Design of a non-linear *hp*-multigrid scheme with static near-wall *p*-adaptation for RANS simulations in a high-order flux reconstruction framework

Saumitra Joshi[a,b,*], Aurelio Hurtado-de-Mendoza[a,b], Jiaqing Kou[a,b], Kunal Puri[a], Charles Hirsch[a], Esteban Ferrer[b]

[a] *NUMECA International S.A., Chaussee de la Hulpe 187, Brussels, B-1170, Belgium*

[b] *ETSIAE, School of Aeronautics, Universidad Politécnica de Madrid, Plaza Cardenal Cisneros 3, Madrid E-28040 Spain*

Abstract

High-order (HO) methods are of academic and industrial interest owing to greater accuracy per degree-of-freedom, favorable parallel scalability and quasi-mesh-independence. Their application to turbulence modeling using Reynolds-averaged Navier-Stokes (RANS) equations and hybrid RANS-LES (Large-Eddy Simulation) techniques is of particular interest to industry, given the fact that pure LES and Direct Numerical Simulation (DNS) still remain infeasible in an industrial context. Convergence acceleration is a major area of research in this context for steady-state problems as well as unsteady problems modeled using pseudo-time-stepping.

This paper analyzes the performance of a combination of *h*-multigrid and *p*-multigrid as applied to steady-state RANS-based turbulent flows. The one-equation Spalart-Allmaras model with negative-correction is used to account for turbulence and is verified through the use of realistic near-wall manufactured solutions. Static *p*-adaptation is used to attain appropriate near-wall resolution and to reduce the computational cost by limiting the degrees-of-freedom.

Through numerical experiments on turbulent flow over a flat-plate at Reynolds number 5 million, we show that the combination of *hp*-multigrid and *p*-adaptation significantly enhances convergence when compared to simple *p*-multigrid. *p*-adaptation achieves the same accuracy as uniform polynomial-orders at a much lower number of degrees-of-freedom. Using even a single additional *h*-level reduces the number of iterations by $\sim 60\%$.

Keywords: *hp*-multigrid, *p*-adaptation, high-order, flux-reconstruction, RANS, Spalart-Allmaras

Contents

*Corresponding author.
Email address: saumitravinay.joshi@numeca.be (Saumitra Joshi)

1. Introduction

Over the last five decades computational fluid dynamics (CFD) has come to be recognized as an indispensable tool in almost all branches of industrial design. Most industrially relevant flows are in the turbulent regime[2], with inertial effects outweighing viscous effects. With a grid-resolution that resolves flow structures at the dissipative length-scales, the Navier-Stokes equations are capable of describing these flows mathematically through the Direct Numerical Simulation (DNS) approach. However, the scaling of computational cost at high Reynolds numbers (Number of cells $N_c \propto \text{Re}^{3.5}$ for wall-bounded flows[1],[3], where Re is based on an integral length scale of the flow) makes DNS infeasible for the forseeable future for most non-academic scenarios. Consequently, focusing on the time-averaged behavior leads to the Reynolds-averaged Navier-Stokes (RANS) approach which is the industry-norm for turbulence modeling today[2],[3]. As RANS involves a temporal averaging of flow-quantities, it inherently smears out details in flow that would have been described by fluctuations at various length-scales and even results in inaccurate flow-behavior, e.g. for massively separated flows[6]. Large-eddy Simulation (LES) offers a compromise between DNS and RANS. It defines a filter length-scale; eddies of size greater than this length-scale are resolved, while those with lesser size are

modeled. Although grid requirements are significantly reduced compared to DNS, correct prediction of near-wall behavior using LES is still computationally intensive ($N_c \propto \mathrm{Re}^{\frac{13}{7}}$ [4], [5]). Hybrid RANS-LES methods look to further reduce computational cost through coarser grids and increased modeling in near-wall regions (e.g. $N_c \propto \mathrm{Re}_L$ for wall-modeled LES[4]. Here, Re_L is the Reynolds number based on the length of the computational domain). Nevertheless, there are several challenges that need to be overcome before LES and hybrid RANS-LES methods achieve industrial robustness, particularly their sensitivity to mesh distribution and parameter settings[5], [6], [8].

An orthogonal and complementary approach to reducing computational costs is to increase the accuracy delivered per degree-of-freedom. The current industrial workhorses are based on second-order accurate finite volume (FV) methods[2]. In comparison, "high-order" (HO) methods offer much higher accuracy per degree-of-freedom. Among the family of HO methods the discontinuous Galerkin (DG) method is by far the most dominant[11]. First introduced in the 70s[10], the DG method shows spectral polynomial convergence (typically scaling as h^{p+1} where h is the mesh resolution and p is the order of the polynomial in each cell), has a compact stencil, and can handle unstructured grids independent of cell-quality[12]. More recently, Huynh[14] proposed the flux reconstruction (FR) method - it unifies several existing HO-methods under a common framework and allows the discovery of new ones[13]. Moreover, it simplifies the implementation by not requiring surface or volume integrals since it uses the differential form of the governing equations[15]. A combination of these HO-methods with RANS and eventually hybrid RANS-LES techniques for turbulence modeling shows promise of making flows of higer Reynolds numbers viable. Moreover, by a better resolution in regions of interest, we can ensure an accurate solution while still keeping the cost in check. For turbulence modeling, this region is the boundary layer. Analogous to h-refinement, p-adaptation in HO-methods by increasing p in regions of interest has been employed in combination with p-multigrid on compressible laminar flows[34], [41]. It has recently been used for implicit under-resolved turbulence simulations[35].

Both steady-state and transient simulations are of importance in CFD. Transient flows can be treated as a sequences of steady-state simulations with timesteps comparable to the timescales of interest. Thus, developing efficient steady-state solvers is a widely researched topic. The defining characteristic of a good steady-state solver is a small time-to-convergence, and multigrid methods are widely used to minimize it[16], [17]. They rely on the fact that errors represented on coarser grids have a higher frequency than on the original fine grid, and are hence damped faster. Since time-accuracy is irrelevant for steady-state simulations, cell-local time-stepping ensures the maximum possible timestep in each cell respecting local stability limits. Traditional multigrid methods are based on successively coarser meshes[18] formed by agglomerating cells from finer meshes. We henceforth refer to them as h-multigrid. In HO-methods, alternatively to mesh-coarsening, we can represent the high-order errors on lower orders, effectively coarsening the polynomial-order p. The transfer operations between different multigrid levels are also much simpler, as the element-structure does not change and all operations are element-local. The resulting p-multigrid is a natural choice for accelerating HO-methods and has been extensively used for the last decade for elliptic, Euler and

compressible Navier-Stokes equations[23], [24], [25], [26], including RANS. Pure h-multigrid has also been used in HO-methods for Navier-Stokes equations[20], [21], [19]. For turbulent flows, there are several studies that use h-multigrid and p-multigrid separately[28], [29]. To extract the most benefit out of multigrid in the context of HO-methods, a combination of h- and p-multigrid is a lucrative option. Indeed, several studies have explored the so-called hp-multigrid for compressible laminar flows[31], [32] and have successfully demonstrated h- and p-independence of convergence. Recent work by Fehn et al[27] conducted a detailed analysis of using hp-multigrid for the Poisson equation in the context of pressure-based Navier-Stokes equations. However, to the best of the authors' knowledge there no published study of the behavior of combined hp-multigrid with RANS-modeled density-based turbulent flows, that also take into account the effects of p-adaptation.

This paper evaluates the use of hp-multigrid with p-adaptation for RANS-based turbulence modeling in the FR-framework. Sections 2 and 3 present the governing equations and the flux reconstruction method respectively. Section 4 introduces the hp-multigrid algorithm with details on the restriction and prolongation operators and calculation of cell-local time-step. It also describes the strategies for handling p-adaptation. In section 5, we ensure correctness of the implementation through the method of manufactured solutions and demonstrate spectral convergence of the FR-method. This is followed by demonstration of hp-multigrid and p-adaptation in section 6 on the standard test-case of turbulent flow over a flat-plate. Finally, section 7 summarizes the key findings of this paper, and draws an outlook for future work along these lines.

2. Governing equations

The flow is governed by the compressible unsteady RANS equations in conservative form with a Newtonian working fluid. Temperature dependence of viscosity is modeled using Sutherland's law[37]. Eddy viscosity is obtained using the modified Spalart-Allmaras (SA) turbulence model[39] to ensure numerical stability in the presence of negative values of the turbulence variable ϑ, especially with coarse spatial discretization of the boundary layer edge. The governing equations are cast into the following compact form:

$$\partial_t \boldsymbol{U} + \partial_j (\boldsymbol{F}_{\text{ivc}} + \boldsymbol{F}_{\text{vsc}}) = \boldsymbol{S} \tag{1}$$

where $\boldsymbol{U} \in \mathbb{R}^{1\times\mathcal{N}}$ is the solution-vector, $\boldsymbol{F}_{\text{ivc}} \in \mathbb{R}^{\mathcal{D}\times\mathcal{N}}$ and $\boldsymbol{F}_{\text{vsc}} \in \mathbb{R}^{\mathcal{D}\times\mathcal{N}}$ are the inviscid- and viscous-flux-vectors respectively, and $\boldsymbol{S} \in \mathbb{R}^{1\times\mathcal{N}}$ is the source-term-vector. $\mathcal{N}$ is the number of solution-variables and $\mathcal{D}$ is the spatial dimension. These are defined as:

$$\boldsymbol{U} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho E \\ \rho \vartheta \end{bmatrix} \quad , \quad \boldsymbol{F}_{\text{ivc}} = \begin{bmatrix} \rho v_1 & \rho v_2 & \rho v_3 \\ p + \rho v_1 v_1 & \rho v_1 v_2 & \rho v_1 v_3 \\ \rho v_2 v_1 & p + \rho v_2 v_2 & \rho v_2 v_3 \\ \rho v_3 v_1 & \rho v_3 v_2 & p + \rho v_3 v_3 \\ \rho v_1 H & \rho v_2 H & \rho v_3 H \\ \rho v_1 \vartheta & \rho v_2 \vartheta & \rho v_3 \vartheta \end{bmatrix} \quad , \tag{2}$$

$$\boldsymbol{F}_{\text{vsc}} = \begin{bmatrix} 0 & 0 & 0 \\ \tau_{11} & \tau_{12} & \tau_{13} \\ \tau_{21} & \tau_{22} & \tau_{23} \\ \tau_{31} & \tau_{32} & \tau_{33} \\ v_i \tau_{i1} + \omega_1 & v_i \tau_{i2} + \omega_2 & v_i \tau_{i3} + \omega_3 \\ \frac{\eta}{\sigma} \partial_1 \vartheta & \frac{\eta}{\sigma} \partial_2 \vartheta & \frac{\eta}{\sigma} \partial_3 \vartheta \end{bmatrix} \quad , \quad \boldsymbol{S} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \mathcal{G} - \mathcal{Y} + \mathcal{K} + \mathcal{T} \end{bmatrix} .$$

Einstein summation convention is used for repeated indices i. The symbols used in the equations above are as follows. ρ is the density, $\boldsymbol{v} = \boldsymbol{e}_i v_i$ is the velocity vector with $\boldsymbol{e}_i$ being the i-th orthonormal basis vector of Euclidean space, E is the total energy per unit mass, i.e. $E = e + \frac{1}{2} v_i v_i$ where e is the internal energy. For a calorically perfect gas, $e = \frac{RT}{\gamma - 1}$ where R is the gas constant and T is the temperature determined as $T = \gamma \mathrm{M}^2 p / \rho$. The total enthalpy H is defined as $H = E + \frac{p}{\rho}$ where p is the pressure, related to energy through the ideal gas law:

$$p = \rho \left(\gamma - 1 \right) \left(E - \frac{1}{2} v_i v_i \right)$$

where γ is the specific heat ratio. τ_{ij} are the components of the viscous stress tensor $\underline{\underline{\tau}}$. For compressible Newtonian fluids:

$$\tau_{ij} = 2(\mu + \mu_t) S_{ij} \quad , \quad S_{ij} = \frac{1}{2} \left(\partial_i v_j + \partial_j v_i \right) - \frac{1}{3} \partial_k v_k \delta_{ij}$$

where μ_t is the eddy viscosity and μ is the dynamic viscosity which is determined as a function of temperature through Sutherland's law. $\omega_j = (\lambda + \lambda_t) \partial_j T$ is the j-th component of the heat flux vector where $\lambda = \frac{\gamma R}{\gamma - 1} \frac{\mu}{\mathrm{Pr}}$ is the molecular conductivity and $\lambda_t = \frac{\gamma R}{\gamma - 1} \frac{\mu_t}{\mathrm{Pr}_t}$ is the eddy conductivity. Unless specified otherwise, the laminar Prandtl number $\mathrm{Pr} = 0.72$ and the turbulent Prandtl number $\mathrm{Pr}_t = 0.9$.

For the SA model, the eddy viscosity μ_t is calculated from the turbulence variable ϑ and the kinematic viscosity ν as

$$\mu_t = \begin{cases} \rho\vartheta f_{v_1} & \text{if } \vartheta \geqslant 0 \\ \\ 0 & \text{otherwise} \end{cases} \quad , \quad f_{v_1} = \frac{\chi^3}{\chi^3 + c_{v_1}^3} \quad , \quad \chi = \vartheta/\nu \quad , \quad c_{v_1} = 7.1. \tag{3}$$

To ensure positivity and C^1-continuity of the diffusion coefficient it is defined as

$$\eta = \begin{cases} \mu\left(1+\chi\right) & \text{if } \vartheta \geqslant 0 \\ \\ \mu\left(1+\chi+\frac{1}{2}\chi^2\right) & \text{otherwise.} \end{cases} \tag{4}$$

The production term $\mathcal{G}$ is defined as

$$\mathcal{G} = \begin{cases} c_{b_1}\tilde{s}\rho\vartheta & \text{if } \vartheta \geqslant 0 \\ \\ c_{b_1} s\rho\vartheta g_n & \text{otherwise} \end{cases} \quad , \quad c_{b_1} = 0.1355 \quad , \quad g_n = 1 - \frac{1000\chi^2}{1+\chi^2}. \tag{5}$$

$s = |\epsilon_{ijk}\partial_j v_k|$ is the vorticity magnitude with ϵ_{ijk} being the Levi-Civita symbol for permutation. $\tilde{s}$ is the modified vorticity defined as

$$\tilde{s} = \begin{cases} s + \bar{s} & \text{if } \bar{s} \geqslant -c_{v_2} s \\ \\ s + \dfrac{s\left(c_{v_2}^2 s + c_{v_3}\bar{s}\right)}{\left(c_{v_3} - 2c_{v_2}\right)s - \bar{s}} & \text{otherwise} \end{cases} \quad , \quad \bar{s} = \frac{\vartheta f_{v_2}}{\kappa^2 d_w^2} \tag{6}$$

$$f_{v_2} = 1 - \frac{\chi}{1+\chi f_{v_1}} \quad , \quad c_{v_2} = 0.7 \quad , \quad c_{v_3} = 0.9 \quad , \quad \kappa = 0.41$$

with d_w being the distance to the nearest wall. Note the difference between $\tilde{s}$ and $\bar{s}$. The destruction term $\mathcal{Y}$ is defined as

$$\mathcal{Y} = \begin{cases} c_{w_1} f_w \dfrac{\rho\vartheta^2}{d_w^2} & \text{if } \vartheta \geqslant 0 \\ \\ -c_{w_1}\dfrac{\rho\vartheta^2}{d_w^2} & \text{otherwise} \end{cases} \quad , \quad c_{w_1} = \frac{c_{b_1}}{\kappa^2} + \frac{1+c_{b_2}}{\sigma} \tag{7}$$

$$f_w = g\left(\frac{1+c_{w_3}^6}{g^6 + c_{w_3}^6}\right)^{\frac{1}{6}} \quad , \quad g = r + c_{w_2}\left(r^6 - r\right) \quad , \quad r = \min\left(\frac{\vartheta}{\tilde{s}\kappa^2 d_w^2}\ ,\ r_{\max}\right)$$

$$r_{\max} = 2 \quad , \quad c_{b_2} = 0.622 \quad , \quad c_{w_2} = 0.3 \quad , \quad c_{w_3} = 2 \quad , \quad \sigma = \frac{2}{3}.$$

The trip term $\mathcal{T}$ is set to 0 throughout this work as we are concerned with flows in the fully turbulent regime. The term $\mathcal{K}$ is defined as

$$\mathcal{K} = \frac{c_{b_2}}{\sigma}\rho\partial_j\vartheta\partial_j\vartheta. \tag{8}$$

3. Flux reconstruction method

Consider the following hyperbolic system of equations living in physical space $\Omega \in \mathbb{R}^{\mathcal{D}}$ with a closed boundary Γ.

$$\begin{aligned} \partial_t \boldsymbol{U} + \nabla \cdot (\boldsymbol{F}_{\text{ivc}} + \boldsymbol{F}_{\text{vsc}}) &= \boldsymbol{S} \\ \boldsymbol{Q} - \nabla \boldsymbol{U} &= 0. \end{aligned} \tag{9}$$

The inviscid flux-vector $\boldsymbol{F}_{\text{ivc}}$, viscous flux-vector $\boldsymbol{F}_{\text{vsc}}$ and source-term vector $\boldsymbol{S}$ can be functions of the solution $\boldsymbol{U}$ and its gradient $\boldsymbol{Q} = \nabla \boldsymbol{U}$.

We discretize space Ω into N_c distinct cells which are $\mathcal{D}$-dimensional orthotopes (quadrangles for $\mathcal{D} = 2$, hexahedra for $\mathcal{D} = 3$). In each cell, along each dimension, the solution is represented by a polynomial of order p at $(N_p)^{\mathcal{D}}$ number of solution-points, where $N_p = p + 1$. $\boldsymbol{x}^{\delta}_{i,\boldsymbol{\mathcal{J}}} \in \mathbb{R}^{\mathcal{D}}$ refers to the solution-point in cell C_i at some index $\boldsymbol{\mathcal{J}}$. Several choices exist for the location of the solution-points; we use "Gauss-points", which are the roots of the Jacobi polynomial of order p. The superscript δ indicates the discrete nature of these points. There exists for each cell a conformal spatial mapping $\mathcal{M} : \boldsymbol{x} \to \boldsymbol{\xi}$ that transforms physical coordinates to standard coordinates that are more convenient for performing cell-local operations.

We also define $(N_p)^{\mathcal{D}-1}$ points on each face of a cell. These points are shared by neighbouring cells on their common face. The fluxes across the boundaries of a cell are defined at these points and hence they are refered to as "flux-points". The same cell-local map $\mathcal{M}$ can be applied to them. $\boldsymbol{x}^{\delta\text{F}}_{i,\boldsymbol{\mathcal{J}}}$ refers to a flux-point of cell C_i at some index $\boldsymbol{\mathcal{J}}$.

Step FR1 Start with the discrete solution at each point, represented by $\boldsymbol{U}^{\delta}_{i,\boldsymbol{\mathcal{J}}}$. The solution at any point in a given cell is calculated by a piecewise-discontinuous polynomial interpolation, which is isotropic in its order. Henceforth, $\mathcal{I}^{p}_{g^{\delta}}$ represents a Lagrange polynomial of order p that interpolates the discrete data g^{δ} of appropriate size and dimension.

Step FR2 Calculate interface solution values $\boldsymbol{U}^{\delta\text{I}}$: For each cell C_i, perform a cell-local interpolation of $\boldsymbol{U}^{\delta}$ to each flux-point of that cell as

$$\boldsymbol{U}^{\delta\text{F}}_{i,\boldsymbol{\mathcal{J}}} = \mathcal{I}^{p}_{\boldsymbol{U}^{\delta}}\left(\boldsymbol{x}^{\delta\text{F}}_{i,\boldsymbol{\mathcal{J}}}\right). \tag{10}$$

This gives us the interpolated solution $\boldsymbol{U}^{\delta\mathrm{F}}$ at flux-points. Since flux-points are shared by cells at their common face, this leads to two values at each flux-point - the value from the "left" cell (C_-) and that from the "right" cell (C_+), with the face-normal $\hat{\boldsymbol{n}}$ going from C_- to C_+. These are referred to as the "discontinuous" values, which for the solution are represented by $\boldsymbol{U}_-^{\delta\mathrm{F}}$ and $\boldsymbol{U}_+^{\delta\mathrm{F}}$. We need to resolve the left and right values into a common "interface" value - this is equivalent to solving a Riemann-problem. For the discontinuous solution at flux-points $\boldsymbol{U}^{\delta\mathrm{F}}$, the interface value $\boldsymbol{U}^{\delta\mathrm{I}}$ is obtained by:

$$\boldsymbol{U}^{\delta\mathrm{I}} = \mathcal{R}_{\boldsymbol{U}^{\delta\mathrm{F}}} = \mathcal{R}_{\boldsymbol{U}^{\delta\mathrm{F}}}\left(\boldsymbol{U}_-^{\delta\mathrm{F}}, \boldsymbol{U}_+^{\delta\mathrm{F}}\right) \tag{11}$$

where $\mathcal{R}_{\boldsymbol{U}^{\delta\mathrm{F}}}$ is the local discontinuous-Galerkin (LDG) Riemann-solver.

Step FR3 Construct continuous solution interpolating polynomial $\mathcal{I}^p_{\boldsymbol{U}^{\delta\mathrm{C}}}$: we need to ensure that the polynomial that interpolates the solution in the cell takes the values of $\boldsymbol{U}^{\delta\mathrm{I}}$ at the interfaces. This is done by adding an appropriate "correction-function" $\mathcal{C}^{p+1}_{\boldsymbol{U}^\delta}$ that yields $\boldsymbol{U}^{\delta\mathrm{I}} - \boldsymbol{U}^{\delta\mathrm{F}}$ at interfaces and zero at all solution-points. A widely adopted choice for $\mathcal{C}$ is a combination of the left- and right-Radau polynomials. Thus, the corrected solution interpolating polynomial is given as

$$\mathcal{I}^{p+1}_{\boldsymbol{U}^{\delta\mathrm{C}}} = \mathcal{I}^p_{\boldsymbol{U}^\delta} + \mathcal{C}^{p+1}_{\boldsymbol{U}^\delta}. \tag{12}$$

Step FR4 Calculate gradient $\boldsymbol{Q}^\delta$ of $\boldsymbol{U}^{\delta\mathrm{C}}$ at solution-points: gradients at solution-points are calculated simply as the gradient of the continuous interpolating solution polynomial at the respective points, i.e.

$$\boldsymbol{Q}^\delta = \nabla^\delta \mathcal{I}^{p+1}_{\boldsymbol{U}^{\delta\mathrm{C}}}\left(\boldsymbol{x}^\delta\right). \tag{13}$$

It involves evaluating the gradient of the interpolation and correction polynomials. ∇^δ represents the discrete gradient operator matrix. Note that $\boldsymbol{Q}^\delta \in \mathbb{R}^{\mathcal{D}\times\mathcal{N}}$.

Step FR5 Calculate interface flux values ${\boldsymbol{F}_{\mathrm{ivc}}}^{\delta\mathrm{I}}$ and ${\boldsymbol{F}_{\mathrm{vsc}}}^{\delta\mathrm{I}}$: we perform a cell-local interpolation of $\boldsymbol{Q}^\delta$ to each flux-point of that cell as

$$\boldsymbol{Q}^{\delta\mathrm{F}} = \mathcal{I}^p_{\boldsymbol{Q}^\delta}\left(\boldsymbol{x}^{\delta\mathrm{F}}\right). \tag{14}$$

Then use the Roe-type Riemann solver with Roe-diffusion set to 0.5 to resolve the inviscid interface fluxes, and a local discontinuous-Galerkin (LDG) Riemann solver to resolve the viscous interface fluxes.

$$
\begin{aligned}
\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{I}} &= \mathcal{R}_{\boldsymbol{F}_{\text{ivc}}}\left(\boldsymbol{U}_{-}^{\delta\text{F}}, \boldsymbol{U}_{+}^{\delta\text{F}}\right) \\
\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{I}} &= \mathcal{R}_{\boldsymbol{F}_{\text{vsc}}}\left(\boldsymbol{U}_{-}^{\delta\text{F}}, \boldsymbol{Q}_{-}^{\delta\text{F}}, \boldsymbol{U}_{+}^{\delta\text{F}}, \boldsymbol{Q}_{+}^{\delta\text{F}}\right).
\end{aligned}
\tag{15}
$$

Note that the interface fluxes are calculated from $\boldsymbol{U}^{\delta\text{F}}$ and $\boldsymbol{Q}^{\delta\text{F}}$ using the flux-functions $\boldsymbol{F}_{\text{ivc}}$ and $\boldsymbol{F}_{\text{ivc}}$. Unlike in *Step FR2*, they are *not* calculated from $\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{F}}$ and $\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{F}}$.

Step FR6 Construct continuous flux interpolating polynomials $\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{C}}}$ and $\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{C}}}$: First, we use $\boldsymbol{U}^{\delta}$ and $\boldsymbol{Q}^{\delta}$ to evaluate inviscid and viscous fluxes at solution-points. These are typically discontinuous at cell-interfaces.

$$
\begin{aligned}
\boldsymbol{F}_{\text{ivc}}{}^{\delta} &= \boldsymbol{F}_{\text{ivc}}\left(\boldsymbol{U}^{\delta}\right) \\
\boldsymbol{F}_{\text{vsc}}{}^{\delta} &= \boldsymbol{F}_{\text{vsc}}\left(\boldsymbol{U}^{\delta}, \boldsymbol{Q}^{\delta}\right).
\end{aligned}
\tag{16}
$$

These are then interpolated to flux-points by constructing Lagrange interpolation poynomials for the fluxes and evaluating them at flux-points.

$$
\begin{aligned}
\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{F}} &= \mathcal{I}^{p}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta}}\left(\boldsymbol{x}^{\delta\text{F}}\right) \\
\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{F}} &= \mathcal{I}^{p}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta}}\left(\boldsymbol{x}^{\delta\text{F}}\right).
\end{aligned}
\tag{17}
$$

To ensure that the polynomials that interpolate fluxes in the cell take values of $\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{I}}$ and $\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{I}}$ at interfaces, we add correction functions $\mathcal{C}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta}}$ and $\mathcal{C}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta}}$ similar to *Step FR3*. Thus, the corrected flux interpolating polynomials are gives as

$$
\begin{aligned}
\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{C}}} &= \mathcal{I}^{p}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta}} + \mathcal{C}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta}} \\
\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{C}}} &= \mathcal{I}^{p}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta}} + \mathcal{C}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta}}.
\end{aligned}
\tag{18}
$$

Step FR7 Calculate the flux-divergence and source-term contribution: calculating the flux-divergence involves evaluating the gradient of the interpolation and correction polynomials as

$$
\nabla^{\delta} \cdot \left(\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{C}}} + \mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{C}}}\right).
\tag{19}
$$

We evaluate the source-term contribution as a function of the discrete solution and solution-gradient as

$$
\boldsymbol{S}^{\delta} = \boldsymbol{S}\left(\boldsymbol{U}^{\delta}, \boldsymbol{Q}^{\delta}\right).
\tag{20}
$$

Having evaluated all spatial terms we can now march the following semi-discrete equation in time using any standard time-integrator.

$$\partial_t \boldsymbol{U} = \text{RHS} = -\nabla^{\delta} \cdot \left(\mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{ivc}}{}^{\delta \text{C}}} + \mathcal{I}^{p+1}_{\boldsymbol{F}_{\text{vsc}}{}^{\delta \text{C}}} \right) + \boldsymbol{S}^{\delta}. \tag{21}$$

Here, we use the Runge-Kutta-54 scheme with optimized coefficients developed recently by Vermeire et al[42]. This optimized RK scheme allows us to take larger timesteps without affecting stability. It comes at a cost of loss in time-accuracy, which is not of concern to when seeking steady-state solutions.

Boundary conditions

In *Step FR2* and *Step FR5* we also need to calculate the common values at the boundary faces. Here we specify the treatment of these steps at different types of boundaries. For every boundary face we presume that the normal $\hat{\boldsymbol{n}}$ point outside the domain.

(a) No-slip adiabatic wall (NSAW):

For common solution values at the boundary,

$$U_k{}^{\delta\Gamma} = \begin{cases} U_{k-}^{\delta \text{F}} & \text{if} \quad k \in \{1, 5\} \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

For calculating $\boldsymbol{F}_{\text{ivc}}{}^{\delta\Gamma}$, we set the right-side momentum-variables by preserving the component parallel to the boundary while reversing the perpendicular component. All other variables are set to their left counterparts. The Roe-diffusion is set to 0.

$$U_{k+}^{\delta \text{F}} = \begin{cases} (\rho v_{k-1})_-^{\delta \text{F}} - 2\hat{n}_{k-1} (\rho \boldsymbol{v})_-^{\delta \text{F}} \cdot \hat{\boldsymbol{n}} & \text{if} \quad k \in [2, 4] \\ U_{k-}^{\delta \text{F}} & \text{otherwise.} \end{cases} \tag{23}$$

We use the left-state values for calculating $\boldsymbol{F}_{\text{vsc}}{}^{\delta\Gamma}$ for all variables, except energy for which it is set to 0:

$$\boldsymbol{F}_{\text{vsc},k}{}^{\delta\Gamma} = \begin{cases} 0 & \text{if} \quad k \in [5] \\ \boldsymbol{F}_{\text{vsc},k} \left(\boldsymbol{U}_-^{\delta \text{F}}, \boldsymbol{Q}_-^{\delta \text{F}} \right) & \text{otherwise.} \end{cases} \tag{24}$$

(b) Characteristic (CHAR), Subsonic pressure outlet (POUT), Symmetry (SYMM):

The Roe-diffusion is set to 0. For calculating both $\boldsymbol{U}^{\delta\Gamma}$ and $\boldsymbol{F}_{\text{ivc}}{}^{\delta\Gamma}$, the right-side solution-state is calculated in the same manner as described by Mengaldo et. al.[43]. The viscous flux contribution $\boldsymbol{F}_{\text{vsc}}{}^{\delta\Gamma}$ is set to 0 for characteristic and symmetry.

4. *hp*-multigrid with *p*-adaptation

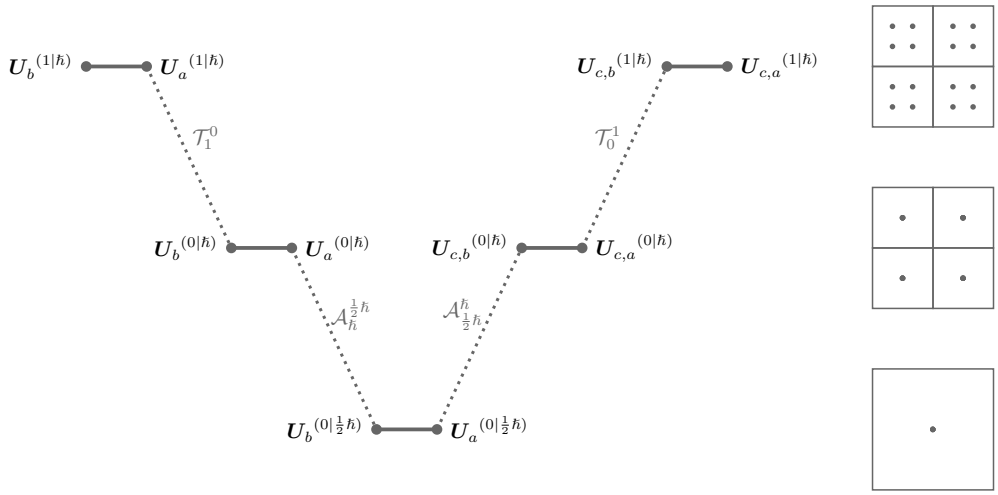


Figure 1: Schematic depiction of one V-cycle of the *hp*-multigrid algorithm. On the right is a representation of the degrees of freedom at every level.

The core concept in any multigrid algorithm is that errors represented on coarser discrete space have a higher frequency than on the original discrete space, and are hence damped faster. To illustrate its steps, we describe the steps on a single cycle of a simple three-level *hp*-multigrid Full Approximation Scheme (FAS) as depicted in the schematic in Figure 1. For simplicity, we drop the δ superscript and assume that the discrete nature of all terms involved is implied.

4.1. hp-multigrid Algorithm

Consider a solution-state $\boldsymbol{U}_b{}^{(1|\hbar)}$ at polynomial-order 1 and mesh-resolution $\hbar$. The corresponding semi-discrete form of the problem is

$$\partial_t \boldsymbol{U}^{(1|\hbar)} = \mathrm{RHS}^{(1|\hbar)}\left(\boldsymbol{U}^{(1|\hbar)}\right) - f^{(1|\hbar)} \tag{25}$$

where $f^{(1|\hbar)}$ is the multigrid source-term. If 1 is the polynomial-order p of the original problem and likewise $\hbar$ is the mesh-resolution h of the original problem, then $f^{(p|h)} = 0$. Let us denote this system as the multigrid "level" $l^{(1|\hbar)}$. The steps of one multigrid V-cycle are:

Step MG1 Perform $n_{\mathrm{pre}}{}^{(1|\hbar)}$ number of RK-iterations at $l^{(1|\hbar)}$ to obtain a smoothed solution $\boldsymbol{U}_a{}^{(1|\hbar)}$. The subscripts b and a denote a state before and after smoothing respectively. The defect at this level is calculated from the smoothed solution as

$$d^{(1|\hbar)} = f^{(1|\hbar)} - \mathrm{RHS}^{(1|\hbar)}\left(\boldsymbol{U}_a{}^{(1|\hbar)}\right). \tag{26}$$

Step MG2 Transfer the smoothed solution ${\boldsymbol{U}_a}^{(1|\hbar)}$ and the defect $d^{(1|\hbar)}$ to a coarser level $l^{(0|\hbar)}$ using an $\mathbb{L}^2$ projection. This is denoted by the operator $\mathcal{T}_1^0$.

$$\begin{aligned} {\boldsymbol{U}_b}^{(0|\hbar)} &= \mathcal{T}_1^0 {\boldsymbol{U}_a}^{(1|\hbar)} \\ f^{(0|\hbar)} &= \mathrm{RHS}^{(0|\hbar)}\left({\boldsymbol{U}_b}^{(0|\hbar)}\right) - \mathcal{T}_1^0 d^{(1|\hbar)}. \end{aligned} \tag{27}$$

The corresponding semi-discrete form is

$$\partial_t \boldsymbol{U}^{(0|\hbar)} = \mathrm{RHS}^{(0|\hbar)}\left(\boldsymbol{U}^{(0|\hbar)}\right) - f^{(0|\hbar)} \tag{28}$$

using which ${n_{\mathrm{pre}}}^{(0|\hbar)}$ smoothing steps are performed to obtain the smoothed solution ${\boldsymbol{U}_a}^{(0|\hbar)}$. Note that for order 0 the solution in a given cell is constant. The defect at this level is calculated from the smoothed solution as

$$d^{(0|\hbar)} = f^{(0|\hbar)} - \mathrm{RHS}^{(0|\hbar)}\left({\boldsymbol{U}_a}^{(0|\hbar)}\right). \tag{29}$$

Step MG3 Similar to *Step MG2*, transfer ${\boldsymbol{U}_a}^{(0|\hbar)}$ and $d^{(0|\hbar)}$ to a coarser level $l^{(0|\frac{1}{2}\hbar)}$, this time coarsening the mesh-resolution. For every set of agglomerated cells, the transfer operator $\mathcal{A}_{\hbar}^{\frac{1}{2}\hbar}$ assigns a simple averaging of their solution-values to the coarser unified cell. Thus,

$$\begin{aligned} {\boldsymbol{U}_b}^{(0|\frac{1}{2}\hbar)} &= \mathcal{A}_{\hbar}^{\frac{1}{2}\hbar} {\boldsymbol{U}_a}^{(0|\hbar)} \\ f^{(0|\frac{1}{2}\hbar)} &= \mathrm{RHS}^{(0|\frac{1}{2}\hbar)}\left({\boldsymbol{U}_b}^{(0|\frac{1}{2}\hbar)}\right) - \mathcal{A}_{\hbar}^{\frac{1}{2}\hbar} d^{(0|\hbar)}. \end{aligned} \tag{30}$$

The corresponding semi-discrete form is

$$\partial_t \boldsymbol{U}^{(0|\frac{1}{2}\hbar)} = \mathrm{RHS}^{(0|\frac{1}{2}\hbar)}\left(\boldsymbol{U}^{(0|\frac{1}{2}\hbar)}\right) - f^{(0|\frac{1}{2}\hbar)} \tag{31}$$

using which ${n_{\mathrm{pre}}}^{(0|\frac{1}{2}\hbar)}$ smoothing steps are performed to obtain ${\boldsymbol{U}_a}^{(0|\frac{1}{2}\hbar)}$.

Step MG4 Compute the correction for $l^{(0|\frac{1}{2}\hbar)}$ and transfer it to $l^{(0|\hbar)}$ as

$$\begin{aligned} \Delta \boldsymbol{U}^{(0|\frac{1}{2}\hbar)} &= {\boldsymbol{U}_a}^{(0|\frac{1}{2}\hbar)} - {\boldsymbol{U}_b}^{(0|\frac{1}{2}\hbar)} \\ {\boldsymbol{U}_{c,b}}^{(0|\hbar)} &= {\boldsymbol{U}_a}^{(0|\hbar)} + \alpha_{\frac{1}{2}\hbar}^{\hbar} \mathcal{A}_{\frac{1}{2}\hbar}^{\hbar} \Delta \boldsymbol{U}^{(0|\frac{1}{2}\hbar)}. \end{aligned} \tag{32}$$

For every coarse unified cell, the transfer operator $\mathcal{A}_{\frac{1}{2}\hbar}^{\hbar}$ simply sets its value to all the agglomerated cells that constitute it. $\alpha_{\frac{1}{2}\hbar}^{\hbar} \in (0, 1]$ is an under-relaxation factor used to stabilize the transfer. The subscript c, b refers to the corrected solution before smoothing. Perform ${n_{\mathrm{post}}}^{(0|\hbar)}$ post-smoothing iterations using the system in equation (28) to get ${\boldsymbol{U}_{c,a}}^{(0|\hbar)}$.

Step MG5 Similar to *Step MG4* compute the correction at $l^{(0|\hbar)}$ and transfer it to $l^{(1|\hbar)}$ using an $\mathbb{L}^2$ projection as

$$\begin{aligned} \Delta \boldsymbol{U}^{(0|\hbar)} &= \boldsymbol{U}_{c,a}{}^{(0|\hbar)} - \boldsymbol{U}_b{}^{(0|\hbar)} \\ \boldsymbol{U}_{c,b}{}^{(1|\hbar)} &= \boldsymbol{U}_a{}^{(1|\hbar)} + \mathcal{T}_0^1 \Delta \boldsymbol{U}^{(0|\hbar)}. \end{aligned} \tag{33}$$

Perform $n_{\text{post}}{}^{(1|\hbar)}$ post-smoothing iterations using the system in equation (25) to get $\boldsymbol{U}_{c,a}{}^{(1|\hbar)}$.

This completes one V-cycle of the *hp*-multigrid algorithm, with $\boldsymbol{U}_{c,a}{}^{(1|\hbar)}$ as the updated solution. Two issues need to be addressed at this point: the first is the calculation of the cell-local time-step. Since we are interested in the steady-state solution, time-accuracy is not important. We therefore can take as large a timestep as permissible by stability limits, which can vary from cell-to-cell. For a given cell C_i, with Δx_i being the shortest distance between two solution-points in that cell and CFL being the appropriately selected Courant-Friedrich-Lewy number, the local time-step is

$$\Delta t_i = \text{CFL} \cdot \frac{\Delta x_i}{\left(\|\boldsymbol{v}_i\| + \dfrac{\gamma p_i}{\rho_i} \right) + \left(\dfrac{\nu_i + \dfrac{\mu_{ti}}{\rho_i}}{\Delta x_i} \right)}. \tag{34}$$

4.2. Source-term Transfer

The second issue is the handling of the turbulent source-term. From equations (4) to (8) we see that the source-terms in the transport of the turbulence variable ϑ are highly non-linear. This can introduce aliasing errors when transferring solutions between different multigrid levels. Let us illustrate this with a short example. Consider simple advection with a non-linear source-term as follows

$$\partial_t u + \partial_{x_i} u = s \quad , \quad s = \left(\partial_{x_i} u\right)^2 \tag{35}$$

with the solution $u^{(5)}$ at order 5 as depicted in Figure 2-(i) and gradient in Figure 2-(ii). The solutions at coarser orders are obtained by the transfer of solutions from higher orders, and then their gradients are evaluated. Although $u^{(p)}$ at all orders are representative of $u^{(5)}$, the gradient $\partial_{x_i} u^{(3)}$ shows a large deviation. Thus, in Figure 2-(iii) we see that using the transferred solution (and subsequently its gradient) to evaluate the source-term as

$$s_{\text{std}}^{(p)} = s^{(p)} \left(\mathcal{T}_{p+1}^{p} u^{(p+1)} \right) \quad \forall \quad p < 5 \tag{36}$$

leads to a large mismatch on order 3. This can lead to instabilities and the eventual blowup of simulations. To avoid this, instead of recomputing them at each multigrid-level from the solution at that level, the source-

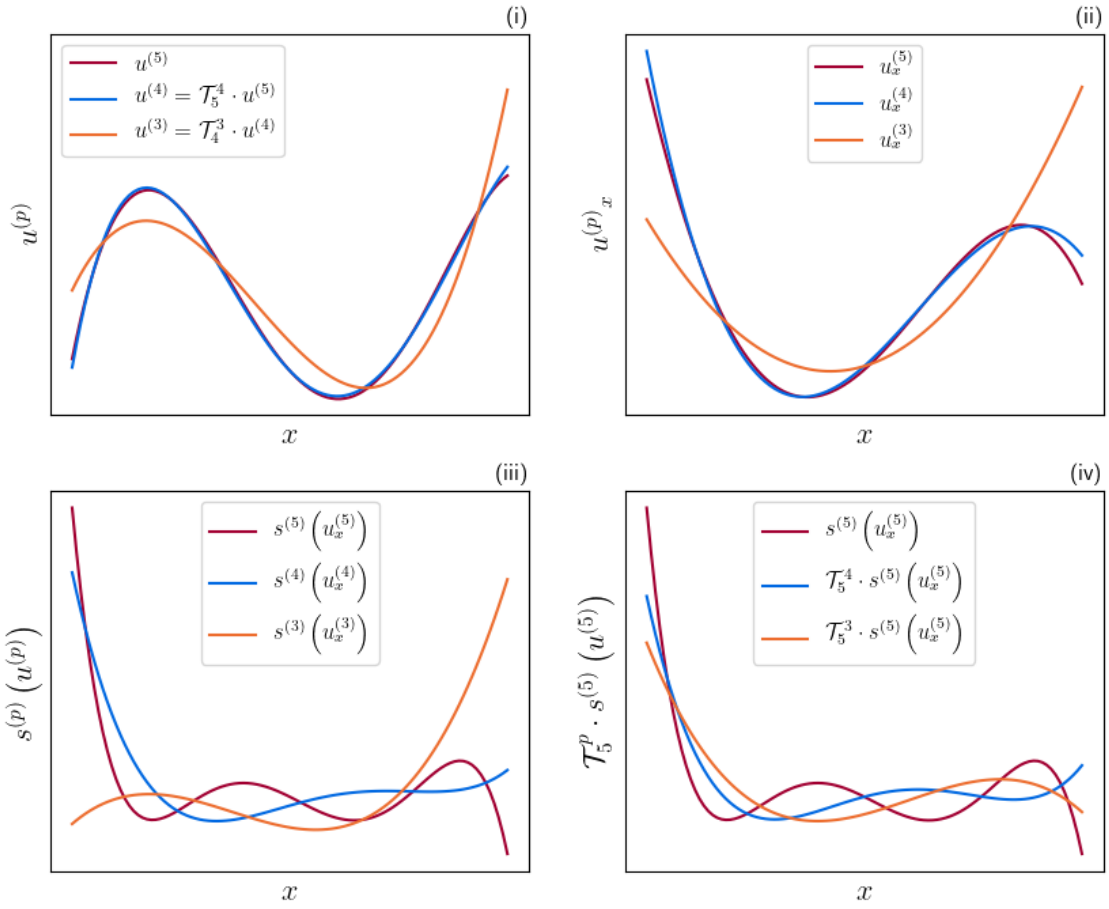


Figure 2: Illustration of source-term transfer. (i) The solution $u^{(5)}$ transferred to orders 4 and 3 during a multigrid cycle (ii) Gradient of $u^{(p)}$ (iii) Source-term at p computed using gradient of $u^{(p)}$ showing large mismatch for order 3 (iv) Source-term from order 5 transferred to coarser orders, which shows a better match across all orders.

terms at lower orders are obtained by an $\mathbb{L}^2$ -transfer of the source-term computed at the finest order (which in this example is 5), i.e.

$$s_{\text{trsfr}}^{(p)} = \mathcal{T}_5^p s^{(5)}\left(u^{(5)}\right) \quad \forall \quad p < 5. \tag{37}$$

Figure 2-(iv) shows how this leads to a better match between the source-terms at all orders. We therefore adopt source-term transfer (i.e. $s_{\text{trsfr}}^{(p)}$) for all simulations in this paper.

4.3. Local p-adaptation

Analogous to global uniform mesh refinement in FV methods, increasing the global polynomial order p is a straightforward way to improve results in HO methods. However, this leads to some regions having higher

resolution than required. A better alternative is local p-adaptation, i.e. to increase p only in regions of interest and have a lower order in others, ensuring the same accuracy for a lower computational cost. It is dynamic if the order-distribution changes throughout the simulation; in this paper however, we only deal with a fixed order-distribution throughout the simulation. For steady-state RANS simulations, the region of interest lies near the wall and in the wake. We therefore apply a local p-adaptation strategy by p-refining the cells near the flat-plate.

Let us introduce some terminology for later. The lowest polynomial-order in the simulation is called the background order and is represented by p_{bg}. The highest polynomial-order in the simulation is called the adapted order and is represented by the p_{ad}. The number of buffer-cells N_{buf} is the *minimum* number of cells of polynomial-order p that exist between cells of polynomial-orders $p+1$ and $p-1$. When calculating the interface solution and interface fluxes in *Step FR2* and *Step FR5* respectively, we need to handle the p-non-conformity for neighboring cells with different polynomial-orders. This issue is handled by transferring $\boldsymbol{U}^{\delta\text{F}}$ (or $\boldsymbol{F}_{\text{ivc}}{}^{\delta\text{F}}$ and $\boldsymbol{F}_{\text{vsc}}{}^{\delta\text{F}}$) of the lower-order cell to the higher-order using an $\mathbb{L}^2$-projection, computing the common value, and then $\mathbb{L}^2$-projecting it back to the lower order.

For p-adaptation in the context of multigrid, there remains the question of what strategy to apply when constructing the p-levels. Consider a case with the solution on the topmost level as $\boldsymbol{U}^{(2,1|h)}$. We have two options:

Option A Flatten-first: The second p-level is constructed by converting all P2 cells to P1 cells, i.e. $\boldsymbol{U}^{(1|h)}$, and then the next level as the standard $\boldsymbol{U}^{(0|h)}$.

Option B Flatten-last: The second p-level is constructed by dropping the polynomial-order of all cells by 1, i.e. $\boldsymbol{U}^{(1,0|h)}$, and then converting all P1 cells to P0 cells as $\boldsymbol{U}^{(0|h)}$.

Option B has several disadvantages:

- Most computations are performed in parallel, which raises the question of load-balancing. Typically, more smoothing-iterations are performed on the coarsest levels as compared to finer levels – this implies that the load-balancing must be optimized for coarser levels. Since the coarsest levels have a uniform polynomial-order in all cells, this inadvertently means that the levels that contain more than one orders across the domain will not be optimally balanced. *Option B* has more levels for which this load-imbalance exists and is hence computationally inefficient in comparison with *Option A* (In the example above there are two levels for *Option B* at which load-imbalance exists as opposed to just one for *Option A*).
- Along similar lines, we need to handle p-non-conformity on more levels for *Option B* – $\mathbb{L}^2$-projection is a matrix-vector product, and therefore adds to the cost.

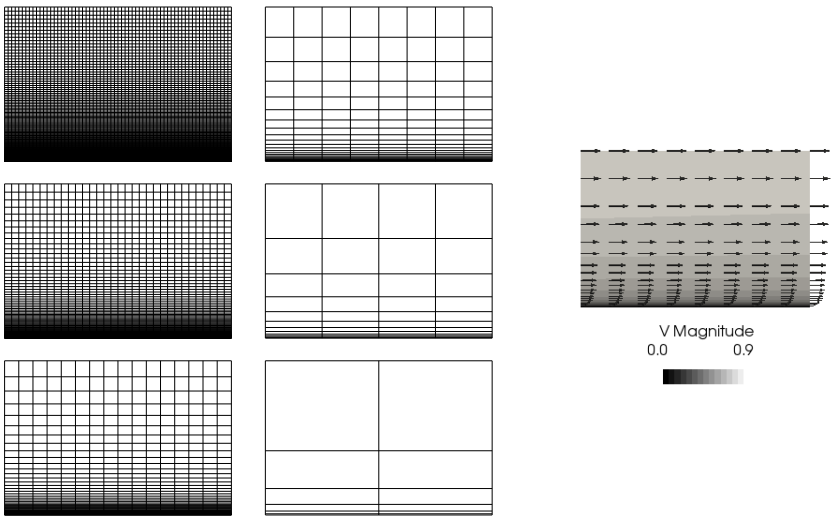


Figure 3: Grid hierarchy used for MMS[45]. Each coarse grid is obtained by removing every second line in each direction from the finer grid. On the right is $\boldsymbol{v}^{\mathrm{MS}}$ showing the realistic near-wall nature of the flow field.

Thanks to the reasons stated above, we choose *Option A* – the "flatten-first" approach to construct the p-levels for hp-multigrid with p-adaptation.

5. Verification: manufactured solutions

Code verification is an essential step to detect implementation errors in any part of the simulation framework and ensure that the discretization error is the only major source of numerical error. Here, we adopt the method of manufactured solutions (MMS) for this purpose. The core idea of MMS is to force the discrete system towards a predetermined continuous analytical "manufactured" solution $\boldsymbol{U}^{\mathrm{MS}}$ through the use of source-terms. As the degrees of freedom (DOFs) are increased in h and p we should observe the error between the converged discrete solution $\boldsymbol{U}^{\delta}$ and $\boldsymbol{U}^{\mathrm{MS}}$ drop at the expected rate of h^{p+1}. For a detailed explanation the reader is referred to a systematic study[44],[45] performed by Navah and Nadarajah.

The choice of $\boldsymbol{U}^{\mathrm{MS}}$ must investigate the code's performance on realistic RANS flows. Recently, Navah and Nadarajah[45] applied a wall-bounded manufactured solution for verification of high-order RANS solvers. Apart from well-representing realistic near-wall flow scenarios (see Figure 3-right), the presence of the no-slip wall boundary also tests the correctness of the implementation of the NSAW boundary-condition; it is imperative for accuracy of physically relevant quantities such as drag. We therefore choose their "MS4" solution as $\boldsymbol{U}^{\mathrm{MS}}$. The analysis is done for a uniform polynomial order in each run, with a series of runs with $p \in [1, 3]$. A hierarchy of six grids is used in h for each p (see Figure 3). All runs are driven to steady-state until the $\mathbb{L}^2$-norm of the residuals falls below 10^{-8}.

Figure 4 shows the drop in L_∞-error with refinement in h and p for all flow-variables. We observe that with enough refinement the rate of drop in error for all variables and all orders aligns with the expected rate of

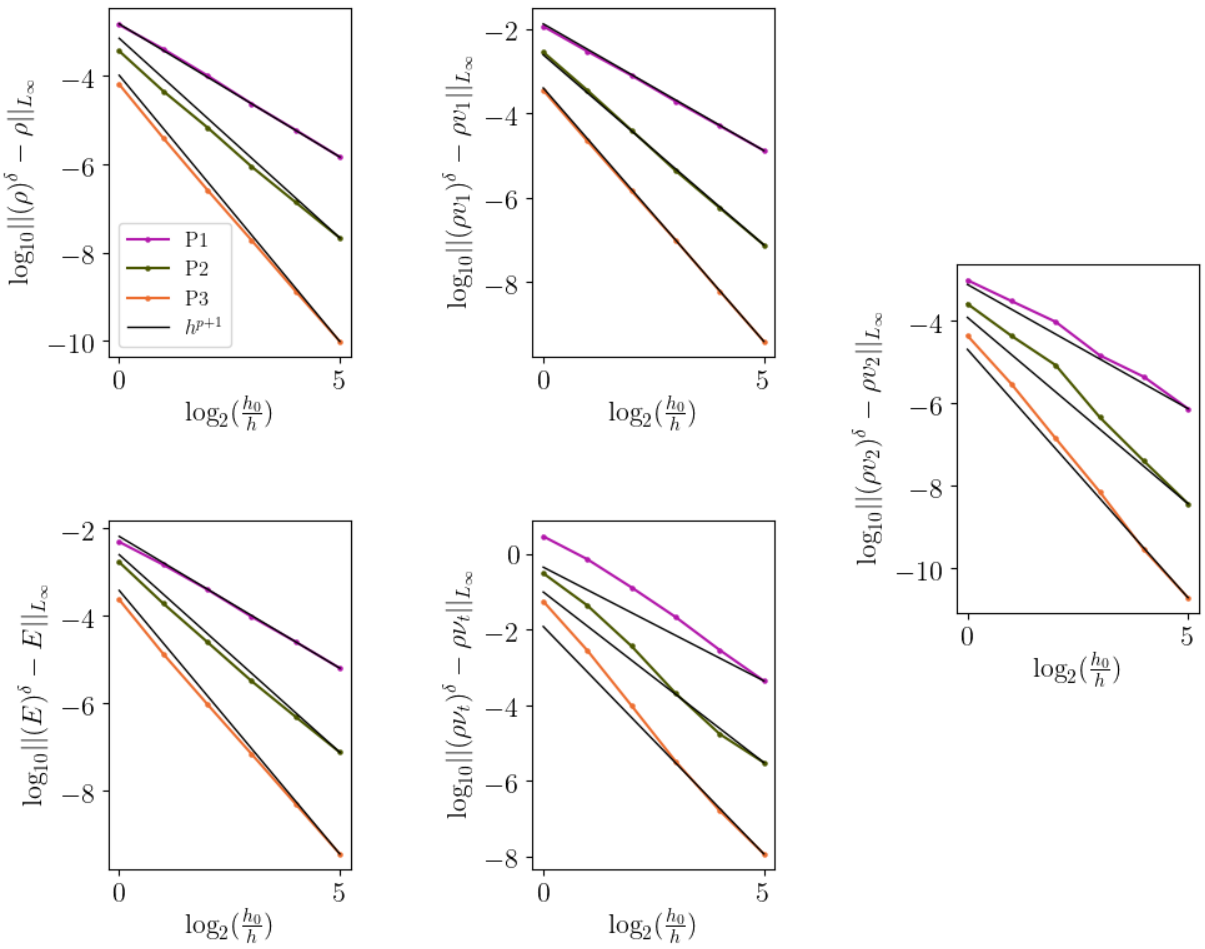


Figure 4: *hp*-convergence for MMS for all five flow-variables compared against the expected drop in error of h^{p+1}. The abscissa represents increasingly finer mesh-resolution. We observe that, with sufficiently finer meshes, all orders eventually align with the expected rate of h^{p+1}.

h^{p+1}. This verifies the correctness of the implementation.

6. Results and discussion

Having verified that the implementation is correct, we now demonstrate the effectiveness of *hp*-multigrid with *p*-adaptation on the standard steady-state turbulent flow over a flat-plate. This case belongs to the suite of validation cases of NASA's turbulence modeling resources[46]. It simulates flow with $\text{Re} = 5,000,000$ over a flat plate incident at 0°. The case-setup is described in Table 1. Flow Mach number is 0.2 at a reference temperature of $300K$. The turbulence variable ϑ takes a value of 3.0 at the inlet.

Several physical quantities of interest are monitored for all runs. These are:

- Skin-friction coefficient C_f along the wall,
- Time-history of C_f at a distance of 0.97008 m from the leading-edge (henceforth Position-X),
- Time-history of drag-coefficient C_d along the entire plate,
- Profiles of wall-normalized x-velocity v_1^+ in the x_2 direction at Position-X and at a distance of 1.90334 m from the leading-edge (henceforth Position-Y), and
- Profiles of turbulence variable ϑ in the x_2 direction at Position-X.

Unless stated otherwise, each run with polynomial-order p is accelerated to steady-state using *hp*-multigrid with local-timestepping on a *p*-hierarchy of $p+1$ levels and on the "MB" *h*-hierarchy (4 levels – see Figure 5) until the variation in C_d is below 1 count (i.e. $< 10^{-4}$).

6.1. hp-multigrid vs p-multigrid

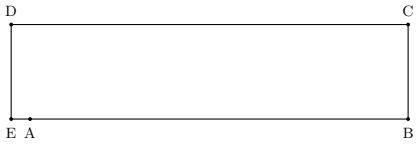
A series of runs is conducted, accelerated with *p*-multigrid and *hp*-multigrid, in both cases with *p*-adaptation. The goal is to assess the additional acceleration gained with *hp*-multigrid.

6.1.1. Fixed original-mesh and background order p_{bg}, varying adapted order p_{ad}

The first series of runs uses the same original-mesh MB_1. The background order p_{bg} is kept fixed at 2, and the adapted order p_{ad} is varied from 3 to 5.

From Figure 6 we observe a significant reduction in the number of V-cycles necessary to converge to a reasonable tolerance (the reduction-factor ranging from 4 to 6 – monitoring C_d for P5 we see that *p*-multigrid needs $\sim$ 60000 V-cycles, while *hp*-multigrid converges in $\sim$ 15000 V-cycles). The number of V-cycles does not vary much between different orders when using *hp*-multigrid, which is in agreement with the behavior observed by other authors[27], [30], [33]. Both the drag- and skin-friction coefficients converge to the reference

Table 1: Schematic diagram of the simulation domain and boundary-conditions for turbulent flow over a flat-plate.



Boundary	Length [m]	Condition	Details
AB	2	NSAW	–
BC	0.5	POUT	$T = 300K$
CD	2.1	CHAR	$\mathrm{Re} = 5e6$
DE	0.5		
EA	0.1	SYMM	–

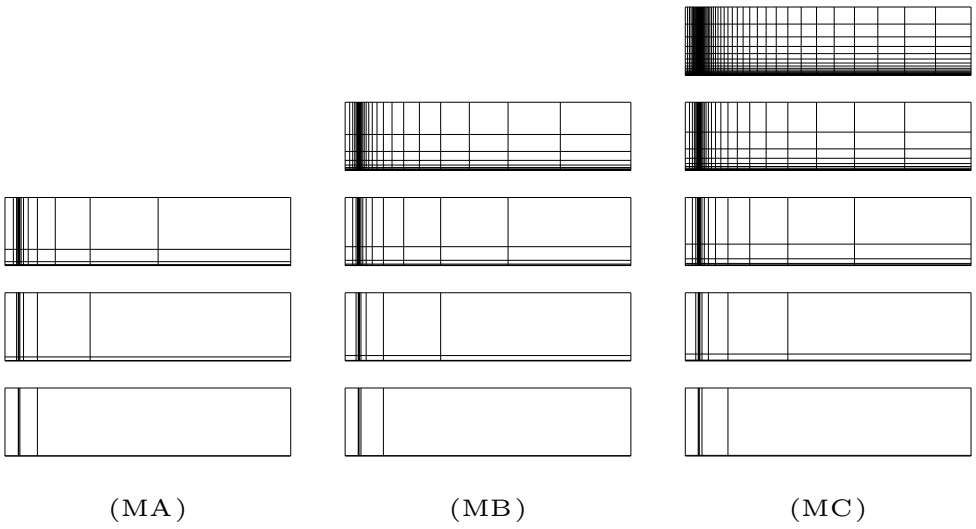


Figure 5: Hierarchies of h-meshes used in hp-multigrid for the turbulent flat-plate case. Each column represents a hierarchy-set. The top-most mesh in each set is the original mesh, and has the suffix "1". E.g. the original mesh in set MA has the label MA_1. Each subsequent mesh in the hierarchy is created by removing every other mesh-line in each direction, reducing the number of cells by a factor of 4 each time. The number of cells in the original mesh of each set is: 192 for MA_1, 768 for MB_1, and 3072 for MC_1. The x_2^+ of the first near-wall cell for the original mesh of each set is 4.

values. We also observe that hp-multigrid does not influence the quality of the results as shown by an excellent match of the profiles of all physical quantities of interest with reference data.

6.1.2. Fixed background order p_{bg} and adapted order p_{ad}, varying original-mesh

The second series of runs uses a fixed background order p_{bg} of 2 and a fixed adapted order p_{ad} of 5. The original-meshes are varied between MA_1, MB_1 and MC_1 with the respective h-multigrid mesh-hierarchy.

A clear mesh-dependence of convergence is observed for p-multigrid from Figure 7, which is reduced significantly by the use of hp-multigrid. This behavior has been demonstrated by various authors for the Poisson and Euler equations[27], [30], [33] which is hence confirm for RANS.

These experiments prove the effectiveness of the hp-multigrid over p-multigrid in reducing the V-cycle-count and achieving h and p independence in convergence.

6.2. p-convergence with p-adaptation

We now show p-convergence of quantities of interest both with and without p-adaptation. Two suites of cases are run: Suite-A contains four runs, each with uniform p in all cells. The runs are with polynomial-orders 2, 3, 4 and 5. Suite-B contains three p-adaptation runs with background order $p_{\mathrm{bg}} = 2$ and adapted order $p_{\mathrm{ad}} = 3$, 4 and 5 respectively. The number of buffer-cells between different orders $N_{\mathrm{buf}} = 3$ for all three runs.

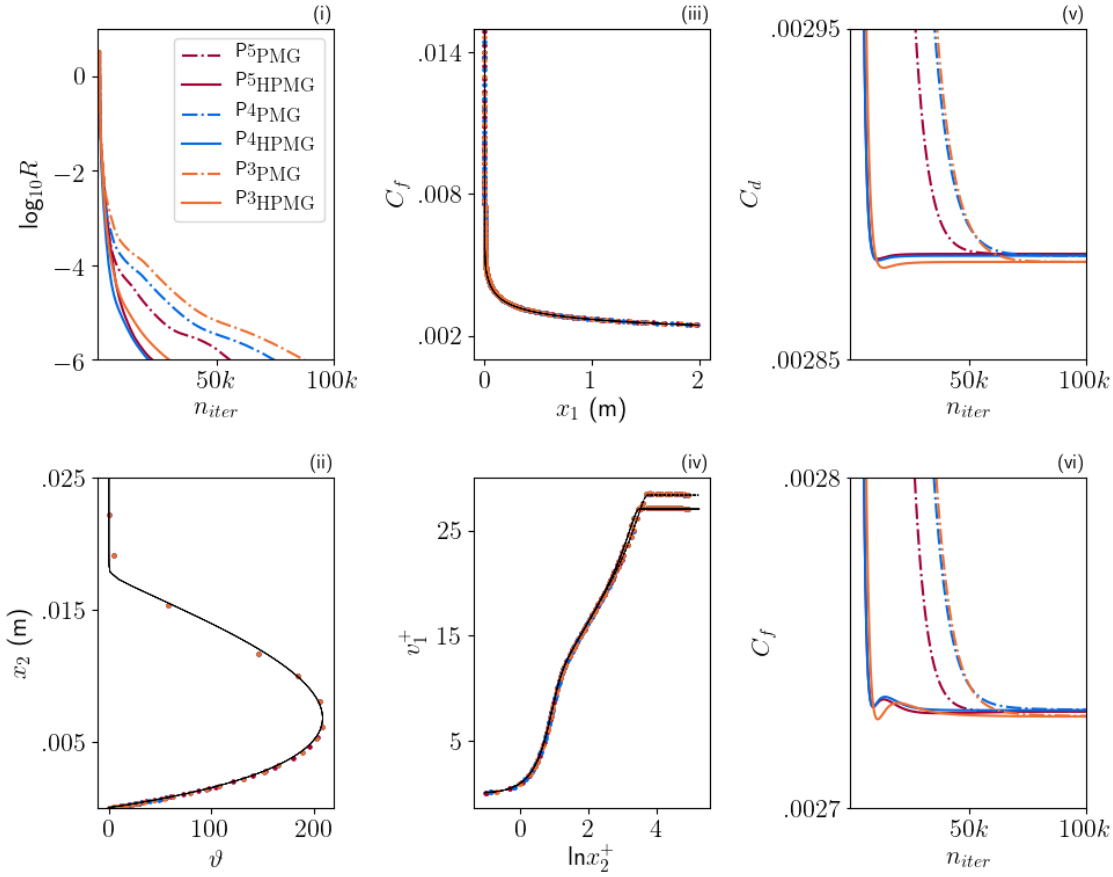


Figure 6: p-multigrid versus hp-multigrid for fixed original-mesh MA_1 and varying p_{ad}. All runs use p-adaptation. Comparison of residual evolution (i), wall-normal variation of ϑ at Position-X (ii), C_f along plate (iii), wall-normal variation of v_2 at Position-X and Position-Y (iv), with reference data (solid lines). Comparison evolution of C_d (v) and of C_f at Position-X (vi). We observe a clear reduction in the number of V-cycles necessary for convergence in hp-multigrid as compared to p-multigrid, in this case by a factor of 4 to 5. At the same time, the results agree well with reference data.

Figure 8 shows a comparison of the evolution of the drag-coefficient and of the skin-friction-coefficient at Position-X. We observe the high level of accuracy per degree-of-freedom for HO-methods as compared to FV-methods[46]. Interestingly, p-adaptation maintains the desired accuracy at fewer N_{dof} in comparison to a uniform order.

Figure 9 shows the residual evolution and several quantities of physical interest. Again, for all quantities a good agreement is observed with reference data[46]. We see a slight improvement in residual drop rate with p-adaptation- this is expected, as the dissipation of errors is faster in far-field cells which have a lower p as opposed to Suite-A runs where all cells have a uniform p.

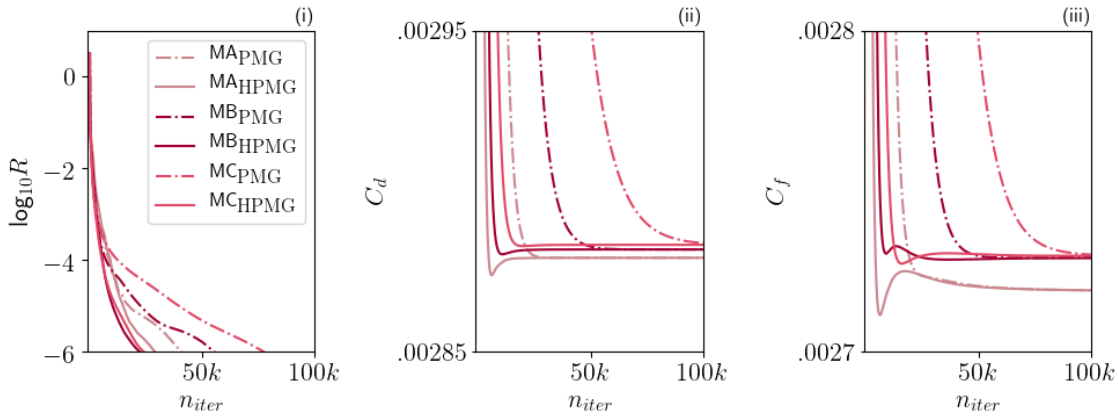


Figure 7: p-multigrid versus hp-multigrid for fixed p_{ad} of 5 and varying original-meshes. All runs use p-adaptation. Description of plots is identical to Figure 6. We observe a clear mesh-dependence for p-multigrid which is considerably reduced by hp-multigrid.

6.3. Influence of number of h-levels

So far all runs were performed with 4 h-levels. We now investigate the effect of reducing the number of h-levels in the hp-multigrid hierarchy. The mesh-hierarchies "MB" and "MC" are analyzed, starting from two h-levels (MB_1 – MB_2 and MC_1 – MC_2) to all h-levels (MB_1 – MB_4 and MC_1 – MC_5). All other parameters are left unchanged. From Figure 10 and Figure 11 we observe that increasing the number of h-levels improves the rate of convergence, with the first few levels having the most impact.

7. Conclusion

This study analyzes hp-multigrid with cell-local time-stepping as a convergence accelerator, particularly in combination with static near-wall p-adaptation, for high-order flux reconstruction methods applied to RANS-modeled turbulent flow. Turbulence is accounted for using the one-equation Spalart-Allmaras model with modifications for handling negative values of eddy-viscosity near the edge of the boundary layer. Source-terms in the ϑ-equation are transferred from the finest p-level to coarser levels using an $\mathbb{L}^2$-projection to avoid aliasing errors for sharp changes in ϑ. Correctness of the implementation is demonstrated through spectral convergence using realistic near-wall manufactured solutions on inflation-layer type meshes. Numerical experiments are conducted on the standard case of turbulent flow over a flat-plate[46]. We show the following major observations:

1. hp-multigrid adds a significant speed-up over only p-multigrid (factor of 4 for P5) in highly anisotropic meshes characteristic of boundary layers,

2. Near-wall p-adaptation with hp-multigrid achieves the same accuracy as uniform polynomial-order and

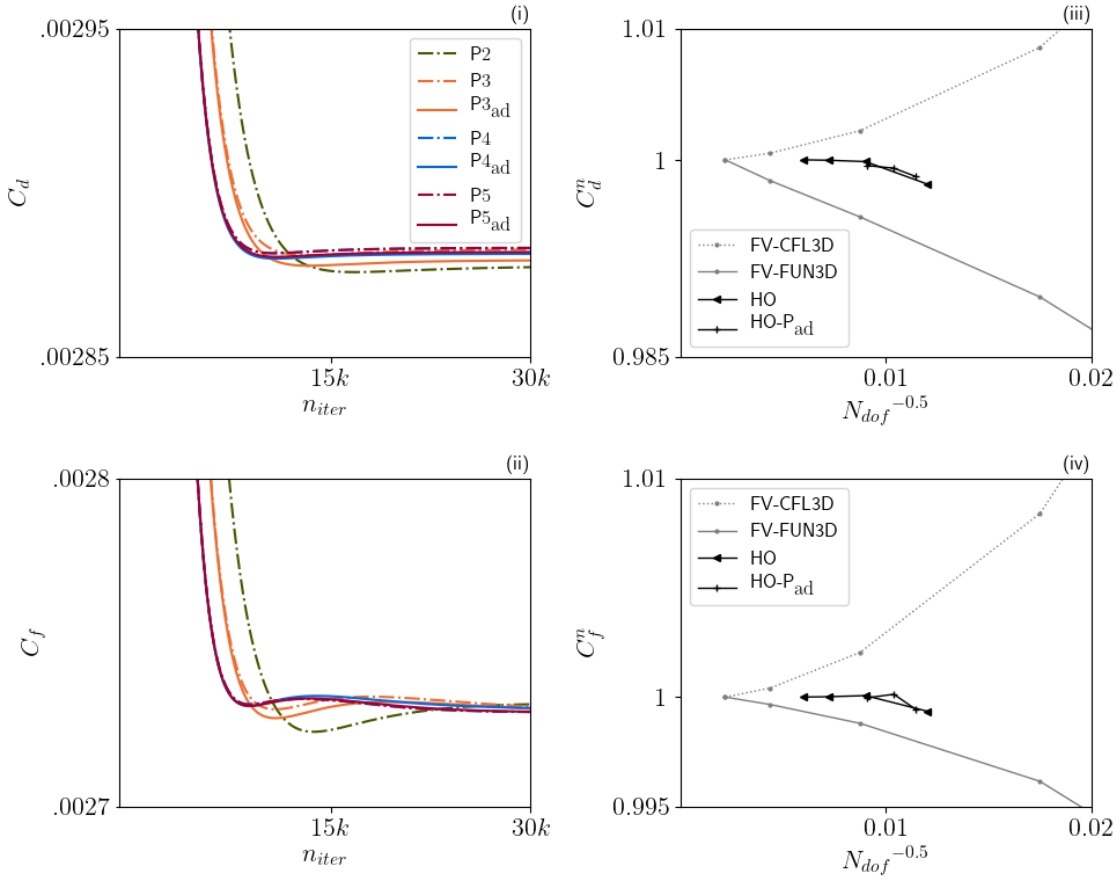


Figure 8: Uniform p versus p-adaptation. All runs use hp-multigrid. Comparison evolution of C_d (i) and of C_f at Position-X (ii). Plot of $C_d^n = C_d/C_d^{\text{ref}}$ (iii) and $C_f^n = C_f/C_f^{\text{ref}}$ (iv) against $N_{dof}^{-0.5}$ compared with state-of-the-art FV data. It shows the power of HO methods to gain the same accuracy as FV-methods at much lower N_{dof}. This is further improved with p-adaptation as seen in the figure.

at a lower number of degrees-of-freedom, enhancing the already commendable accuracy-per-N_{dof} ratio of HO-methods, and

3. Adding even a single h-level over p-multigrid gives a massive advantage, reducing the number of V-cycles by $\sim 60\%$.

These observations conclusively prove the benefits of hp-multigrid and p-adaptation for FR-methods, particularly for RANS-modeled turbulent flows where the near-wall region needs to be well-resolved for accuracy. Our future work will focus on applying the hp-multigrid-p-adaptation combination to more complicated scenarios in 3D.

Conflict of interest

The work presented in this paper does not have any conflict of interest with other organizations.

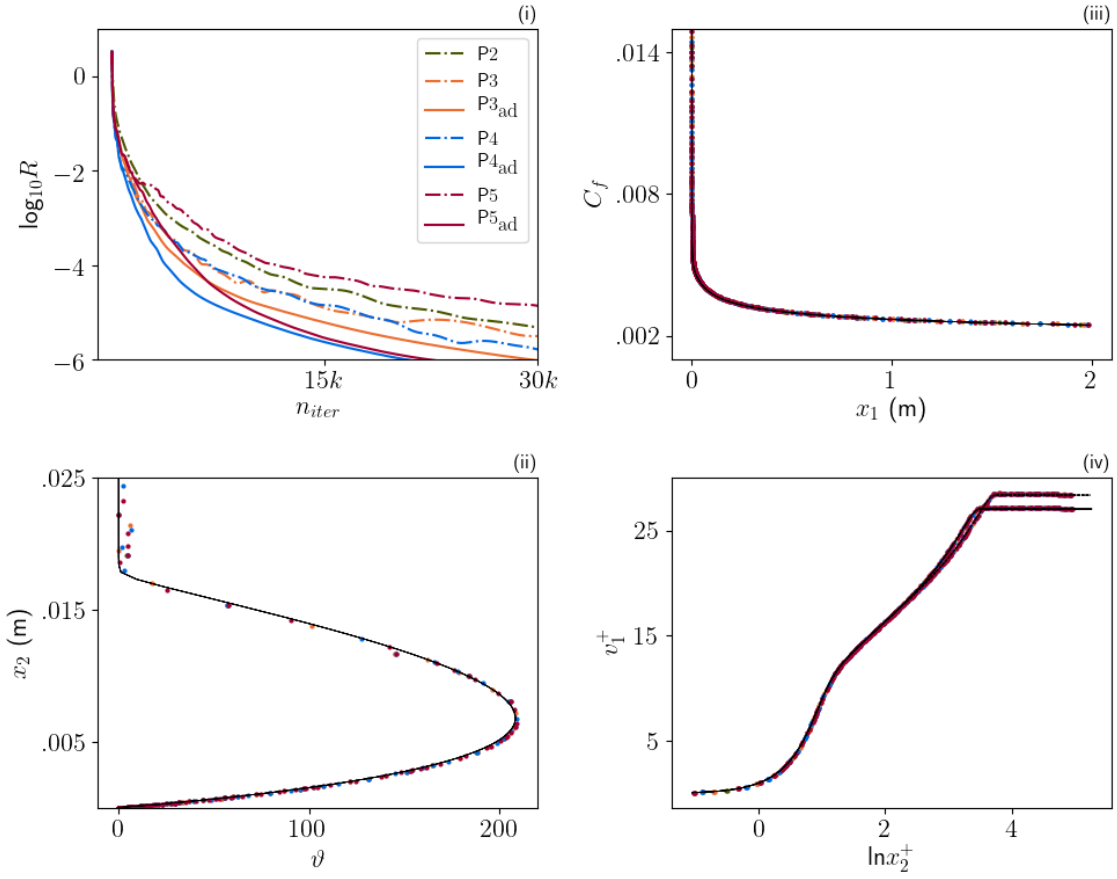


Figure 9: Uniform p versus p-adaptation. All runs use hp-multigrid. Comparison of residual evolution (i), C_f along plate (iii), wall-normal variation of ϑ at Position-X (ii) and wall-normal variation of v_2 at Position-X and Position-Y, with reference data (iv). They clearly demonstrate that p-adaptation conserves the accuracy of uniform p for a lower number of degrees-of-freedom. There is also a slightly faster drop of residuals.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement number 813605[47].

References

[1] Lee, M. and Moser, R.D.: Direct numerical simulation of a turbulent channel flow up to $\mathrm{Re}_\tau \approx 5200$, J. Fluid. Mech. (2015)

[2] Slotnick, J. et al: CFD vision 2030 study: A path to revolutionary computational aerosciences (2014)

[3] Argyropoulus, C.D., Markatos, N.C.: Recent advances on the numerical modelling of turbulent flows, App. Math. Mod. (2015)

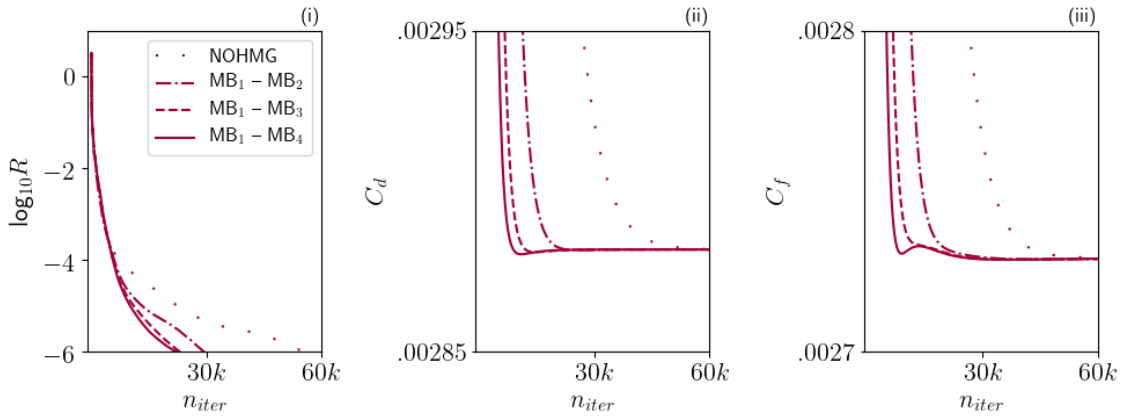


Figure 10: Effect of varying the number of h-levels for the "MB" mesh-hierarchy on the evolution of (i) residuals, (ii) C_d and (iii) C_f at Position-X. Even though there is a non-negligible effect of increasing the number of h-levels, it is clear that the first few h-levels have the strongest impact.

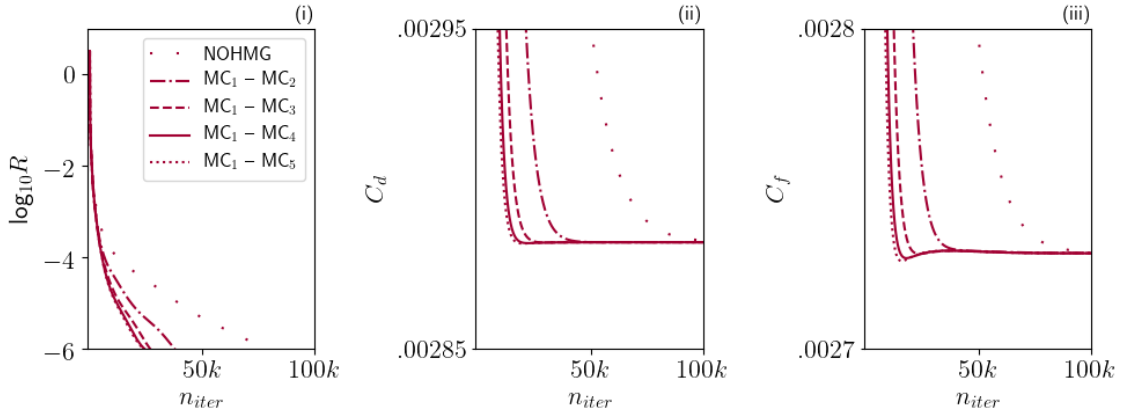


Figure 11: Effect of varying the number of h-levels for the "MC" mesh-hierarchy. A similar trend as Figure 10 is observed.

[4] Choi, H. and Moin, P. Grid-point requirements for large eddy simulations: Chapman's esitmates revisited, Phys. Fluids (2012)

[5] Bose, S.T. and Park, G.I.: Wall-modeled large-eddy simulation for complex turbulent flows, Ann. Review. of Fluid Mech. (2018)

[6] Heinz, S.: A review of hybrid RANS-LES methods for turbulent flows: concepts and applications, Progress in Aero. Sci. (2020)

[7] Bian, X. et. al.: Hybrid RANS/LES study of complex turbulence characteristics and flow mechanisms on the highly-loaded turbine endwall, Aero. Sci. and Tech. (2019)

[8] Ekman, P. et. al.: Assessment of hybrid RANS-LES methods for accurate automotive aerodynamic simulations, J. Wind. Engg. and Indus. Aero. (2020)

[9] Rumsey, C.L. and Slotnick, J.P.: Overview and summary of the second AIAA high-lift prediction workshop, AIAA Paper (2014)

[10] Reed, W.H. and Hill, T.R.: Triangular mesh methods for the neutron transport equation, United States (1973)

[11] Abgrall, R., Ricchiuto, M.: High order methods for CFD, Encyclopedia of Computational Mechanics (2017)

[12] Hesthaven, J.S., Warburton, T.: Nodal discontinuous Galerkin methods: algorithms, analysis, and applications, Springer (2008)

[13] Vincent, P.E., Castonguay, P., Jameson, A.: A new class of high-order energy stable flux reconstruction schemes, J. Sci. Comp. (2011)

[14] Huynh, H.T.: A flux reconstruction approach to high-order schemes including discontinuous galerkin methods, 18th AIAA CFD Conference (2007)

[15] Huynh, H.T., Wang, Z.J., Vincent, P.E.: High-order methods for computational fluid dynamics: A brief review of compact differential formulations on unstructured grids, Computers and Fluids (2014)

[16] Wesseling, P. and Oosterlee, C.W.: Geometric multigrid with applications to computational fluid dynamics, J. Comp. and Applied. Math. (2001)

[17] Stüben, K.: A review of algebraic multigrid, J. Comp. and Applied. Math. (2001)

[18] Brandt, A.: Multi-level adaptive solutions to boundary value problems, Math. Comp. (1977)

[19] Fehn, N. et. al.: Efficiency of high-performance discontinuous Galerkin spectral element methods for under-resolved turbulent incompressible flows, Int. J. for Num. Methods in Fluids (2018)

[20] Botti, L. et. al.: h-multigrid agglomeration based solution strategies for discontinuous Galerkin discretizations of incompressible flow problems, J. Comp. Phy. (2017)

[21] Krank, B. et. al.: A high-order semi-explicit discontinuous Galerkin solver for 3D incompressible flow with application to DNS and LES of turbulent channel flow, J. Comp. Phy. (2017)

[22] Klaij, C.M. et. al.: h-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comp. Phy. (2007)

[23] Haga, T. et. al.: Efficient solution techniques for high-order methods on 3D anisotropic hybrid meshes, AIAA Aerospace Science Meet (2011)

[24] Luo, H. et. al.: An implicit discontinuous Galerkin method for the unsteady compressible Navier–Stokes equations, Computers Fluids (2012)

[25] Ghidoni, A. et. al.: Efficient p-multigrid discontinuous Galerkin solver for complex viscous flows on stretched grids, Int. J. for Num. Methods in Fluids (2014)

[26] Franciolini, M. et. al.: p-Multigrid matrix-free discontinuous Galerkin solution strategies for the under-resolved simulation of incompressible turbulent flows, Computers Fluids (2020)

[27] Fehn, N. et. al.: Hybrid multigrid methods for high-order discontinuous Galerkin discretizations, J. Comp. Phy. (2020)

[28] Wallraff, M. and Leicht, T.: Higher order multigrid algorithms for a discontinuous Galerkin RANS solver, AIAA SciTech (2014)

[29] Wallraff, M. et. al., Multigrid solver algorithms for DG methods and applications to aerodynamic flows, IDIHOM, Springer (2015)

[30] Kronbichler, M. and Wall, W.A.: A performance comparison of continuous and discontinuous Galerkin methods with fast multigrid solvers, SIAM Journal on Scientific Computing (2018)

[31] Nastase, C.R. and Mavriplis, D.J.: High-order discontinuous Galerkin methods using an hp-multigrid approach, J. Comp. Phy. (2006)

[32] Shahbazi, K., et. al.: Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comp. Phy. (2009)

[33] Sundar, H. et. al.: Comparison of multigrid algorithms for high-order continuous finite element discretizations, Numerical Linear Algebra with Applications (2015)

[34] Menier, V. et. al.: Multigrid strategies coupled with anisotropic mesh adaptation, AIAA Aerospace Sciences Meet (2015)

[35] Wang, L. et. al.: A dynamically load-balanced parallel p-adaptive implicit high-order flux reconstruction method for under-resolved turbulence simulation, J. Comp. Phy. (2020)

[36] Castonguay, P.: High-order energy stable flux reconstruction schemes for fluid flow simulations on unstructured grids, Ph.D. thesis, Stanford University (2012)

[37] Schlichting, H.: Boundary Layer Theory, McGraw-Hill (1979)

[38] Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flows, 30th AIAA meeting (1992)

[39] Oliver, T.A.: A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-Averaged Navier-Stokes equations. MIT, Cambridge, Dept. of Aeronautics and Astronautics (2008)

[40] Allmaras, S.R. et. al.: Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model, ICCFD7-1902 (2012)

[41] Rueda-Ramírez, A.M. et. al.: A p-multigrid strategy with anisotropic p-adaptation based on truncation errors for high-order discontinuous Galerkin methods, J. Comp. Phy. (2019)

[42] Vermeire, B.C. et. al.: Optimal Runge–Kutta schemes for pseudo time-stepping with high-order unstructured methods, J. Comp. Phy. (2019)

[43] Mengaldo, G. et. al.: A Guide to the implementation of boundary conditions in compact high-order methods for compressible aerodynamics, AIAA-TFMC (2014)

[44] Navah, F., Nadarajah, S.: A comprehensive high-order solver verification methodology for free fluid flows, Aerospace Sci. and Tech. (2018)

[45] Navah, F., Nadarajah, S.: On the verification of CFD solvers of all orders of accuracy on curved wall-bounded domains and for realistic RANS flows, Computers and Fluids (2020)

[46] `https://turbmodels.larc.nasa.gov/flatplate_sa.html` (last accessed 23.10.2020)

[47] Advanced High-Order Simulation Methods for Industrial Applications. `https://cordis.europa.eu/project/id/813605`