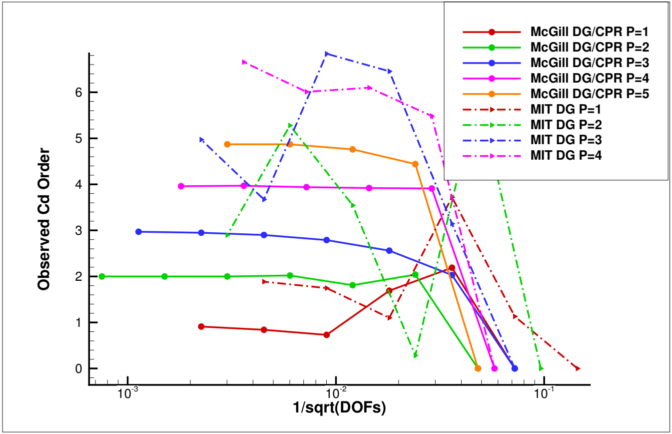


⇒ CPR: very steady convergence; SANS: more accurate (adjoint-consistent)

MS-4: Realistic turbulent boundary layer flow

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-quads

Observed C_d orders



$\Rightarrow$ CPR: very steady convergence to $\mathcal{O}(h^P)$; SANS: erratic convergence to higher orders

Part II, Wall-bounded Flows verification summary

Property	**Feature**	MS-1	MS-2	MS-3	MS-4	**Cum.**
Re	Inviscid	✓	✓	✓	✓	✓
	Viscous	✗	✓	✓	✓	✓
	Turbulent	✗	✗	✓	✓	✓
Ma	Supersonic	✓	✗	✓	✗	✓
	Transonic	✓	✗	✓	✗	✓
	Subsonic	✓	✓	✓	✓	✓
Boundary Conditions	Riemann	✓	✓	✓	✓	✓
	Viscous	✗	✓	✓	✓	✓
	Slip Wall	✓	✓	✓	✓	✓
	No-slip Wall	✗	✓	✓	✓	✓
Mapping	Curved Domain	✓	✓	✗	✗	✓

Special thanks to Marshall Galbraith from MIT

and to Jean-Marie Le Gouez from ONERA

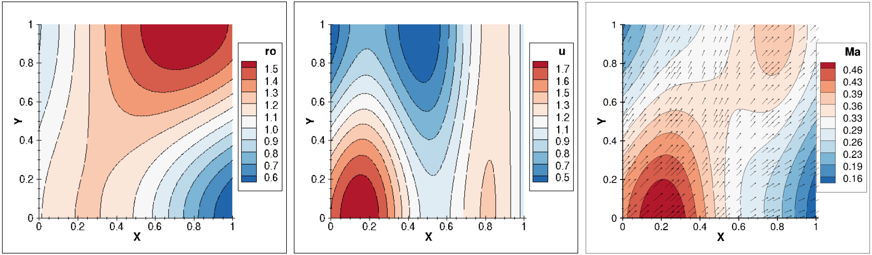
Thank you for your attention!

Part I. extra slides

MS-1: Inviscid subsonic flow

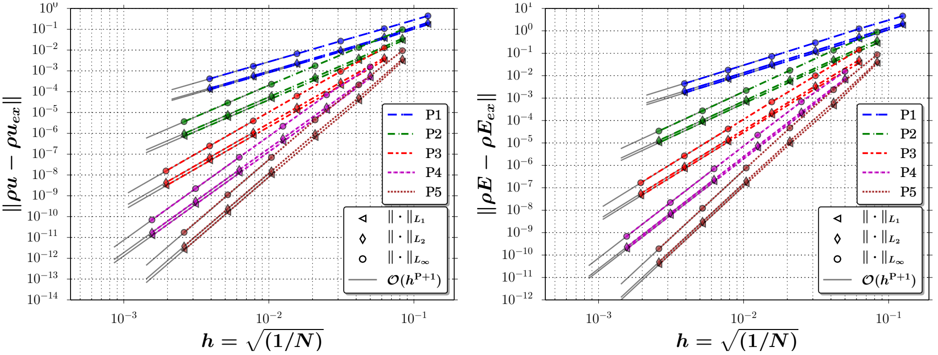
$(\cdot)$	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a_{(\cdot)x}$	$a_{(\cdot)y}$	$a_{(\cdot)xy}$
ρ	1.0	0.3	−0.2	0.3	1.0	1.0	1.0
u	1.0	0.3	0.3	0.3	3.0	1.0	1.0
v	1.0	0.3	0.3	0.3	1.0	1.0	1.0
p	18.0	5.0	5.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Some solution fields:



MS-1: Inviscid subsonic flow

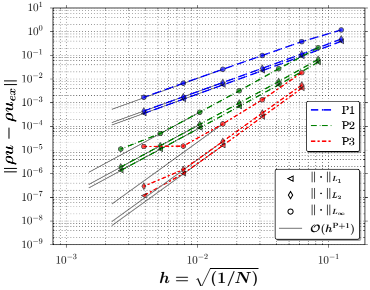
Some typical results:



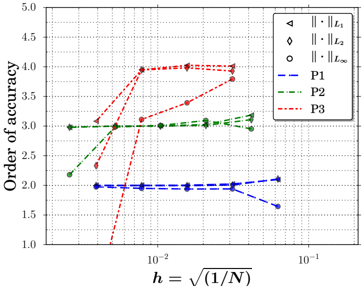
MS-2: Inviscid supersonic flow on curved grids

Importance of L_∞ norm:

Introducing a spurious boundary condition of $(\rho u)^{BC} = 1.000001 \times (\rho^{\mathrm{MS}} u^{\mathrm{MS}})|_{BC}$ at $(x, y) = (0, 0)$



Discretization error



Order of accuracy

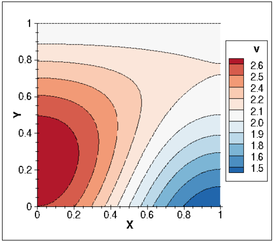
$\Rightarrow$ L_∞ always picks the bug faster than L_1 and L_2

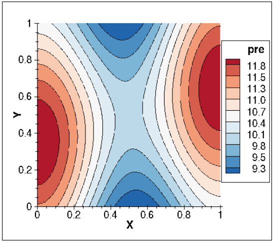
MS-3: Laminar flows

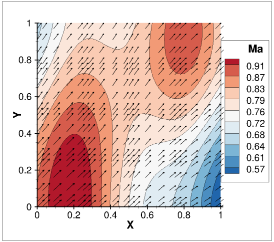
$(\cdot)$	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a_{(\cdot)x}$	$a_{(\cdot)y}$	$a_{(\cdot)xy}$
ρ	1.0	0.1	−0.2	0.1	1.0	1.0	1.0
u	2.0	0.3	0.3	0.3	3.0	1.0	1.0
v	2.0	0.3	0.3	0.3	1.0	1.0	1.0
p	10.0	1.0	1.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	0.0	0.0	0.0	0.0	0.0	0.0	0.0

$$\mu = 1\times10^{-1}$$

Some solution fields:

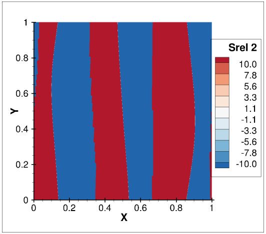




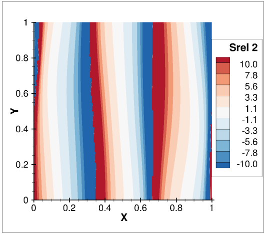


MS-3: Laminar flows

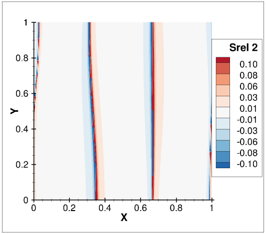
Forcing function balancing, $\mathrm{S}^{\mathrm{rel}}(\mathbf{x}) = \mathrm{S}^{\mathrm{inv}}(\mathbf{x})/\mathrm{S}^{\mathrm{vis}}(\mathbf{x})$, testing stability of diffusion schemes (IP, BR2, LDG, etc.):



$\mu = 1 \times 10^{-4}$



$\boldsymbol{\mu = 1 \times 10^{-1}}$

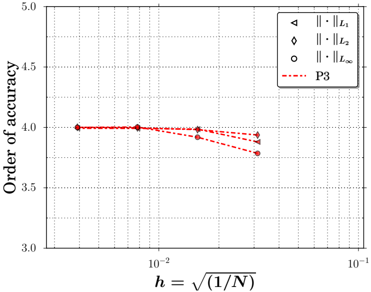


$\mu = 1 \times 10^{+2}$

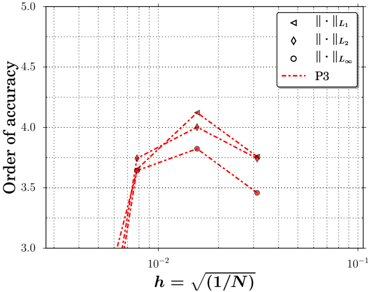
MS-3: Laminar flows

Importance of a fair forcing function balancing

Introducing a bug in the heat flux of the energy equation; the effect on P3 ρE orders:



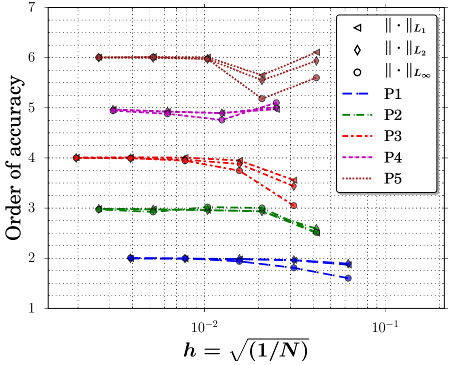
$\mu = 1 \times 10^{-4}$



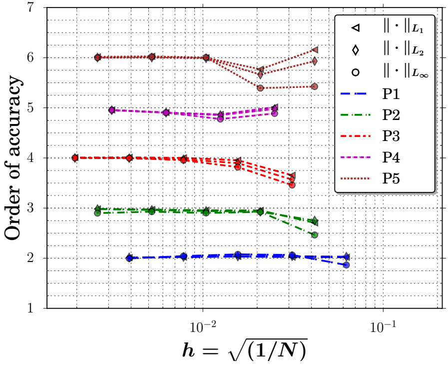
$\mu = 1 \times 10^{-1}$

MS-3: Laminar flows (L norms)

Some typical results in L_1, L_2 and L_∞



ρv



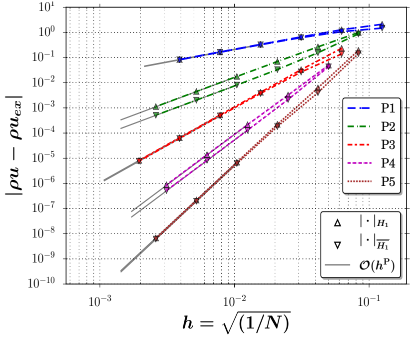
ρE

MS-3: Laminar flows (H_1 semi-norms)

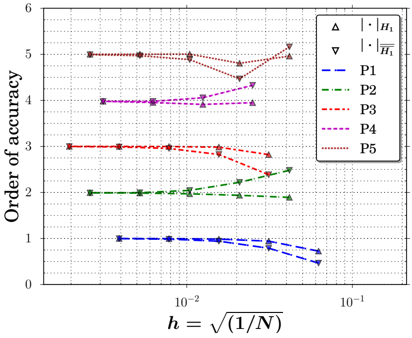
H_1: uncorrected derivatives ($\partial_q Q_k$)

$\overline{\overline{H_1}}$: fully corrected BR2 derivatives ($\overline{\overline{\partial_q Q_k}}$)

Some typical results:



ρu



ρu

Negative Spalart-Allmaras turbulence model definitions ICCFD7-1902

$$\partial_t(\rho\tilde{\nu}) + \partial_j(\rho u_j\tilde{\nu}) - \frac{1}{\sigma}\partial_j[(\mu + \rho\tilde{\nu}f_n)\partial_j\tilde{\nu}] = \frac{c_{b2}}{\sigma}\rho\,\partial_j\tilde{\nu}\,\partial_j\tilde{\nu} + \rho\mathcal{P} - \rho\mathcal{D} - \frac{1}{\sigma}(\nu + \tilde{\nu}f_n)\partial_j(\rho\partial_j\tilde{\nu})$$

$$\mu_t = \rho\nu_t \begin{cases} \rho\tilde{\nu}f_{v1} & \tilde{\nu} \geq 0, \\ 0 & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \tilde{\nu}/\nu, \quad c_{v1} = 7.1$$

$$\mathcal{P} = \begin{cases} c_{b1}\tilde{s}\tilde{\nu} & \tilde{\nu} \geq 0, \\ c_{b1}s\tilde{\nu} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad c_{b1} = 0.1355, \quad f_n = \begin{cases} 1 & \tilde{\nu} \geq 0, \\ \dfrac{c_{n1} + \chi^3}{c_{n1} - \chi^3} & \tilde{\nu} < 0, \end{cases}$$

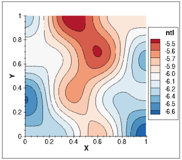
$$\tilde{s} = \begin{cases} s + \bar{s} & \bar{s} \geq -c_{v2}s, \\ s + \dfrac{s(c_{v2}^2 s + c_{v3}\bar{s})}{(c_{v3} - 2c_{v2})s - \bar{s}} & \bar{s} < -c_{v2}s, \end{cases} \quad \text{where} \quad \begin{array}{l} \bar{s} = \dfrac{\tilde{\nu}f_{\nu 2}}{\kappa^2 d_w^2}, \ f_{v2} = 1 - \dfrac{\chi}{1 + \chi f_{v1}} \\ c_{v2} = 0.7, \ c_{v3} = 0.9, \ \kappa = 0.41 \end{array}$$

$$\mathcal{D} = \begin{cases} c_{w1}f_w\dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} \geq 0, \\ -c_{w1}\dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad \begin{array}{l} c_{w1} = \dfrac{c_{b1}}{\kappa^2} + \dfrac{1 + c_{b2}}{\sigma} \\ c_{b2} = 0.622, \quad \sigma = 2/3 \end{array}$$

where d_w is the distance to the closest wall. See [7] for the full set of definitions.

MS-5: Turbulent flows – RANS + **negative** SA model

$(\cdot)$	$(\cdot)_0$	$(\cdot)_x$	$(\cdot)_y$	$(\cdot)_{xy}$	$a_{(\cdot)x}$	$a_{(\cdot)y}$	$a_{(\cdot)xy}$
ρ	1.0	0.1	−0.2	0.1	1.0	1.0	1.0
u	2.0	0.3	0.3	0.3	3.0	1.0	1.0
v	2.0	0.3	0.3	0.3	1.0	1.0	1.0
p	10.0	1.0	1.0	0.5	2.0	1.0	1.0
$\tilde{\nu}$	−6.0	−0.3	−0.2	0.2	2.0	1.0	3.0

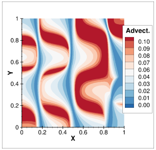


Same solution fields as MS-3 ; $-6.6 \lessapprox \tilde{\nu} \lessapprox -5.5$; $\mu = 1 \times 10^{-1}$

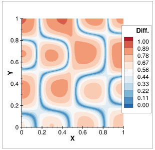
SA Forcing function sensitivity analysis $|\mathrm{S_{term}}|/\mathrm{S_{SA}^{sum}}$:

$$\mathrm{S_{SA}^{sum}} = \sum_{term} |\mathrm{S_{term}}| = \Big|\underbrace{\partial_j(\rho u_j \tilde{\nu})}_{\text{Advection}}\Big| + \Big|\underbrace{-\,\partial_j\left(\frac{1}{\sigma}(\mu + \rho\tilde{\nu}\frac{c_{n1}+\chi^3}{c_{n1}-\chi^3})\partial_j\tilde{\nu}\right)}_{\text{Diffusion}}\Big| + \Big|\underbrace{\rho\, c_{b1}(1 - c_{t3})\, s\, \tilde{\nu}}_{\text{Production}}\Big| +$$

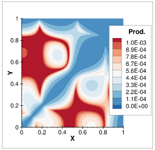
$$\Big|\underbrace{c_{w1}\frac{\tilde{\nu}^2}{d_w^2}}_{\text{Destruction}}\Big| + \Big|\underbrace{\frac{c_{b2}}{\sigma}\rho\,\partial_j\tilde{\nu}\,\partial_j\tilde{\nu}}_{\text{Distribution}}\Big| + \Big|\underbrace{\frac{1}{\sigma}(\nu + \tilde{\nu}\frac{c_{n1}+\chi^3}{c_{n1}-\chi^3})\,\partial_j(\rho\,\partial_j\tilde{\nu})}_{\text{Conservation}}\Big|.$$



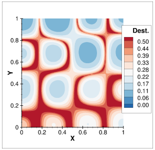
Advection



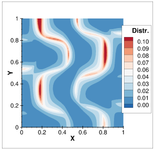
Diffusion



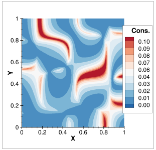
Production



Destruction



Distribution

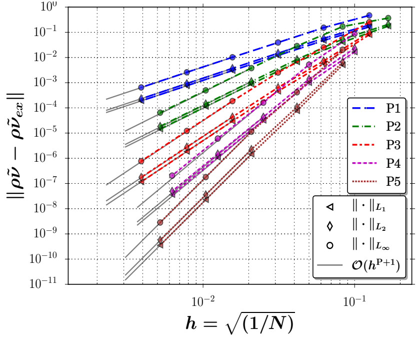


Conservation

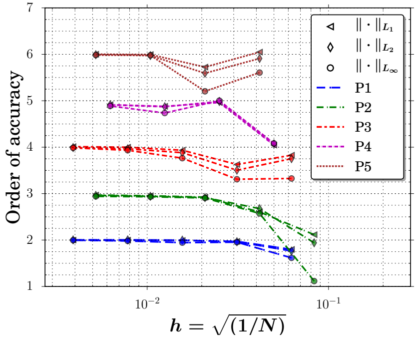
MS-5: Turbulent flows – RANS + **negative** SA model

L_1, L_2 and L_∞

$\mu_t = 0 \Rightarrow$ the same results as MS-3 for ρ, ρu, ρv and ρE



$\rho\tilde{\nu}$ Errors

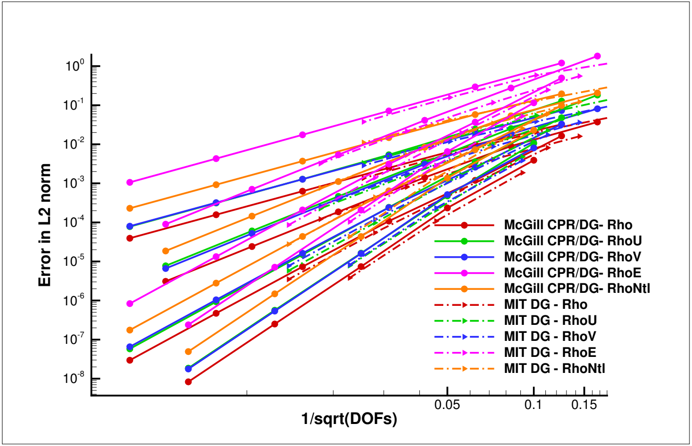


$\rho\tilde{\nu}$ Orders

MS-5: Turbulent flows – RANS + **negative** SA model

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-tris

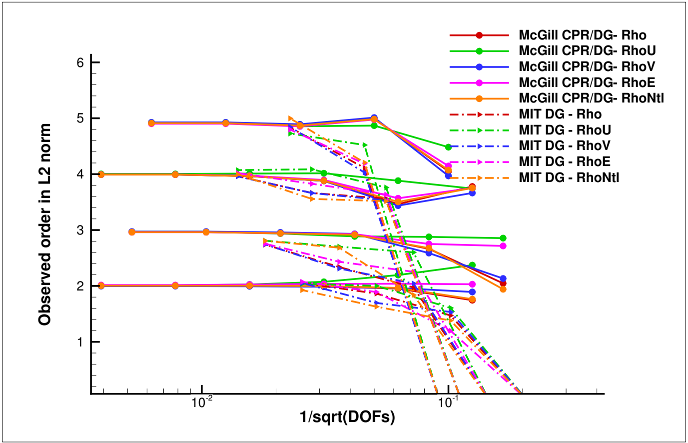
P1-P4



MS-5: Turbulent flows – RANS + **negative** SA model

Comparative results via [9]: McGill-CPR/DG-quads and MIT-SANS/DG-tris

P1-P4



Negative Spalart-Allmaras turbulence model definitions ICCFD7-1902

$$\partial_t(\rho\tilde{\nu}) + \partial_j(\rho u_j \tilde{\nu}) - \frac{1}{\sigma}\partial_j[(\mu + \rho\tilde{\nu} f_n)\partial_j \tilde{\nu}] = \frac{c_{b2}}{\sigma}\rho\, \partial_j\tilde{\nu}\, \partial_j\tilde{\nu} + \rho\mathcal{P} - \rho\mathcal{D} - \frac{1}{\sigma}(\nu + \tilde{\nu} f_n)\partial_j(\rho\partial_j\tilde{\nu})$$

$$\mu_t = \rho\nu_t \begin{cases} \rho\tilde{\nu} f_{v1} & \tilde{\nu} \geq 0, \\ 0 & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \tilde{\nu}/\nu, \quad c_{v1} = 7.1$$

$$\mathcal{P} = \begin{cases} c_{b1}\tilde{s}\tilde{\nu} & \tilde{\nu} \geq 0, \\ c_{b1} s\tilde{\nu} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad c_{b1} = 0.1355, \quad f_n = \begin{cases} 1 & \tilde{\nu} \geq 0, \\ \dfrac{c_{n1} + \chi^3}{c_{n1} - \chi^3} & \tilde{\nu} < 0, \end{cases}$$

$$\tilde{s} = \begin{cases} s + \bar{s} & \bar{s} \geq -c_{v2}s, \\ s + \dfrac{s(c_{v2}^2 s + c_{v3}\bar{s})}{(c_{v3} - 2c_{v2})s - \bar{s}} & \bar{s} < -c_{v2}s, \end{cases} \quad \text{where} \quad \begin{array}{l} \bar{s} = \dfrac{\tilde{\nu} f_{\nu 2}}{\kappa^2 d_w^2}, \; f_{v2} = 1 - \dfrac{\chi}{1 + \chi f_{v1}} \\ c_{v2} = 0.7, \; c_{v3} = 0.9, \; \kappa = 0.41 \end{array}$$

$$\mathcal{D} = \begin{cases} c_{w1} f_w \dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} \geq 0, \\ -c_{w1} \dfrac{\tilde{\nu}^2}{d_w^2} & \tilde{\nu} < 0, \end{cases} \quad \text{where} \quad \begin{array}{l} c_{w1} = \dfrac{c_{b1}}{\kappa^2} + \dfrac{1 + c_{b2}}{\sigma} \\ c_{b2} = 0.622, \quad \sigma = 2/3 \end{array}$$

where d_w is the distance to the closest wall. See [7] for the full set of definitions.

Unverified term, activated in wall-bounded flows (verified in Part II).

Part II. extra slides

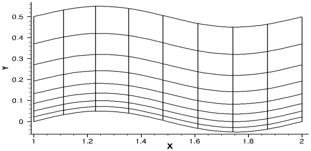
MS-2: Laminar flow on curved no-slip wall

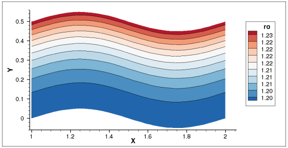
$$\rho^{\mathrm{MS}} \equiv \rho_0 + \frac{\mathcal{Y}^2}{\rho_c^2}, \qquad u^{\mathrm{MS}} = \mathrm{erf}(\eta),$$

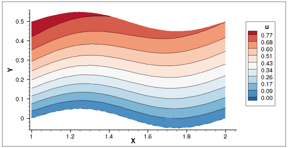
$$v^{\mathrm{MS}} = \frac{\partial y}{\partial x}\, u^{\mathrm{MS}}, \qquad p^{\mathrm{MS}} = p_0 + \mathcal{Y}^2,$$

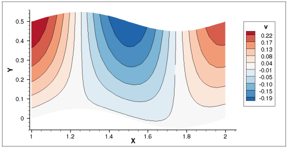
where $\rho_0 = 1.2$, $\rho_c = 3.0$, $\eta = \sigma \dfrac{\mathcal{Y}}{\sqrt{\frac{4}{3}(\mathcal{X}^2 + \frac{1}{2})}}$,

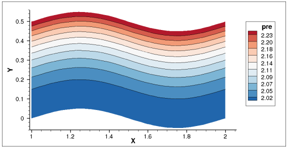
$\sigma = 2.0$, $p_0 = 2.0$, $\mu = 10^{-5}$.

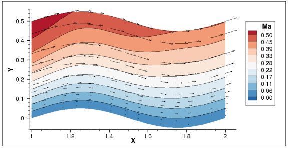






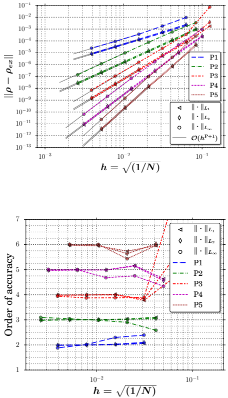




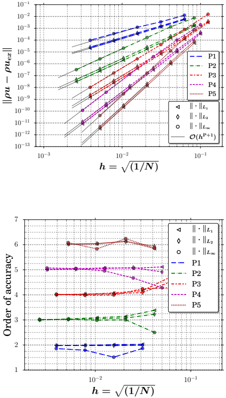


MS-2: Laminar flow on curved no-slip wall

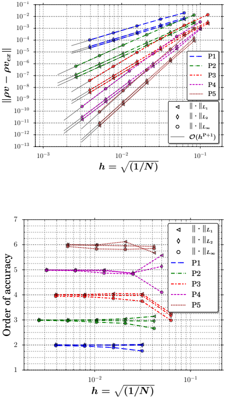
L_1, L_2 and L_∞



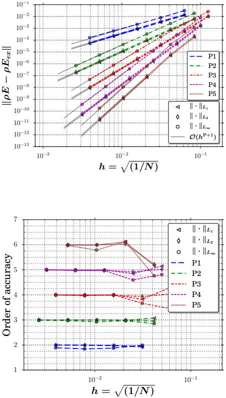
ρ



ρu



ρv

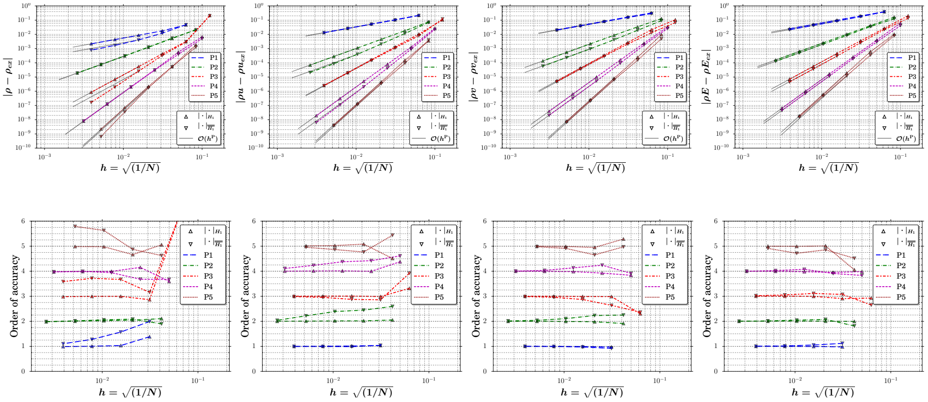


ρE

MS-2: Laminar flow on curved no-slip wall

H_1: uncorrected derivatives $(\partial_q Q_k)$

$\overline{\overline{H_1}}$: fully corrected BR2 derivatives $(\overline{\overline{\partial_q Q_k}})$

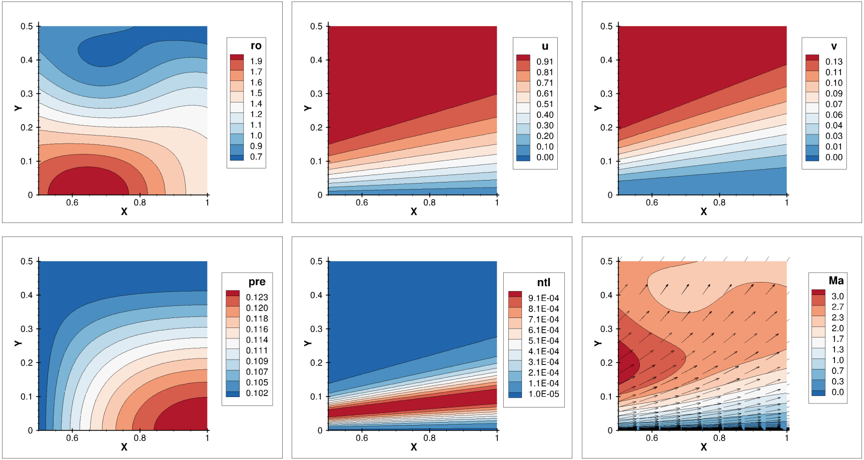


ρ ρu ρv ρE

MS-3: Transonic turbulent boundary layer flow

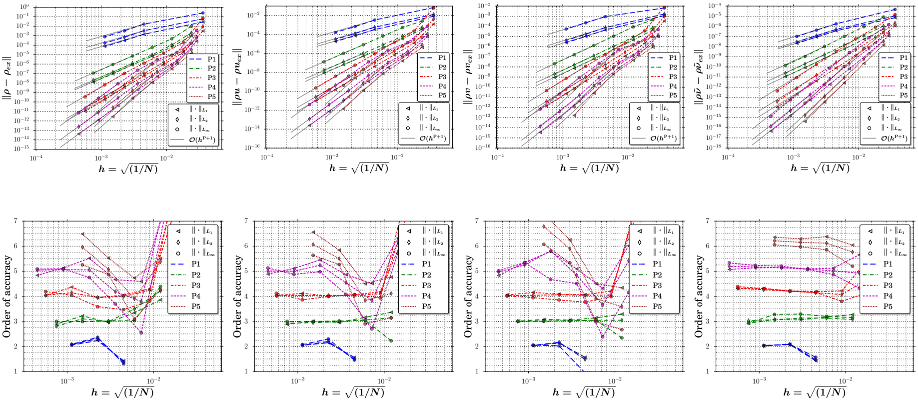
Adopted from Eça et al. (2007) [11] and adapted to compressible high-order solvers

See [8] for full details



MS-3: Transonic turbulent boundary layer flow

L_1, L_2 and L_∞



ρ $\qquad$ ρu $\qquad$ ρv $\qquad$ $\rho\tilde{\nu}$

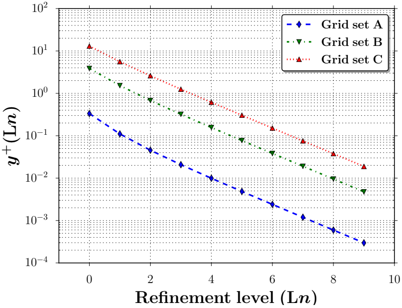
MS-3: Transonic turbulent boundary layer flow

A few remarks:

- Missing some elements from realistic RANS flows [10].
- Slow appearance of the asymptotic range

MS-4: Realistic turbulent boundary layer flow
Grid effect on solutions

Same number of elements
clustering at the wall : Grid A >>Grid B >> Grid C

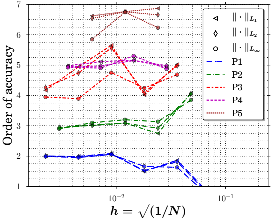


y^+

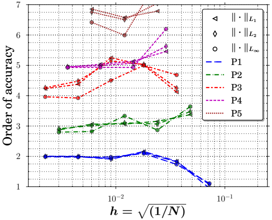
Figure: Comparison of exact y^+ based on the 1^{st} element height at the wall at $x = 0.525$ for grid sets A, B and C of MS-4

MS-4: Realistic turbulent boundary layer flow

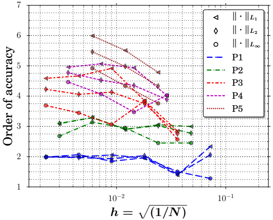
Grid effect on ρu errors



Grid set A



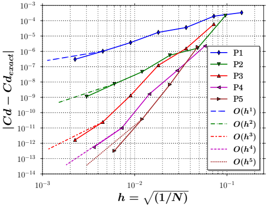
Grid set B



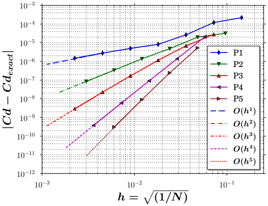
Grid set C

MS-4: Realistic turbulent boundary layer flow

Grid effect on output errors

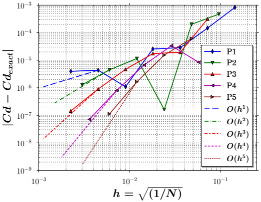


Grid set A



Grid set B

The orders of the output (C_d) seem to have a larger sensitivity to element distribution compared to the orders of ρu error norms. (Grid sets A vs. B)



Grid set C

References I

[1] W. L. Oberkampf, T. G. Trucano, Verification and validation in computational fluid dynamics, Progress in Aerospace Sciences 38 (3) (2002) 209–272.
doi:10.1016/S0376-0421(02)00005-2.
URL http://linkinghub.elsevier.com/retrieve/pii/S0376042102000052

[2] F. Navah, S. Nadarajah, Motivations and methods of verification for high-order RANS solvers and solutions, in: 23^{rd} AIAA Computational Fluid Dynamics Conference, AIAA, Denver, Colorado, 2017, pp. 1–37.
doi:10.2514/6.2017-3290.

References II

[3] Turbulence Modeling Resource: VERIF/2DZP: 2D Zero Pressure Gradient Flat Plate Verification Case - Intro Page, https://turbmodels.larc.nasa.gov/flatplate.html, [Online; accessed 1-January-2018] (2018).

[4] M. C. Galbraith, C. Ollivier-Gooch, 4^{th} High-order CFD Workshop, Joukowski Summary, slide 34/51, https://how4.cenaero.be/system/files/filedepot/12/BI2_BL1_BR1_Summary_Galbraith.pdf, [Online; accessed 4-January-2018] (2016).

References III

[5] S. R. Allmaras, F. T. Johnson, P. R. Spalart, Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model, in: 7th International Conference on Computational Fluid Dynamics, 2012.

[6] Z. J. Wang, H. Gao, T. Haga, A unifying discontinuous formulation for hybrid meshes, in: Adaptive High-Order Methods in Computational Fluid Dynamics, World Scientific Pub., 2011, Ch. 15, pp. 423–454.
URL http://www.worldscientific.com/worldscibooks/10.1142/7792

References IV

[7] F. Navah, S. Nadarajah, A comprehensive high-order solver verification methodology for free fluid flows, arXiv 1712.09478.
URL http://arxiv.org/abs/1712.09478

[8] F. Navah, S. Nadarajah, On the verification of CFD solvers of all orders of accuracy on curved wall-bounded domains and for realistic RANS flows, arXiv 1801.00021.
URL https://arxiv.org/abs/1801.00021

[9] F. Navah, Tools for the verification of high-order CFD solvers via the method of manufactured solutions (MMS), https://github.com/fanav/Verification_MMS_Tools, [Online; accessed 26-December-2017] (2017).

References V

[10] T. Oliver, K. Estacio-Hiroms, N. Malaya, G. Carey, Manufactured Solutions for the Favre-Averaged Navier-Stokes Equations with Eddy-Viscosity Turbulence Models, in: 50^{th} AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, American Institute of Aeronautics and Astronautics, Nashville, Tennessee, 2012, pp. 1–19.
doi:10.2514/6.2012-80.
URL http://arc.aiaa.org/doi/10.2514/6.2012-80

[11] L. Eça, M. Hoekstra, A. Hay, D. Pelletier, Verification of RANS solvers with manufactured solutions, Engineering with Computers 23 (2007) 253–270.
doi:10.1007/s00366-007-0067-9.