Journal of Computational Physics 231 (2012) 7537–7563

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

hp-Multigrid as Smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows: Part I. Multilevel analysis

J.J.W. van der Vegt [a,*], S. Rhebergen [a,b]

[a] Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 Enschede, AE, The Netherlands
[b] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 12 November 2011
Received in revised form 7 May 2012
Accepted 28 May 2012
Available online 19 June 2012

Keywords:
Multigrid algorithms
Discontinuous Galerkin methods
Higher order accurate discretizations
Space–time methods
Runge–Kutta methods
Fourier analysis
Multilevel analysis

ABSTRACT

The *hp*-Multigrid as Smoother algorithm (*hp-MGS*) for the solution of higher order accurate space–time discontinuous Galerkin discretizations of advection dominated flows is presented. This algorithm combines *p*-multigrid with *h*-multigrid at all *p*-levels, where the *h*-multigrid acts as smoother in the *p*-multigrid. The performance of the *hp-MGS* algorithm is further improved using semi-coarsening in combination with a new semi-implicit Runge–Kutta method as smoother. A detailed multilevel analysis of the *hp-MGS* algorithm is presented to obtain more insight into the theoretical performance of the algorithm. As model problem a fourth order accurate space–time discontinuous Galerkin discretization of the advection–diffusion equation is considered. The multilevel analysis shows that the *hp-MGS* algorithm has excellent convergence rates, both for steady state and time-dependent problems, and low and high cell Reynolds numbers, including highly stretched meshes.

1. Introduction

Discontinuous Galerkin finite element methods are well suited to obtain higher order accurate discretizations on unstructured meshes. The use of basis functions which are only weakly coupled to neighboring elements results in a local discretization which allows, in combination with *hp*-mesh adaptation, the efficient capturing of detailed structures in the solution, and is also beneficial for parallel computing. During the past decade this has stimulated a large amount of research in both the development and analysis of DG methods and resulted in a wide variety of applications. For an overview of various aspects of DG methods, see e.g. [6,12].

Space–time discontinuous Galerkin methods are a special class of DG methods in which space and time are simultaneously discretized using basis functions which are discontinuous, both in space and time. The resulting discretization belongs to the class of arbitrary Lagrangian Eulerian (ALE) methods, is implicit in time and fully conservative on moving and deforming meshes as occur in fluid–structure interaction and free boundary problems, see e.g. [14,29,30,33].

For higher order accurate DG discretizations the efficient solution of the algebraic system resulting from an implicit time discretization is, however, non-trivial, in particular for steady state solutions of advection dominated flows. For these problems standard iterative techniques, such as multigrid and Krylov subspace methods, are generally suboptimal, especially on highly stretched meshes in boundary layers. This lack of computational efficiency currently seriously hampers the

* Corresponding author.
E-mail addresses: j.j.w.vandervegt@math.utwente.nl (J.J.W. van der Vegt), srheberg@umn.edu (S. Rhebergen).

0021-9991/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2012.05.038

application of higher order accurate DG methods to large scale industrial applications. An important reason for this relatively slow convergence rate is that the algebraic system resulting from a higher order accurate DG discretization has quite different mathematical properties compared to lower order discretizations. The straightforward application of iterative techniques originally developed for lower order methods is therefore generally not optimal and should be supported by a more detailed mathematical analysis.

The need for improved convergence rates in the iterative solution of higher order accurate DG discretizations has motivated the research presented in this and the companion article [32], to which we will refer as Part II. The objectives of this research are to develop, analyze and optimize new multigrid algorithms for higher order accurate space–time DG discretizations of advection dominated flows. For this purpose we introduce the *hp*-Multigrid as Smoother algorithm. This algorithm combines *p*-multigrid with *h*-multigrid at all *p*-levels, where the *h*-multigrid acts as smoother in the *p*-multigrid. The theoretical tool to investigate the performance of the *hp-MGS* algorithm will be a detailed multilevel analysis, which is the main topic of this article. In Part II we will use this analysis to optimize the semi-implicit Runge–Kutta smoother in the *hp-MGS* algorithm in order to account for the special features of higher order accurate DG discretizations. In addition, numerical simulations will be presented which show the excellent performance of the *hp-MGS* algorithm on a number of test cases, including thin boundary layers and non-constant coefficients. In this article we will focus on space–time DG discretizations, but the results and techniques can be straightforwardly extended to other types of implicit DG discretizations, both for steady state and time-accurate problems.

As background information we start with a brief overview of the main algorithms developed during the past decade for the iterative solution of higher order accurate DG discretizations of the compressible Euler and Navier–Stokes equations, which are important models for advection dominated flows. The main techniques to solve these equations have been multigrid and preconditioned Krylov methods, in particular flexible GMRES. In this article we will focus on multigrid methods. For preconditioned Krylov methods we refer to [7,23,26]. Multigrid methods can, however, also be efficient preconditioners for flexible GMRES, see e.g. [26].

Multigrid methods applied to higher order accurate DG discretizations can be classified as *p*-, *h*-, and *hp*-multigrid methods. In *p*-multigrid the coarser levels are obtained using a sequence of lower order discretizations, whereas in *h*-multigrid coarser meshes are used. Here p refers to the polynomial order of the basis functions in the DG discretization and h to the mesh size. Combinations of both methods result in *hp*-multigrid.

The main benefit of *p*-multigrid is its simplicity since at all levels the same mesh is used, which makes the implementation on unstructured meshes straightforward. Applications of *p*-multigrid to higher order accurate DG discretizations of advection dominated flows can be found in [2,8,17–19,21]. The resulting algebraic system at the coarsest *p*-multigrid level can, however, still be very large. For the Euler equations an implicit Euler time integration method at the $p = 0$ level, with GMRES in combination with an ILU preconditioner or an LU-SGS algorithm to solve the resulting algebraic system, is suitable [2,17]. For the compressible Navier–Stokes equations an *hp*-multigrid method is a better alternative [21,26]. In most studies of the compressible Navier–Stokes equations a polynomial order $p = 1$ is used at the coarsest level, which gives significantly better results than $p = 0$, see e.g. [26]. In this multigrid method the algebraic system at the coarsest *p*-level is solved with an *h*-multigrid method. For nonlinear problems it was concluded in [26] that the linear or Newton *h*-multigrid method is significantly more efficient as a coarse grid solver in *hp*-multigrid than the nonlinear Full Approximation Scheme.

A crucial element in both *p*- and *hp*-multigrid algorithms are the smoothers. Many different types of smoothers have been tested for higher order accurate DG discretizations. A serious problem with many of these smoothers is their lack of robustness. Often significant under-relaxation is necessary to ensure stability of the iterative method. Under-relaxation is, however, not necessary when block Jacobi and (symmetric) block Gauss–Seidel methods are used [8,18,21,26]. For problems containing boundary layers line smoothers are generally necessary to deal with large aspect ratio meshes [8,26]. Explicit and (semi)-implicit time integration methods have also been used as smoothers [2,3,16,25]. In particular, Runge–Kutta methods can be developed into efficient multigrid smoothers when they are used as pseudo-time integrators, which was originally proposed in [13], see also [20]. Since time-accuracy is not important in pseudo-time significant freedom is available to optimize Runge–Kutta smoothers for good multigrid performance [16,25,29].

The theoretical analysis of multigrid algorithms for DG discretizations of advection dominated flows has been quite limited. Many of these studies considered the advection–diffusion equation or linearized versions of the compressible Euler equations. The main analysis tool to understand the performance of the multigrid algorithm has been single grid and two-level Fourier analysis [8,9,16,18,24,25,33]. For a more general discussion of these techniques we refer to [11,28,35,37].

Despite this extensive research currently available multigrid algorithms for higher order DG discretizations do not yet achieve optimal performance. In this article we present therefore a new approach, viz. the *hp*-Multigrid as Smoother (*hp-MGS*) algorithm. The *hp-MGS* algorithm is an extension of the Multigrid as Smoother algorithm, which was originally proposed in [22,34], to higher order accurate DG discretizations. The main focus in this article is on the multilevel analysis of the *hp-MGS* algorithm, which is crucial to understand and optimize its performance. In the multilevel analysis three *p*-levels and three uniformly and three semi-coarsened *h*-levels are used in order to obtain accurate estimates of the operator norms and spectral radius of the *hp-MGS* multigrid error transformation operator. In Part II this analysis will be used to optimize the coefficients in the semi-implicit Runge–Kutta smoother for a fourth order accurate space–time DG discretization of the two-dimensional advection–diffusion equation.

The outline of this article is as follows. In Section 2 we briefly discuss the space–time DG discretization and in Section 3 we introduce the *hp-MGS* algorithm and the semi-implicit Runge–Kutta smoother. The multigrid error transformation operator of the *hp-MGS* algorithm is briefly discussed in Section 4 and a detailed description of the multilevel Fourier analysis of the *hp-MGS* algorithm is given in Section 5. In Section 6 the multilevel analysis of the *hp-MGS* algorithm is used to investigate the performance of the *hp-MGS* algorithm for a fourth order accurate space–time DG discretization of the two-dimensional advection–diffusion equation. Finally, conclusions are drawn in Section 7.

2. Space–time DG discretization of the advection–diffusion equation

In a space–time DG formulation, the space and time variables are discretized simultaneously. The space–time framework is well suited for problems formulated on time dependent domains, but in this study we consider the advection–diffusion equation on a fixed space–time domain as a model problem for the multigrid analysis. A point at time $t = x_0$, with position vector $\bar{x} = (\bar{x}_1, \dots, \bar{x}_d) \in \mathbb{R}^d$, has Cartesian coordinates $x = (x_0, \bar{x})$ in the open domain $\mathcal{E} = \Omega \times (t_0, T) \subset \mathbb{R}^{d+1}$, with t_0 and T the initial and final time of the solution and $\Omega \subset \mathbb{R}^d$ the spatial domain with dimension d. For simplicity we assume here that Ω is a polyhedral domain. The 2D advection–diffusion equation for a scalar function $u : \mathcal{E} \to \mathbb{R}$ can be written as

$$\begin{cases} \frac{\partial u}{\partial t} + \nabla \cdot (au) = \nu \triangle u, & \text{on } \mathcal{E}, \\ u(t_0, \bar{x}) = u_0(\bar{x}), & \text{for } \bar{x} \in \Omega, \\ u(t, \bar{x}) = u^b(t, \bar{x}), & \text{for } \bar{x} \in \partial\Omega,\ t \in (t_0, T), \end{cases}$$

where $\nu \in \mathbb{R}^+$ is a constant diffusion coefficient, $a \in \mathbb{R}^d$ the advection velocity, and $\nabla = \left(\frac{\partial}{\partial \bar{x}_1}, \dots, \frac{\partial}{\partial \bar{x}_d}\right) \in \mathbb{R}^d$ the nabla operator. Furthermore, the Laplacian operator is denoted as $\triangle$, the initial flow field by u_0 and the boundary data by u^b.

The space–time domain is subdivided into space–time slabs, which are defined as $\mathcal{E}^n := \{(t, \bar{x}) \in \mathcal{E} \mid t \in (t_n, t_{n+1})\}$. In the spatial domain Ω we define a tessellation T_h by subdividing the domain into non-overlapping quadrilateral ($d = 2$) or hexahedral ($d = 3$) spatial elements $K \subset \mathbb{R}^d$. In each space–time slab we introduce now the space–time tessellation $\mathcal{T}_h^n := \{\mathcal{K} := K \times (t_n, t_{n+1}) \subset \mathcal{E}^n \mid \cup\mathcal{K} = \mathcal{E}^n\}$. Each of the space–time elements $\mathcal{K} \in \mathcal{T}_h^n$ is connected to the reference element $\hat{\mathcal{K}} := (-1, 1)^d$ using the isoparametric mapping $G_{\mathcal{K}}^n$. Within a space–time slab we distinguish faces connecting space–time slabs, viz. $K(t_n) := K \times \{t_n\}$ and $K(t_{n+1})$, internal faces $\mathcal{S}_I^n := \{\mathcal{S} \subset \partial\mathcal{K}^- \cap \partial\mathcal{K}^+ \mid \mathcal{K}^\pm \in \mathcal{T}_h^n,\ \mathcal{K}^- \cap \mathcal{K}^+ = \emptyset\}$ and boundary faces $\mathcal{S}_B^n := \{\mathcal{S} \subset \partial\mathcal{K} \cap \partial\Omega \mid \mathcal{K} \in \mathcal{T}_h^n\}$. The outward space–time normal vector at the boundary $\partial\mathcal{K}$ of a space–time element $\mathcal{K} \in \mathcal{T}_h^n$ is denoted by $n = (n_t, \bar{n}) \subset \mathbb{R}^{d+1}$, with n_t the temporal and $\bar{n}$ the spatial part of n. On an internal face $\mathcal{S} \in \mathcal{S}_I$, the traces from the left and right element are denoted by $(\cdot)^-$ and $(\cdot)^+$, respectively. The average operator is defined as $\{\{\cdot\}\} = \frac{1}{2}((\cdot)^- + (\cdot)^+)$ and the jump operator as $[[\cdot]] = (\cdot)^-\bar{n}^- + (\cdot)^+\bar{n}^+$, where $n^+ = -n^-$.

We consider approximations u_h and test functions v in the finite element space W_h, defined as

$$W_h = \left\{W \in L_2(\mathcal{E}^n) \mid W|_{\mathcal{K}} \circ G_{\mathcal{K}}^n \in P^p(\hat{\mathcal{K}}), \quad \forall \mathcal{K} \in \mathcal{T}_h\right\},$$

where $L_2(\mathcal{E}^n)$ is the space of square integrable functions on $\mathcal{E}^n$ and $P^p(\hat{\mathcal{K}})$ the space of polynomials of maximum degree p on the reference element $\hat{\mathcal{K}}$. Furthermore, we also need the following space

$$V_h = \left\{V \in (L_2(\mathcal{E}^n))^d \mid V|_{\mathcal{K}} \circ G_{\mathcal{K}}^n \in (P^p(\hat{\mathcal{K}}))^d, \quad \forall \mathcal{K} \in \mathcal{T}_h\right\}.$$

The space–time DG weak formulation of the two-dimensional advection–diffusion equation can now be formulated as: Find a $u_h \in W_h$, such that for all $v \in W_h$

$$\begin{aligned} &-\sum_{\mathcal{K}\in\mathcal{T}_h^n} \int_{\mathcal{K}} \left(\frac{\partial v}{\partial t} u_h + \nabla v \cdot a u_h - \nu \nabla v \cdot \nabla u_h\right) d\mathcal{K} + \sum_{\mathcal{S}\in\mathcal{S}_I^n} \int_{\mathcal{S}} [[v]] \cdot \hat{f}(u_h^-, u_h^+) d\mathcal{S} + \sum_{\mathcal{K}\in\mathcal{T}_h^n} \left(\int_{K(t_{n+1})} v^- u_h^- dK - \int_{K(t_n)} v^- u_h^+ dK\right) \\ &\quad + \sum_{\mathcal{S}\in\mathcal{S}_B^n} \int_{\mathcal{S}} v^- \hat{f}(u_h^-, u_h^+) \cdot \bar{n} d\mathcal{S} - \sum_{\mathcal{S}\in\mathcal{S}_I^n} \int_{\mathcal{S}} [[v]] \cdot \nu \{\{\nabla u_h - \eta^{\mathcal{S}} \mathcal{R}_h^{\mathcal{S}}\}\} d\mathcal{S} - \sum_{\mathcal{S}\in\mathcal{S}_B^n} \int_{\mathcal{S}} v^- \nu (\nabla u_h^- - \eta^{\mathcal{S}} \mathcal{R}_h^{\mathcal{S}}) \cdot \bar{n} d\mathcal{S} \\ &\quad - \sum_{\mathcal{S}\in\mathcal{S}_I^n} \int_{\mathcal{S}} \nu \{\{\nabla v\}\} \cdot [[u_h]] d\mathcal{S} - \sum_{\mathcal{S}\in\mathcal{S}_B^n} \int_{\mathcal{S}} \nabla v^L \cdot \nu (u_h^- - u^b) \bar{n} d\mathcal{S} = 0. \end{aligned} \tag{1}$$

Here $u_h^\pm = \lim_{\epsilon\downarrow 0} u_h(x \pm \epsilon n_{\mathcal{K}})$, with $n_{\mathcal{K}}$ the outward space–time normal vector at $\partial\mathcal{K}$ and $\hat{f}(u_h^-, u_h^+)$ an upwind numerical flux. The space–time formulation (1) uses a space–time generalization of the approaches by Bassi and Rebay [1] and Brezzi [5] for the discretization of the viscous flux. The local lifting operator $\mathcal{R}_h^{\mathcal{S}}$ is defined as in [14]: Find an $\mathcal{R}_h^{\mathcal{S}} \in V_h$, such that for all $w \in V_h$

$$\sum_{\mathcal{K}\in\mathcal{T}_h^n} \int_{\mathcal{K}} w \cdot \mathcal{R}_h^{\mathcal{S}} d\mathcal{K} = \begin{cases} \int_{\mathcal{S}} \{\{w\}\} \cdot [[u_h]] d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_I^n, \\ \int_{\mathcal{S}} w^L \cdot (u_h^- - u^b) \bar{n}^L d\mathcal{S} & \text{for } \mathcal{S} \in \mathcal{S}_B^n. \end{cases} \tag{2}$$

The stabilization parameter $\eta^{\mathcal{S}}$ is constant and should be chosen greater than or equal to the number of space-faces of an element, see [27].

The space–time DG discretization is obtained by approximating u_h, v and $\mathcal{R}_h^S$ in each element with Legendre polynomials of degree p. After introducing these polynomial approximations into (1) and (2) the resulting system of algebraic equations in the space–time slab $\mathcal{E}^n$ can be represented as

$$L_h U_h^n = f_h, \tag{3}$$

with the discretization matrix L_h, the DG coefficients U_h^n, and righthand side f_h, which depends on the known DG coefficients U_h^{n-1} from the previous space–time slab. We refer to [27] for a detailed derivation and full error analysis of the space–time DG algorithm for the advection–diffusion equation.

In order to simplify notation we define in the remainder of this article the product and division of vectors element-wise. Hence for $a, b \in \mathbb{R}^d$ we have

$$ab := (a_1 b_1, \ldots, a_d b_d) \in \mathbb{R}^d \quad \text{and} \quad a/b := (a_1/b_1, \ldots, a_d/b_d) \in \mathbb{R}^d.$$

This notation will be particularly useful in the discrete Fourier analysis, but also to indicate the various (semi)-coarsened meshes.

3. Multigrid algorithm

3.1. hp-Multigrid as Smoother algorithm

In this section we present the *hp*-Multigrid as Smoother algorithm for the solution of higher order accurate discontinuous Galerkin discretizations. This algorithm combines a V-cycle *p*-multigrid algorithm with *h*-multigrid, which acts as smoother at each polynomial level. For a schematic overview, see Fig. 1. The *h*-multigrid smoother is provided by a semi-coarsening multigrid algorithm, see Fig. 2, in combination with a semi-implicit pseudo-time Runge–Kutta method, which will be discussed in Section 3.2. The Runge–Kutta method is semi-implicit in order to obtain also good multigrid performance in boundary layers.

Algorithm 1. *hp-MGS* Algorithm ($HP_{nh,p}$)

$v_{nh,p} := HP_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, n, p, \gamma_1, \gamma_2, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$
{
if polynomial level $p == 1$ **then**
  $v_{nh,p} := HU_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, n, p, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
  return
end if
// pre-smoothing with *h-MGS* algorithm
for $it = 1, \ldots, \gamma_1$ **do**
  $v_{nh,p} := HU_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, n, p, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
end for
// lower order polynomial solution
$r_{nh,p} := f_{nh,p} - L_{nh,p} v_{nh,p}$;
$f_{nh,p-1} := Q_{nh,p}^{p-1} r_{nh,p}$;
$v_{nh,p-1} := 0$;
$v_{nh,p-1} := HP_{nh,p}(L_{nh,p-1}, f_{nh,p-1}, v_{nh,p-1}, n, p-1, \gamma_1, \gamma_2, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
// lower order polynomial correction
$v_{nh,p} := v_{nh,p} + T_{nh,p-1}^{p} v_{nh,p-1}$;
// post-smoothing with *h-MGS* algorithm
for $it = 1, \ldots, \gamma_2$ **do**
  $v_{nh,p} := HU_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, n, p, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
end for
}

This new multigrid algorithm combines various techniques, viz. the *hp*-multigrid method, see e.g. [21,26], and the Multigrid as Smoother algorithm proposed in [22,34]. There are, however, a number of crucial differences. The *h*-multigrid algorithm is used at each polynomial level, instead of only at the coarsest polynomial level. This was motivated by the fact that after extensive multilevel computations only limited improvement in the multigrid convergence rate was obtained if at the coarsest polynomial level an exact solution was used instead of an *h*-multigrid algorithm. Hence, even an optimal *h*-multigrid at the coarsest polynomial level would only provide a limited improvement in multigrid performance. The second difference with *hp*-multigrid is that semi-coarsening multigrid is used as smoother in the *h*-multigrid. Finally, the coef-

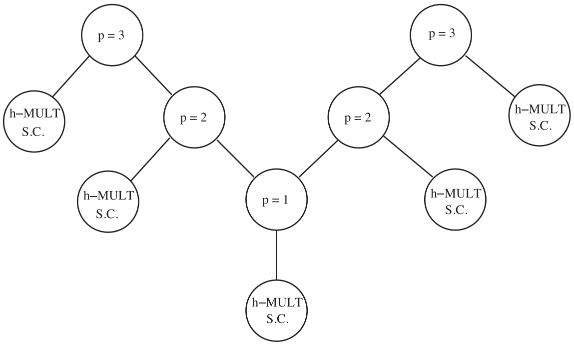


Fig. 1. *hp-MGS* algorithm combining *p*-multigrid and the *h*-Multigrid as Smoother algorithm at each polynomial level. The *h*-Multigrid as Smoother algorithm uses semi-coarsening in the local $\bar{x}_1$- and $\bar{x}_2$-directions and a semi-implicit Runge–Kutta method.

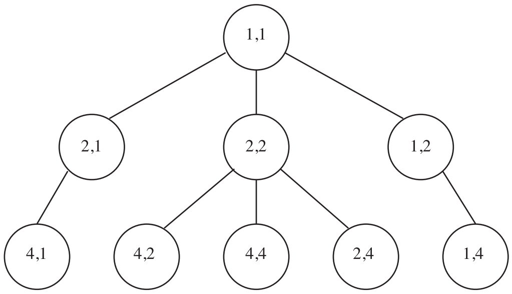


Fig. 2. *h*-Multigrid as Smoother algorithm used at each polynomial level *p* as smoother in the *hp-MGS* algorithm. The indices refer to grid coarsening. Mesh $(1,1)$ is the fine mesh and e.g. Mesh $(4,1)$ has mesh size $(4h_1, h_2)$.

ficients of the pseudo-time Runge–Kutta smoother in the semi-coarsening multigrid are optimized using multilevel analysis. This optimization process will be discussed in detail in Part II and makes it possible to account for the specific properties of the DG discretization and local flow conditions, such as the cell Reynolds number.

The *hp-MGS*-multigrid algorithm for the solution of the linear system (3) is described in Algorithms 1–3, with $n = (n_1, n_2) \in \mathbb{N}^2$ and $h = (h_1, h_2) \in (\mathbb{R}^+)^2$. We also use the notation $nh := (n_1h_1, n_2h_2)$. The computational meshes are indicated with $\mathcal{M}_{nh}$. The first part of the *hp-MGS* algorithm is defined recursively in Algorithm 1 and consists of the V-cycle *p*-multigrid algorithm $HP_{nh,p}$, with the *h-MGS* algorithm $HU_{nh,p}$, defined in Algorithm 2, as smoother. In Algorithm 1 the linear system is denoted as $L_{nh,p}$. The multigrid solution of the linear system is $v_{nh,p}$ and the known righthand side $f_{nh,p}$. The linear system originates from a numerical discretization with polynomial order p and mesh sizes h_1 and h_2 in the different local coordinate directions. The mesh coarsening is indicated by the integer $n = (n_1, n_2)$. The parameters $\gamma_1,\ \gamma_2,\ \nu_1,\ \nu_2,\ \mu_1,\ \mu_2$, and μ_3 are used to control the multigrid algorithm, such as the number of pre- and post-relations at each grid level and polynomial order. The $HP_{nh,p}$-multigrid algorithm uses the prolongation operators $T^p_{nh,p-1}$ and the restriction operators $Q^{p-1}_{nh,p}$. The prolongation operators $T^p_{nh,p-1}$ interpolate data from a discretization with polynomial order $p-1$ to a discretization with polynomial order p using an L_2 projection. The restriction operators $Q^{p-1}_{nh,p}$ project data from a discretization with polynomial order p to a discretization with polynomial order $p-1$. The restriction operators are the transpose of the prolongation operators, viz. $Q^{p-1}_{nh,p} = \left(T^p_{nh,p-1}\right)^T$.

In the $HU_{nh,p}$-multigrid algorithm, defined recursively in Algorithm 2, the semi-coarsening multigrid algorithm $HS^i_{nh,p}$, with $i = 1, 2$, is used as smoother in the local i-direction of each element. The restriction of the data from the mesh $\mathcal{M}_{nh}$ to the mesh $\mathcal{M}_{mh}$, with $m_1 \geqslant n_1$ and $m_2 \geqslant n_2$, is indicated by the restriction operators $R^{mh}_{nh,p}$. The prolongation of the data from the mesh $\mathcal{M}_{mh}$ to the mesh $\mathcal{M}_{nh}$ is given by the prolongation operators $P^{nh}_{mh,p}$. The prolongation operators $P^{nh}_{mh,p}$ are defined as the L_2 projection from the coarse grid element onto the fine grid elements which are a subset of the coarse grid element. The restriction operators are defined as $R^{mh}_{nh,p} = \left(P^{nh}_{mh,p}\right)^T / (n_1 n_2)$.

The semi-coarsening *h*-multigrid smoothers $HS^i_{nh,p}$, with $i = 1, 2$, are defined recursively in Algorithm 3. Here, i denotes the direction of the semi-coarsening, e.g. a coordinate direction or local face index in an unstructured mesh. The smoother in the direction i is indicated with $S^i_{nh,p}$ and will be discussed in detail in Section 3.2. At the coarsest levels in the semi-coarsened meshes we use μ_3 smoother iterations. Using the inverse of $L_{nh,p}$ at the coarsest semi-coarsened meshes is not practical since these meshes are still much larger than the coarsest uniformly coarsened mesh.

Various multigrid algorithms can be obtained by simplifying the *hp-MGS* algorithm given by Algorithms 1–3. The first simplification is obtained by replacing in the $HP_{nh,p}$ algorithm for polynomial levels $p > 1$ the *h-MGS*-multigrid smoother $HU_{nh,p}$ with the smoothers $S^2_{nh,p}S^1_{nh,p}$ in the pre-smoothing step and $S^1_{nh,p}S^2_{nh,p}$ in the post-smoothing step. We denote this algorithm as the *hp-MGS*(1) algorithm, since the *h-MGS* algorithm is now only used at the $p = 1$ level. The second simplification is to use only uniformly coarsened meshes in the *hp-MGS*(1) algorithm instead of semi-coarsened meshes. In addition, the semi-coarsening smoothers $HS^i_{nh,p}$ in the $HU_{nh,p}$ algorithm are replaced by the smoothers $S^i_{nh,p}$ for $i = 1, 2$. We denote this algorithm as *hp*-multigrid.

3.2. Pseudo-time multigrid smoothers

As multigrid smoothers we use in Algorithm 3 a pseudo-time integration method. In a pseudo-time integration method the linear system

$$L_{nh,p}v_{nh,p} = f_{nh,p} \tag{4}$$

is solved by adding a pseudo-time derivative. This results in a system of ordinary differential equations

$$\frac{\partial v^*_{nh,p}}{\partial \sigma} = -\frac{1}{\triangle t}\left(L_{nh,p}v^*_{nh,p} - f_{nh,p}\right), \tag{5}$$

Algorithm 2. *h-MGS* Algorithm $(HU_{nh,p})$

$v_{nh,p} := HU_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, n, p, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$
{
if coarsest uniformly coarsened mesh **then**
  $v_{nh,p} := L^{-1}_{nh,p}f_{nh,p}$;
  return
end if
// pre-smoothing using semi-coarsening multigrid
for $it = 1, \ldots, \nu_1$ **do**
  $v_{nh,p} := HS^1_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, 1, n, p, \mu_1, \mu_2, \mu_3)$;
  $v_{nh,p} := HS^2_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, 2, n, p, \mu_1, \mu_2, \mu_3)$;
end for
// coarse grid solution
$r_{nh,p} := f_{nh,p} - L_{nh,p}v_{nh,p}$;
$f_{2nh,p} := R^{2nh}_{nh,p}r_{nh,p}$;
$v_{2nh,p} := 0$;
$v_{2nh,p} := HU_{nh,p}(L_{2nh,p}, f_{2nh,p}, v_{2nh,p}, 2n, p, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
// coarse grid correction
$v_{nh,p} := v_{nh,p} + P^{nh}_{2nh,p}v_{2nh,p}$;
// post-smoothing using semi-coarsening multigrid
for $it = 1, \ldots, \nu_2$ **do**
  $v_{nh,p} := HS^2_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, 2, n, p, \mu_1, \mu_2, \mu_3)$;
  $v_{nh,p} := HS^1_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, 1, n, p, \mu_1, \mu_2, \mu_3)$;
end for
}

which is integrated to steady-state in pseudo-time. At steady state, $v_{nh,p} = v^*_{nh,p}$. Note, for nonlinear problems this system is obtained after linearization. The matrix $L_{nh,p}$ is then the Jacobian of the nonlinear algebraic system. The *hp-MGS* algorithm therefore naturally combines with a Newton multigrid method for nonlinear problems.

Since the goal of the pseudo-time integration is to reach steady state as efficiently as possible, time accuracy is not important. This allows the use of low order time integration methods, which can be optimized to improve multigrid convergence to steady state. In [15,29] optimized explicit pseudo-time Runge–Kutta methods are presented, which are used for the solution of second order accurate space–time DG discretizations of the compressible Euler and Navier–Stokes equations [14,29]. An important benefit of these explicit pseudo-time smoothers is that they can be directly applied to nonlinear problems without linearization. For higher order accurate DG discretizations, in particular for problems with thin boundary layers, the performance of these smoothers is, however, insufficient. This motivated the development of a semi-implicit Runge–Kutta pseudo-time integration method, which will be discussed in the next section.

Algorithm 3. Semi-coarsening multigrid algorithm $\left(HS^{i}_{nh,p}\right)$

$v_{nh,p} := HS^{i}_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, i, n, p, \mu_1, \mu_2, \mu_3)$
{
if ($i == 1$ **and** coarsest mesh in local i_1-direction) or ($i == 2$ **and** coarsest mesh in local i_2-direction) **then**
  for $it = 1, \dots, \mu_3$ **do**
    $v_{nh,p} := S^{i}_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p})$;
  end for
  return
end if
// pre-smoothing
for $it = 1, \dots, \mu_1$ **do**
  $v_{nh,p} := S^{i}_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p})$;
end for
// coarse grid solution on semi-coarsened meshes
$r_{nh,p} := f_{nh,p} - L_{nh,p} v_{nh,p}$;
if ($i == 1$) **then**
  // semi-coarsening in local i_1-direction
  $f_{(2n_1,n_2)h,p} := R^{(2n_1,n_2)h}_{nh,p} r_{nh,p}$;
  $v_{(2n_1,n_2)h,p} := 0$;
  $v_{(2n_1,n_2)h,p} := HS^{1}_{nh,p}(L_{(2n_1,n_2)h,p}, f_{(2n_1,n_2)h,p}, v_{(2n_1,n_2)h,p}, i, (2n_1, n_2), p, \mu_1, \mu_2, \mu_3)$;
  $v_{nh,p} := v_{nh,p} + P^{nh}_{(2n_1,n_2)h,p} v_{(2n_1,n_2)h,p}$;
else if ($i == 2$) **then**
  // semi-coarsening in local i_2-direction
  $f_{(n_1,2n_2)h,p} := R^{(n_1,2n_2)h}_{nh,p} r_{nh,p}$;
  $v_{(n_1,2n_2)h,p} := 0$;
  $v_{(n_1,2n_2)h,p} := HS^{2}_{nh,p}(L_{(n_1,2n_2)h,p}, f_{(n_1,2n_2)h,p}, v_{(n_1,2n_2)h,p}, i, (n_1, 2n_2), p, \mu_1, \mu_2, \mu_3)$;
  $v_{nh,p} := v_{nh,p} + P^{nh}_{(n_1,2n_2)h,p} v_{(n_1,2n_2)h,p}$;
end if
// post-smoothing
for $it = 1, \dots, \mu_2$ **do**
  $v_{nh,p} := S^{i}_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p})$;
end for
}

3.2.1. Semi-implicit Runge–Kutta smoother

The system of ordinary differential Eq. (5) will be solved using a five-stage semi-implicit Runge–Kutta method. In the semi-implicit Runge–Kutta method we use the fact that the *hp-MGS* algorithm uses semi-coarsening in the local i_1- and i_2-directions of each element. This makes it a natural choice to use a Runge–Kutta pseudo-time integrator which is implicit in the local directions used for the semi-coarsening. Also, the space–time DG discretization uses, next to data on the element itself, only data from elements connected to each of its faces. This results in a linear system with a block matrix structure. It is therefore straightforward to use a Runge–Kutta pseudo-time integrator which is alternating implicit in the local i_1 and i_2-direction. The linear system then consists of uncoupled systems of block tridiagonal matrices, which can be efficiently solved with a direct method. The semi-implicit pseudo-time integration method then can efficiently deal with highly stretched meshes in boundary layers. For this purpose we split the matrix $L_{nh,p}$, when sweeping in the i_1-direction, as

$$L_{nh,p} = L^{i_{11}}_{nh,p} + L^{i_{12}}_{nh,p}$$

and for sweeps in the i_2-direction as

$$L_{nh,p} = L^{i_{21}}_{nh,p} + L^{i_{22}}_{nh,p}.$$

The matrices $L^{i_{11}}_{nh,p}$ and $L^{i_{21}}_{nh,p}$ contain the contribution from the element itself and the elements connected to each face in the i_1-direction, respectively, i_2-direction, which are treated implicitly. The matrices $L^{i_{12}}_{nh,p}$ and $L^{i_{22}}_{nh,p}$ contain the contribution from each face in the i_2-direction, respectively, i_1-direction, which are treated explicitly. Since the DG discretization only uses information from nearest neighboring elements this provides a very natural way to define the lines along which the discretization is implicit. The semi-implicit Runge–Kutta method for sweeps in the i_1-direction then can be defined for the $l+1$ pseudo-time step as

$$\begin{aligned} v_0 &= v^l_{nh,p} \\ v_k &= \left(I_{nh,p} + \beta_k \lambda_\sigma L^{i_{11}}_{nh,p}\right)^{-1} \left(v_0 - \lambda_\sigma \sum_{j=0}^{k-1} \alpha_{kj} (L^{i_{12}}_{nh,p} v_j - f_{nh,p})\right), \quad k = 1,\ldots,5, \\ v^{l+1}_{nh,p} &= S^i_{nh,p} v^l_{nh,p} = v_5, \end{aligned} \tag{6}$$

with a similar relation for sweeps in the i_2-direction, where i_{11} is replaced by i_{21} and i_{12} with i_{22}. Here, α_{kj} are the Runge–Kutta coefficients, $\beta_k = \sum_{j=0}^{k-1} \alpha_{kj}$ for $k = 1,\ldots 5, \lambda_\sigma = \triangle\sigma/\triangle t$, with $\triangle\sigma$ the pseudo-time step. At steady state of the σ-pseudo-time integration we obtain the solution of the linear system (4). The coefficients β_k ensure that the semi-implicit Runge–Kutta operator is the identity operator if $v^l_{nh,p}$ is the exact steady state solution of (5). Without this condition the pseudo-time integration method would not converge to a steady state. The only requirement we impose on the Runge–Kutta coefficients α_{kj} is that the algorithm is first order accurate in pseudo-time, which implies the consistency condition

$$\sum_{j=0}^{4} \alpha_{5j} = 1.$$

For each polynomial level all other Runge–Kutta coefficients can be optimized to improve the pseudo-time convergence in combination with the *hp-MGS* algorithm. For the computation of the multigrid error transformation operator we define the semi-implicit Runge–Kutta operator $Q^1_{nh,p}$ recursively for sweeps in the i_1-direction as

$$\begin{aligned} Q_0 &= I_{nh,p} \\ Q_k &= \left(I_{nh,p} + \beta_k \lambda_\sigma L^{i_{11}}_{nh,p}\right)^{-1} \left(I_{nh,p} - \lambda_\sigma \sum_{j=0}^{k-1} \alpha_{kj} L^{i_{12}}_{nh,p} Q_j\right), \quad k = 1,\ldots,5, \\ Q^1_{nh,p} &= Q_5, \end{aligned} \tag{7}$$

with a similar expression for $Q^2_{nh,p}$ in the i_2-direction, only with i_{11} and i_{12} replaced by, respectively, i_{21} and i_{22}.

4. *hp-MGS* error transformation operator

The performance of the *hp-MGS* algorithm defined in Algorithms 1–3 is determined by the multigrid error transformation operator. This operator determines the change in the error after one application of the full *hp-MGS* algorithm. We assume that the linear system (4) is obtained from a space–time DG discretization using polynomial basis functions of order p. The initial error in the solution of the algebraic system on the grid $\mathcal{M}_{nh}$ is defined as

$$e^0_{nh,p} = U_{nh,p} - v^0_{nh,p}.$$

Here, $U_{nh,p}$ is the exact solution of the algebraic system

$$L_{nh,p} U_{nh,p} = f_{nh,p}$$

and $v^0_{nh,p}$ the initial guess used in the multigrid algorithm. Similarly, the error after one application of the multigrid algorithm is defined as

$$e^1_{nh,p} = U_{nh,p} - v^1_{nh,p},$$

with $v^1_{nh,p} = HP_{nh,p} v^0_{nh,p}$. The operator $HP_{nh,p}$ denotes the action of the *hp-MGS* algorithm defined in Algorithm 1. The initial and multigrid error are related through the *hp-MGS* error transformation operator $M_{nh,p}$, viz.

$$e^1_{nh,p} = M_{nh,p} e^0_{nh,p}.$$

The detailed formulation of the error transformation operator of the *hp-MGS* algorithm can now be obtained by computing the error transformation operators of Algorithms 1–3, defined in Section 3.1, and the pseudo-time smoother, defined in Section 3.2. For more details on the computation of the error transformation operator, see e.g. [10,28].

The *hp-MGS* error transformation operator $M_{nh,p}$ for the $HP_{nh,p}$ multigrid algorithm can be defined recursively as

$$\begin{aligned} M_{nh,p} &= (HU_{nh,p})^{\gamma_2} \left(I_{nh,p} - T^p_{nh,p-1} (I_{nh,p-1} - M_{nh,p-1}) (L_{nh,p-1})^{-1} Q^{p-1}_{nh,p} L_{nh,p}\right) (HU_{nh,p})^{\gamma_1} \text{ if } p > 1, \\ &= HU_{nh,1} \text{ if } p = 1. \end{aligned} \tag{8}$$

In the *h-MGS* step we first compute the error reduction using the $HU_{nh,p}$ algorithm, defined in Algorithm 2. The *h-MGS* error transformation operator $HU_{nh,p}$ is equal to

$$\begin{aligned} HU_{nh,p} &= \left(HS^1_{nh,p}HS^2_{nh,p}\right)^{\nu_2}\left(I_{nh,p} - P^{nh}_{2nh,p}(I_{2nh,p} - HU_{2nh,p})(L_{2nh,p})^{-1}R^{2nh}_{nh,p}L_{nh,p}\right)\left(HS^2_{nh,p}HS^1_{nh,p}\right)^{\nu_1}, \text{ if } n < m, \\ &= 0, \text{ if } n = m. \end{aligned} \tag{9}$$

The $HU_{nh,p}$ error transformation operator (9) can also be used to obtain the semi-coarsening multigrid error transformation operators $HS^1_{nh,p}$ and $HS^2_{nh,p}$, defined in Algorithm 3, which are equal to

$$\begin{aligned} HS^1_{nh,p} &= \left(S^1_{nh,p}\right)^{\mu_2}\left(I_{nh,p} - P^{nh}_{(2n_1,n_2)h,p}(I_{(2n_1,n_2)h,p} - HS^1_{(2n_1,n_2)h,p})(L_{(2n_1,n_2)h,p})^{-1}R^{(2n_1,n_2)h}_{nh,p}L_{nh,p}\right)\left(S^1_{nh,p}\right)^{\mu_1}, \text{ if } n < m, \\ &= I_{nh,p} - \left(S^1_{nh,p}\right)^{\mu_3}, \text{ if } n = m, \\ HS^2_{nh,p} &= \left(S^2_{nh,p}\right)^{\mu_2}\left(I_{nh,p} - P^{nh}_{(n_1,2n_2)h,p}(I_{(n_1,2n_2)h,p} - HS^2_{(n_1,2n_2)h,p})(L_{(n_1,2n_2)h,p})^{-1}R^{(n_1,2n_2)h}_{nh,p}L_{nh,p}\right)\left(S^2_{nh,p}\right)^{\mu_1}, \text{ if } n < m, \\ &= I_{nh,p} - \left(S^2_{nh,p}\right)^{\mu_3}, \text{ if } n = m. \end{aligned}$$

Next, we discuss the error transformation operator of the semi-implicit Runge–Kutta pseudo-time smoother, defined in Section 3.2. The error after the lst and l + 1th semi-implicit Runge–Kutta pseudo-time integration step is

$$\begin{aligned} \tilde{e}^0_{nh,p} &= v_{nh,p} - v^l_{nh,p} \\ \tilde{e}^1_{nh,p} &= v_{nh,p} - v^{l+1}_{nh,p} \end{aligned}$$

and the error in each Runge–Kutta stage as

$$\bar{e}_k = v_{nh,p} - v_k,$$

with $\bar{e}_0 = \tilde{e}^0_{nh,p}$. The error after one semi-implicit Runge–Kutta step can now be defined recursively as

$$\begin{aligned} \bar{e}_0 &= \tilde{e}^0_{nh,p} \\ \bar{e}_k &= \left(I_{nh,p} + \beta_k\lambda_\sigma L^{i_{11}}_{nh,p}\right)^{-1}\left(\bar{e}_0 - \lambda_\sigma\sum_{j=0}^{k-1}\alpha_{kj}L^{i_{12}}_{nh,p}\bar{e}_j\right), \quad k = 1,\ldots,5, \\ \bar{e}^1_{nh,p} &= S^1_{nh,p}\bar{e}^0_{nh,p} = Q^1_{nh,p}\bar{e}^0_{nh,p} = \bar{e}_5. \end{aligned}$$

A similar expression is obtained for $S^2_{nh,p}$, when the Runge–Kutta method is implicit in the i_2-direction. Only i_{11} and i_{12} are replaced by, respectively, i_{21} and i_{22}. Combining all contributions gives the *hp-MGS* error transformation operator $M_{nh,p}$.

5. Fourier analysis of *hp-MGS* Algorithm

The analysis of the *hp-MGS* error transformation operator can be performed using discrete Fourier analysis. This allows the efficient computation of the operator norm and spectral radius of the multigrid error transformation operator, which will be used in Part II to optimize the pseudo-time Runge–Kutta smoother. The analysis of the *hp-MGS* algorithm will consider three polynomial levels and three semi-coarsened and uniformly coarsened mesh levels. The large number of multigrid levels in combination with the different types of mesh coarsening make the multilevel analysis intricate. We start in Sections 5.1 and 5.2 with some important definitions and discuss the aliasing of modes, which depends on the type of mesh coarsening. Next, we describe in Section 5.3 the Fourier symbols of the discrete operators, viz. the spatial discretization operators and smoothers, and the restriction and prolongation operators for all types of meshes considered in this study. The Fourier symbols of the discrete operators will then be used in Section 5.4 to give a unified description of three-level analysis, suitable for both uniformly and semi-coarsened meshes. Finally, in Section 5.5 the different parts are combined into the Fourier symbol of the *hp-MGS* error transformation operator. More details on the discrete Fourier multilevel analysis of the *hp-MGS* error transformation operator can be found in [31]. General information on discrete Fourier analysis of multigrid algorithms is available in [4,10,11,28,35–37].

5.1. Definitions

In this section we will introduce some definitions which will be used throughout the multilevel analysis. Assume a finite mesh $G^N_{nh} \subset \mathbb{R}^2$, with $n, N \in \mathbb{N}^2$ and $h \in (\mathbb{R}^+)^2$, which is defined as

$$G^N_{nh} := \{\bar{x} = (\bar{x}_1, \bar{x}_2) = (k_1n_1h_1, k_2n_2h_2) \mid k \in \mathcal{G}^N_n\},$$

with index set $\mathcal{G}^N_n$ given by

$$\mathcal{G}^N_n = \{k \in \mathbb{Z}^2 \mid -N_i/n_i \leqslant k_i \leqslant (N_i/n_i) - 1, N_i/n_i \in \mathbb{N}, \ i = 1, 2\}. \tag{10}$$

We also use the set $\mathcal{G}_n^N$ to enumerate the elements used in the space–time discretization. On G_{nh}^N we define for $v_{nh}, w_{nh} : G_{nh}^N \to \mathbb{C}$ the scaled Euclidian inner product

$$(v_{nh}, w_{nh})_{G_{nh}^N} := \frac{n_1 n_2}{4N_1 N_2} \sum_{\bar{x} \in G_{nh}^N} v_{nh}(\bar{x}) \overline{w_{nh}(\bar{x})} \tag{11}$$

and norm

$$\| v_{nh} \|_{G_{nh}^N} := (v_{nh}, v_{nh})_{G_{nh}^N}^{\frac{1}{2}}.$$

Here an overbar denotes the complex conjugate. We will also consider an infinite mesh G_{nh}, which is defined as

$$G_{nh} := \left\{ \bar{x} = (\bar{x}_1, \bar{x}_2) = (k_1 n_1 h_1, k_2 n_2 h_2) \mid k \in \mathbb{Z}^2 \right\}.$$

Similarly, on G_{nh} we define for $v_{nh}, w_{nh} : G_{nh} \to \mathbb{C}$ the scaled Euclidian inner product as

$$(v_{nh}, w_{nh})_{G_{nh}} := \lim_{N \to \infty} \frac{n_1 n_2}{4N^2} \sum_{\bar{x} \in G_{nh}^N} v_{nh}(\bar{x}) \overline{w_{nh}(\bar{x})}, \tag{12}$$

with associated norm $\| v_{nh} \|_{G_{nh}}$. In $\mathbb{R}^2$ a uniform mesh with mesh sizes (h_1, h_2) can now be represented as $G_h = G_{(h_1,h_2)}$ and a uniformly coarsened mesh as $G_{2h} = G_{(2h_1,2h_2)}$. A mesh with semi-coarsening in the $\bar{x}_1$-, respectively, $\bar{x}_2$-direction is represented as $G_{(2h_1,h_2)}$ and $G_{(h_1,2h_2)}$. Based on the mesh points it is straightforward to construct the finite element mesh consisting of rectangular elements.

The linear system (3) on the mesh G_{nh} using periodic boundary conditions and polynomials of order p in the space–time DG discretization is described in stencil notation as

$$L_{nh,p} v_{nh,p}(\bar{x}) = \sum_{k \in J_n} l_{k,nh,p} v_{nh,p}(\bar{x} + knh), \quad \bar{x} \in G_{nh}, \tag{13}$$

where the stencil coefficients $l_{k,nh,p}$ are $m_p \times m_p$ matrices, with $m_p \geqslant 1$ depending on the polynomial order p used in the space–time DG discretization. Note, in matrix notation the linear system can be represented by a block Toeplitz matrix. The space–time DG coefficients are denoted $v_{nh,p}$ and are associated in the Fourier analysis with the center of each element. The finite index sets $J_n \subset \mathbb{Z}^2$ describe the space–time DG stencil. In two dimensions the space–time DG discretization has a 5-point stencil. The stencil of $L_{nh,p}$ is then given by

$$[L_{nh,p}] = \begin{pmatrix} 0 & l_{(-1,0),nh,p} & 0 \\ l_{(0,-1),nh,p} & l_{(0,0),nh,p} & l_{(0,1),nh,p} \\ 0 & l_{(1,0),nh,p} & 0 \end{pmatrix}.$$

On the infinite mesh $G_{nh} \subset \mathbb{R}^2$, we define for $\bar{x} \in G_{nh}$ the continuous Fourier modes with frequency $\theta = (\theta_1, \theta_2) \in \Pi_n$, with $\Pi_n = [-\frac{\pi}{n_1}, \frac{\pi}{n_1}) \times [-\frac{\pi}{n_2}, \frac{\pi}{n_2})$, as

$$\phi_{nh}(n\theta, \bar{x}) := e^{\imath n\theta \cdot \bar{x}/(nh)}, \tag{14}$$

where $n\theta \cdot \bar{x}/(nh) = \theta_1 \bar{x}_1/h_1 + \theta_2 \bar{x}_2/h_2$, $h \in (\mathbb{R}^+)^2$ and $\imath = \sqrt{-1}$. Note, the Fourier modes are orthonormal with respect to the scaled Euclidian inner product on G_{nh}.

We define the space of bounded grid functions on the infinite mesh G_{nh} as

$$\mathcal{F}(G_{nh}) := \left\{ v_{nh} \mid v_{nh} : G_{nh} \to \mathbb{C} \text{ with } \| v_{nh} \|_{G_{nh}} < \infty \right\}.$$

For each $v_{nh} \in \mathcal{F}(G_{nh})$, there exists a Fourier transform, which is defined as

$$\widehat{v_{nh}}(n\theta) = \frac{n_1 n_2}{4\pi^2} \sum_{\bar{x} \in G_{nh}} v_{nh}(\bar{x}) e^{-\imath n\theta \cdot \bar{x}/(nh)}, \quad \theta \in \Pi_n. \tag{15}$$

The inverse Fourier transform is given by

$$v_{nh}(\bar{x}) = \int_{\theta \in \Pi_n} \widehat{v_{nh}}(n\theta) e^{\imath n\theta \cdot \bar{x}/(nh)} d\theta, \quad \bar{x} \in G_{nh}. \tag{16}$$

Hence v_{nh} can be written as a linear combination of Fourier components.

Due to aliasing, Fourier components with $|\hat{\theta}| := \max\{n_1 |\theta_1|, n_2 |\theta_2|\} \geqslant \pi$ are not visible on G_{nh}. These modes therefore coincide with $e^{\imath n\theta \cdot \bar{x}/(nh)}$, where $\theta = \hat{\theta} (\text{mod} 2\pi/n)$. Hence, the Fourier space

$$\mathcal{F}_n(G_{nh}) := \text{span}\{\phi_h(\theta, \bar{x}) \mid \theta \in \Pi_n, \bar{x} \in G_{nh}\}$$

contains any bounded infinite grid function on G_{nh}.

On a finite domain with mesh G^N_{nh}, where at the domain boundaries periodic boundary conditions are imposed, only a finite number of frequencies can be represented. Hence, for every $v_{nh} \in \mathcal{F}_n(G^N_{nh})$ the discrete Fourier transform is defined as

$$\widehat{v_{nh}}(n\theta_k) = \frac{n_1 n_2}{4N_1N_2} \sum_{\bar{x}\in G^N_{nh}} v_{nh}(\bar{x}) e^{-\imath n\theta_k\cdot\bar{x}/(nh)},$$

with $\theta_k = (\theta_{k_1}, \theta_{k_2}),\ \theta_k = \pi k/N,\ k \in \mathcal{G}^N_n,\ N \in \mathbb{N}^d$. The inverse discrete Fourier transform is given by

$$v_{nh}(\bar{x}) = \sum_{k\in\mathcal{G}^N_n} \widehat{v_{nh}}(n\theta_k) e^{\imath n\theta_k\cdot\bar{x}/(nh)}, \quad \bar{x} \in G^N_{nh}.$$

The results of the discrete Fourier analysis on the infinite mesh G_{nh} and the finite mesh G^N_{nh} are equal for a periodic field at the frequencies $\theta = \theta_k$, with $\theta_k = \pi k/N,\ k \in \mathcal{G}^N_n,\ N \in \mathbb{N}^2$. This equivalence will be used to find approximate results for the discrete Fourier analysis on the infinite mesh G_{nh}, which generally results in eigenvalue problems which can not be solved analytically.

5.2. Aliasing of Fourier modes

In three-level analysis with uniform mesh coarsening 16 modes on the fine mesh $G_{(h_1,h_2)}$ alias to four independent modes on the mesh $G_{(2h_1,2h_2)}$ and to one mode on the coarsest mesh $G_{(4h_1,4h_2)}$, see Fig. 3. We therefore introduce the Fourier harmonics $\mathcal{F}^3_h(\theta)$, with $\theta \in \Pi_{(4,4)}$, as

$$\mathcal{F}^3_h(\theta) := \text{span}\left\{\phi_h(\theta^\alpha_\beta, \bar{x}) \mid \alpha \in \alpha_2, \beta \in \beta_2\right\},$$

with

$$\begin{aligned}
&\theta = \theta^{00}_{00} \in \Pi_{(4,4)} := [-\pi/4, \pi/4)^2,\\
&\theta^{00}_\beta = \theta^{00}_{00} - (\bar{\beta}_1 \text{sign}(\theta_1), \bar{\beta}_2 \text{sign}(\theta_2))\pi,\\
&\theta^\alpha_\beta := \theta^{00}_\beta - (\bar{\alpha}_1 \text{sign}((\theta^{00}_\beta)_1), \bar{\alpha}_2 \text{sign}((\theta^{00}_\beta)_2))\pi,\\
&\alpha_2 = \{(\bar{\alpha}_1, \bar{\alpha}_2) \mid \bar{\alpha}_i \in \{0, 1\},\ i = 1, 2\},\\
&\beta_2 = \left\{(\bar{\beta}_1, \bar{\beta}_2) \mid \bar{\beta}_i \in \left\{0, \frac{1}{2}\right\},\ i = 1, 2\right\}.
\end{aligned} \tag{17}$$

Next to uniform coarsening, the *hp-MGS* algorithm also uses semi-coarsening multigrid. In this case the grid is coarsened in only one direction, which implies that four modes on the fine mesh alias to two modes on the medium mesh, and to one mode on the coarsest mesh, see Figs. 4 and 5.

The aliasing relations for the Fourier modes on the different coarse meshes can be straightforwardly computed using the representation of the modes θ^α_β given by (17). First, assume the following mesh coarsenings $G_h \to G_{nh}$, with

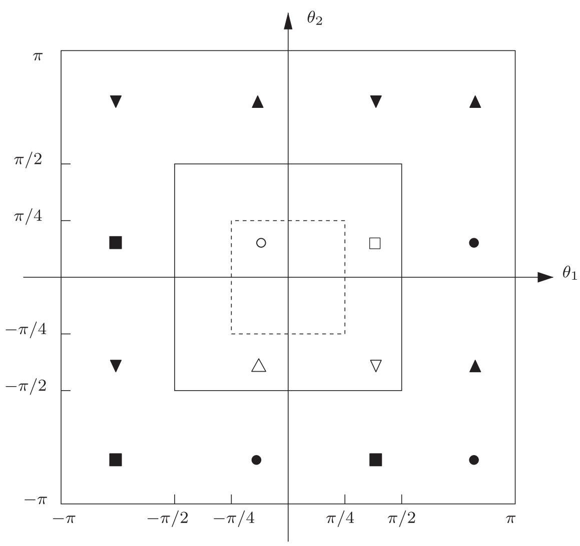


Fig. 3. Aliasing of Fourier modes for uniform-coarsening. Modes with a black symbol alias on the mesh G_{2h} to the mode with equivalent open symbol in the domain $[-\pi/2, \pi/2)^2$. Modes in the domain $[-\pi/2, \pi/2)^2 \setminus [-\pi/4, \pi/4)^2$ alias on the mesh G_{4h} to the mode in $[-\pi/4, \pi/4)^2$.

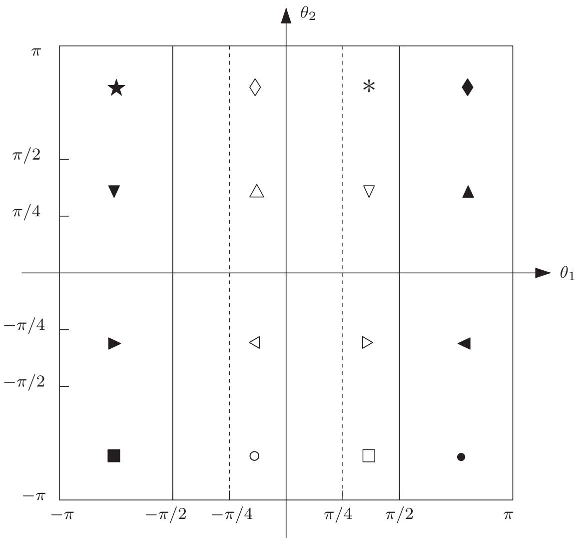


Fig. 4. Aliasing of Fourier modes for semi-coarsening in the $\bar{x}_1$-direction. Modes with a black symbol alias on the mesh $G_{(2h_1,h_2)}$ to the mode with an equivalent open symbol in the domain $[-\pi/2, \pi/2) \times [-\pi, \pi)$. Modes in the domain $\theta \in ([-\pi/2, -\pi/4) \cup [\pi/4, \pi/2)) \times [-\pi, \pi)$ alias on the mesh $G_{(4h_1,h_2)}$ to the mode in $[-\pi/4, \pi/4) \times [-\pi, \pi)$ with the same value of θ_2.

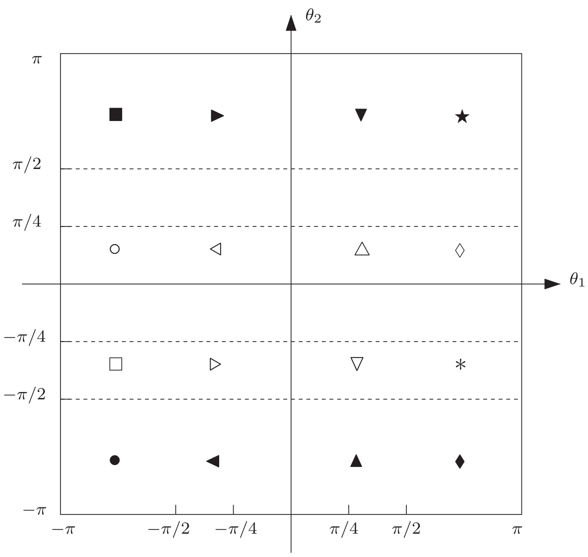


Fig. 5. Aliasing of Fourier modes for semi-coarsening in the $\bar{x}_2$-direction. Modes with a black symbol alias on the mesh $G_{(h_1,2h_2)}$ to the mode with an equivalent open symbol in the domain $[-\pi, \pi) \times [-\pi/2, \pi/2)$. Modes in the domain $\theta \in [-\pi, \pi) \times ([-\pi/2, -\pi/4) \cup [\pi/4, \pi/2))$ alias on the mesh $G_{(h_1,4h_2)}$ to the mode in $[-\pi, \pi) \times [-\pi/4, \pi/4)$ with the same value of θ_1.

$n \in \{(2,2),(2,1),(1,2)\}$, which includes both uniform and semi-coarsening. For $\bar{x} \in G_{nh}$ Fourier modes with frequency $\theta^{\alpha}_{\beta} \in \Pi_{(1,1)}$, with $\alpha \in \alpha_2$, $\beta \in \beta_2$, alias on the mesh G_{nh} to modes with frequency $\theta^{\alpha'}_{\beta} \in \Pi_n$ with

$$\phi_h(\theta^{\alpha}_{\beta}, \bar{x}) = \phi_h(\theta^{\alpha'}_{\beta}, \bar{x}) = \phi_{nh}(n\theta^{\alpha'}_{\beta}, \bar{x}), \quad \theta^{\alpha'}_{\beta} \in \Pi_n, \quad \bar{x} \in G_{nh}$$

and

$$\alpha' = \begin{cases} (0,0) & \text{if } n = (2,2), \\ (0,\bar{\alpha}_2) & \text{if } n = (2,1), \\ (\bar{\alpha}_1, 0) & \text{if } n = (1,2). \end{cases} \tag{18}$$

Analogously, for the mesh coarsening $G_{nh} \to G_{mh}$, with $m \in \{(4,4),(4,1),(1,4)\}$, modes with frequency $\theta_{\beta}^{\alpha'} \in \Pi_n$ alias on the mesh G_{mh} to modes with frequency $\theta_{\beta'}^{\alpha'} \in \Pi_m$ as

$$\phi_{nh}(n\theta_{\beta}^{\alpha'},\bar{x}) = \phi_h(\theta_{\beta'}^{\alpha'},\bar{x}) = \phi_{mh}(m\theta_{\beta'}^{\alpha'},\bar{x}), \quad \theta_{\beta'}^{\alpha'} \in \Pi_m, \quad \bar{x} \in G_{mh},$$

with α' and β' given by

$$\begin{aligned} &\alpha' = (0,0), \quad &&\beta' = (0,0), \text{ if } m = (4,4),\\ &\alpha' = (0,\bar{\alpha}_2), \quad &&\beta' = (0,\bar{\beta}_2) \text{ if } m = (4,1),\\ &\alpha' = (\bar{\alpha}_1,0), \quad &&\beta' = (\bar{\beta}_1,0) \text{ if } m = (1,4). \end{aligned}$$

In order to unify the analysis of uniform and semi-coarsening multigrid we use the sixteen modes θ_{β}^{α} defined in (17) for uniform coarsening also in the semi-coarsening analysis. These modes are, however, subdivided into four independent groups. On the coarser meshes there is no aliasing between modes in different groups, only between modes in the same group.

For the three-level Fourier analysis of semi-coarsening in the $\bar{x}_1$-direction we subdivide the Fourier harmonics with frequencies $\theta_{\beta}^{\alpha},\ \alpha \in \alpha_2,\ \beta \in \beta_2$, on the mesh $G_{(h_1,h_2)}$ into the groups

$$\begin{aligned} \alpha_{(2,1)}^1 &= \{(0,0),(1,0)\} &&\to \quad \gamma_{(2,1)}^1 = (0,0),\\ \alpha_{(2,1)}^2 &= \{(1,1),(0,1)\} &&\to \quad \gamma_{(2,1)}^2 = (0,1),\\ \beta_{(2,1)}^1 &= \left\{(0,0),\left(\frac{1}{2},0\right)\right\} &&\to \quad \delta_{(2,1)}^1 = (0,0),\\ \beta_{(2,1)}^2 &= \left\{\left(\frac{1}{2},\frac{1}{2}\right),\left(0,\frac{1}{2}\right)\right\} &&\to \quad \delta_{(2,1)}^2 = \left(0,\frac{1}{2}\right), \end{aligned}$$

where the index of the mode to which each group of modes aliases on the next coarser mesh level is indicated with an arrow, see also Fig. 4. For example, the modes on the mesh G_h with frequency $\theta_{\beta}^{\alpha}, \alpha \in \alpha_{(2,1)}^1$, alias for each $\beta \in \beta_{(2,1)}^1$ to the frequency $\theta_{\beta}^{\gamma_{(2,1)}^1}$ on the mesh $G_{(2h_1,h_2)}$. Similarly, on the mesh $G_{(2h_1,h_2)}$ the modes $\theta_{\beta}^{\gamma_{(2,1)}^1}, \beta \in \beta_{(2,1)}^1$, alias to the frequency $\theta_{\delta_{(2,1)}^1}^{\gamma_{(2,1)}^1}$ on the mesh $G_{(4h_1,h_2)}$.

Next, for three-level Fourier analysis of semi-coarsening in the $\bar{x}_2$-direction we define the groups

$$\begin{aligned} \alpha_{(1,2)}^1 &= \{(0,0),(0,1)\} &&\to \quad \gamma_{(1,2)}^1 = (0,0),\\ \alpha_{(1,2)}^2 &= \{(1,1),(1,0)\} &&\to \quad \gamma_{(1,2)}^2 = (1,0),\\ \beta_{(1,2)}^1 &= \left\{(0,0),\left(0,\frac{1}{2}\right)\right\} &&\to \quad \delta_{(1,2)}^1 = (0,0),\\ \beta_{(1,2)}^2 &= \left\{\left(\frac{1}{2},\frac{1}{2}\right),\left(\frac{1}{2},0\right)\right\} &&\to \quad \delta_{(1,2)}^2 = \left(\frac{1}{2},0\right), \end{aligned}$$

see Fig. 5. Finally, for uniform mesh coarsening the modes in the three-level Fourier analysis are ordered as

$$\begin{aligned} \alpha_{(2,2)}^1 &= \{(0,0),(1,1),(1,0),(0,1)\} &&\to \quad \gamma_{(2,2)}^1 = (0,0),\\ \beta_{(2,2)}^1 &= \left\{(0,0),\left(\frac{1}{2},\frac{1}{2}\right),\left(\frac{1}{2},0\right),\left(0,\frac{1}{2}\right)\right\} &&\to \quad \delta_{(2,2)}^1 = (0,0), \end{aligned}$$

see Fig. 3. In principle the ordering of modes in the different groups can be changed, but it is important that the same ordering is used in all steps of the multilevel analysis.

5.3. Fourier symbols of discrete operators

In this section we will summarize the Fourier symbols of the multigrid operators, namely the fine and coarse grid operators, the smoothing operators, and the restriction and prolongation operators. We will present the Fourier symbols in a unified way, suitable for both uniform and semi-coarsening multigrid.

5.3.1. Discrete Fourier transform of space–time DG operator

On the mesh G_h we can express (13) in terms of its discrete Fourier transform through the relation

$$(L_{h,p}v_{h,p})(\bar{x}) = \sum_{\alpha\in\alpha_{(2,2)}^1}\sum_{\beta\in\beta_{(2,2)}^1}\int_{\theta\in\Pi_{(4,4)}} \widehat{L_{h,p}}\left(\theta_{\beta}^{\alpha}\right)\widehat{v_{h,p}}\left(\theta_{\beta}^{\alpha}\right)e^{\imath\theta_{\beta}^{\alpha}\cdot\bar{x}/h}d\theta, \tag{19}$$

with $\theta_{\beta}^{\alpha} = \theta_{\beta}^{\alpha}(\theta)$ given by (17). The Fourier symbol $\widehat{L_{h,p}}\left(\theta_{\beta}^{\alpha}\right)$ is defined as

$$\widehat{L_{h,p}}\left(\theta_{\beta}^{\alpha}\right) = \sum_{k\in J_{L_{h,p}}} l_{k,h,p}e^{\imath\theta_{\beta}^{\alpha}\cdot k} \in \mathbb{C}^{m_p\times m_p}.$$

On the mesh G_{nh}, with $n \in \{(2,2),(2,1),(1,2)\}$, we obtain the relation

$$(L_{nh,p}v_{nh,p})(\bar{x}) = \sum_{i\in s_n}\sum_{j\in s_n}\sum_{\beta\in\beta_n^j}\int_{\theta\in\Pi_{(4,4)}} \widehat{L_{nh,p}}\left(n\theta_\beta^{\gamma_n^i}\right)\widehat{v_{nh,p}}\left(n\theta_\beta^{\gamma_n^i}\right)e^{\imath n\theta_\beta^\alpha\cdot\bar{x}/(nh)}d\theta,$$

where the set $s_n = \{1,2\}$ if $n = (2,1),\ (1,2),\ (4,1),\ (1,4)$ and $s_n = \{1\}$ if $n = (2,2),\ (4,4)$. The Fourier symbol $\widehat{L_{nh,p}}\left(n\theta_\beta^{\gamma_n^i}\right)$ is defined as

$$\widehat{L_{nh,p}}\left(n\theta_\beta^{\gamma_n^i}\right) = \sum_{k\in J_{L_{nh,p}}} l_{k,nh,p}e^{\imath n\theta_\beta^{\gamma_n^i}\cdot k} \in \mathbb{C}^{m_p\times m_p}.$$

Finally, on the mesh G_{mh}, with $m \in \{(4,4),(4,1),(1,4)\}$, we can express (13) as

$$(L_{mh,p}v_{mh,p})(\bar{x}) = \sum_{i\in s_m}\sum_{j\in s_m}\int_{\theta\in\Pi_{(4,4)}} \widehat{L_{mh,p}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)\widehat{v_{mh,p}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)e^{\imath m\theta_{\delta_n^j}^{\gamma_n^i}\cdot\bar{x}/(mh)}d\theta,$$

with the Fourier symbol $\widehat{L_{mh,p}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)$ defined as

$$\widehat{L_{mh,p}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right) = \sum_{k\in J_{L_{mh,p}}} l_{k,mh,p}e^{\imath m\theta_{\delta_n^j}^{\gamma_n^i}\cdot k} \in \mathbb{C}^{m_p\times m_p}.$$

5.3.2. Discrete Fourier transform of pseudo-time smoother

Using the relations for the space–time discretization operators $L_{nh,p}$ we obtain the Fourier symbols of the Runge–Kutta pseudo-time integration operator discussed in Section 3.2.1. The Fourier symbol of $S_{h,p}^l, l = 1,2$, on the mesh G_h, given by (6), is equal to

$$\begin{aligned}
\widehat{Q_0}\left(\theta_\beta^\alpha\right) &= I^{m_p},\\
\widehat{Q_k}\left(\theta_\beta^\alpha\right) &= \left(I^{m_p} + \beta_k\lambda_\sigma\widehat{L_{h,p}^{l,1}}\left(\theta_\beta^\alpha\right)\right)^{-1}\left(I^{m_p} - \lambda_\sigma\sum_{j=0}^{k-1}\alpha_{kj}\widehat{L_{h,p}^{l,2}}\left(\theta_\beta^\alpha\right)\widehat{Q_j}\left(\theta_\beta^\alpha\right)\right),\quad k = 1,\ldots,5,\\
\widehat{S_{h,p}^l}\left(\theta_\beta^\alpha\right) &= \widehat{Q_5}\left(\theta_\beta^\alpha\right),\quad \forall\alpha\in\alpha_2,\ \forall\beta\in\beta_2.
\end{aligned}$$

On the coarse meshes G_{nh} the Fourier symbol of the semi-implicit pseudo-time Runge–Kutta operator $S_{nh,p}^l, l = 1,2$, is equal to

$$\widehat{S_{nh,p}^l}(n\theta_\beta^{\gamma_n^r}) = \widehat{Q_5}(n\theta_\beta^{\gamma_n^r}),\quad \forall\beta\in\beta_n^s,\ r,s\in s_n.$$

5.3.3. Restriction operators for h-multigrid

Define the restriction operator $R_{h,p}^{nh} : \mathcal{F}(G_h) \to \mathcal{F}(G_{nh})$, with $n \in \{(2,2),(2,1),(1,2)\}$, as

$$(R_{h,p}^{nh}v_{h,p})(\bar{x}) = \sum_{k\in J_{R_{h,p}^{nh}}} r_{k,nh,p}v_{h,p}(\bar{x}+kh),\quad \bar{x}\in G_{nh},\ \bar{x}+kh\in G_h,$$

with $J_{R_{h,p}^{nh}}$ the stencil of the restriction operator and $r_{k,nh,p} \in \mathbb{R}^{m_p\times m_p}$ the matrices defining the restriction operator. On the mesh G_{nh} the restriction operator can be related to its discrete Fourier transform through the relation

$$\left(R_{h,p}^{nh}v_{h,p}\right)(\bar{x}) = \sum_{i\in s_n}\sum_{j\in s_n}\sum_{\beta\in\beta_n^j}\int_{\theta\in\Pi_{(4,4)}}\sum_{\alpha\in\alpha_n^i}\widehat{R_{h,p}^{nh}}\left(\theta_\beta^\alpha\right)\widehat{v_{h,p}}\left(\theta_\beta^\alpha\right)e^{\imath n\theta_\beta^{\gamma_n^i}(\theta)\cdot\bar{x}/(nh)}d\theta, \tag{20}$$

with the Fourier symbol $\widehat{R_{h,p}^{nh}}\left(\theta_\beta^\alpha\right)$ given by

$$\widehat{R_{h,p}^{nh}}\left(\theta_\beta^\alpha\right) = \sum_{k\in J_{R_{h,p}^{nh}}} r_{k,nh,p}e^{\imath\theta_\beta^\alpha\cdot k}.$$

Note, in (20) we used the subdivision of modes with frequency $\theta \in \Pi_n$ into different groups as discussed in Section 5.2.

Next, we define the restriction operator $R_{nh,p}^{mh} : \mathcal{F}(G_{nh}) \to \mathcal{F}(G_{mh})$, with $m \in \{(4,4),(4,1),(1,4)\}$, as

$$\left(R_{nh,p}^{mh}v_{nh,p}\right)(\bar{x}) = \sum_{k\in J_{R_{nh,p}^{mh}}} r_{k,mh,p}v_{nh,p}(\bar{x}+knh),\quad \bar{x}\in G_{mh},\ \bar{x}+knh\in G_{nh}.$$

On the mesh G_{mh} the restriction operator can be related to its discrete Fourier transform through the relation

$$\left(R^{mh}_{nh,p}v_{nh,p}\right)(\bar{x}) = \sum_{i\in s_n}\sum_{j\in s_n}\int_{\theta\in\Pi_{(4,4)}}\sum_{\beta\in\beta^j_n}\widehat{R^{mh}_{nh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)\widehat{v_{nh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)e^{\imath n\theta^{\gamma^i_n}_{\delta^j_n}\cdot\bar{x}/(mh)}d\theta, \tag{21}$$

with the Fourier symbol $\widehat{R^{mh}_{nh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)$ given by

$$\widehat{R^{mh}_{nh,p}}\left(n\theta^{\gamma^i_n}_\beta\right) = \sum_{k\in J_{R^{mh}_{nh,p}}} r_{k,mh,p}e^{\imath n\theta^{\gamma^i_n}_\beta\cdot k}, \quad \forall\beta\in\beta^j_n,\ i,j\in s_n.$$

5.3.4. Prolongation operators for h-multigrid

The definition of the prolongation operator $P^h_{nh,p}:\mathcal{F}(G_{nh})\to\mathcal{F}(G_h)$, with $n\in\{(2,2),(2,1),(1,2)\}$, requires the introduction of subsets of the mesh G_h, which each have a different prolongation operator. Define the meshes G^κ_h as

$$G^\kappa_h := \left\{(x_1,x_2)\in\mathbb{R}^2 \mid (x_1,x_2) = ((n_1j_1+\bar{\kappa}_1)h_1,(n_2j_2+\bar{\kappa}_2)h_2),\ j\in\mathbb{Z}^2\right\},$$

with $\kappa\in\kappa_n := \{\kappa=(\bar{\kappa}_1,\bar{\kappa}_2)\mid\bar{\kappa}_i\in\{0,n_i-1\},i=1,2\}$. The prolongation operator related to the mesh G^κ_h then is equal to

$$(P^h_{nh,p}v_h)(\bar{x}) = \sum_{k\in J^\kappa_{P^h_{nh,p}}} p^\kappa_{k,h,p}v_{nh}(\bar{x}+kh), \quad \bar{x}\in G^\kappa_h,\ \bar{x}+kh\in G_{nh},$$

where the index set $J^\kappa_{P^h_{nh,p}}\subset\mathbb{Z}^2$ and matrices $p^\kappa_{k,h,p}\in\mathbb{R}^{m_p\times m_p}$ are used to define the prolongation operator on each mesh. We consider now the prolongation operator $P^n_{nh,p}:\mathcal{F}(G_{nh})\to\mathcal{F}(G_h)$, with $n\in\{(2,2),(2,1),(1,2)\}$. The prolongation operator $P^h_{nh,p}$ is related to its discrete Fourier transform through the relation

$$\left(P^h_{nh,p}v_{nh,p}\right)(\bar{x}) = \sum_{i\in s_n}\sum_{j\in s_n}\sum_{\alpha\in\alpha^i_n}\sum_{\beta\in\beta^j_n}\int_{\theta\in\Pi_{(4,4)}}\widehat{P^h_{nh,p}}\left(\theta^\alpha_\beta\right)\widehat{v_{nh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)e^{\imath\theta^\alpha_\beta\cdot\bar{x}/h}d\theta, \tag{22}$$

with the Fourier symbol $\widehat{P^h_{nh,p}}\left(\theta^\alpha_\beta\right)$ given by

$$\widehat{P^h_{nh,p}}\left(\theta^\alpha_\beta\right) = \frac{1}{n_1n_2}\sum_{\kappa\in\kappa_2}\sum_{k\in J^\kappa_{P^{nh}_{mh,p}}} p^\kappa_{k,h,p}e^{\imath\theta^\alpha_\beta\cdot k}.$$

Next, we consider the prolongation operator $P^{nh}_{mh,p}:\mathcal{F}(G_{mh})\to\mathcal{F}(G_{nh})$, with $m\in\{(4,4),(4,1),(1,4)\}$. The definition of the prolongation operator requires the introduction of subsets of the mesh G_{nh}. Define the meshes G^κ_{nh} as

$$G^\kappa_{nh} := \left\{(x_1,x_2)\in\mathbb{R}^2 \mid (x_1,x_2) = ((m_1j_1+\bar{\kappa}_1)h_1,(m_2j_2+\bar{\kappa}_2)h_2),\ j\in\mathbb{Z}^2\right\},$$

with $\kappa\in\kappa_m := \{\kappa=(\bar{\kappa}_1,\bar{\kappa}_2)\mid\bar{\kappa}_i\in\{0,(2m_i-2)/3\},\ i=1,2\}$. The prolongation operator related to the mesh G^κ_{nh} then is equal to

$$\left(P^{nh}_{mh,p}v_{mh,p}\right)(\bar{x}) = \sum_{k\in J^\kappa_{P^{nh}_{mh,p}}} p^\kappa_{k,nh,p}v_{mh,p}(\bar{x}+knh), \quad \bar{x}\in G^\kappa_{nh},\ \bar{x}+knh\in G_{mh}.$$

The prolongation operator $P^{nh}_{mh,p}$ is related to its discrete Fourier transform through the relation

$$\left(P^{nh}_{mh,p}v_{mh,p}\right)(\bar{x}) = \sum_{i\in s_n}\sum_{j\in s_n}\sum_{\beta\in\beta^j_n}\int_{\theta\in\Pi_{(4,4)}}\widehat{P^{nh}_{mh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)\widehat{v_{mh,p}}\left(m\theta^{\gamma^i_n}_{\delta^j_n}\right)e^{\imath n\theta^{\gamma^i_n}_\beta\cdot\bar{x}/(nh)}d\theta, \tag{23}$$

with the Fourier symbol $\widehat{P^{nh}_{mh,p}}\left(n\theta^{\gamma^i_n}_\beta\right)$ given by

$$\widehat{P^{nh}_{mh,p}}\left(n\theta^{\gamma^i_n}_\beta\right) = \frac{n_1n_2}{m_1m_2}\sum_{\kappa\in\kappa_2}\sum_{k\in J^\kappa_{P^{nh}_{mh,p}}} p^\kappa_{k,nh,p}e^{\imath n\theta^{\gamma^i_n}_\beta\cdot k}.$$

5.3.5. Restriction and prolongation operators for p-multigrid

Define the p-multigrid prolongation operators $T^p_{h,p-1}:\mathcal{F}(G_h)\to\mathcal{F}(G_h)$ in stencil notation as

$$\left(T^p_{h,p-1}v_{h,p-1}\right)(\bar{x}) = t_{h,p}v_{h,p}(\bar{x}), \quad \bar{x}\in G_h,$$

where $t_{h,p}\in\mathbb{R}^{m_p\times m_p}$ is the matrix defining the p-multigrid prolongation operator in a space–time element. Since this is a purely element based operator it immediately follows that its Fourier symbol is equal to

$$\widehat{T^p_{h,p-1}} = t_{h,p}. \tag{24}$$

The p-multigrid restriction operator $Q_{h,p}^{p-1}:\mathcal{F}(G_h)\to\mathcal{F}(G_h)$ is equal to the transpose of the p-multigrid prolongation operator. The Fourier symbol of the p-restriction operator then is equal to

$$\widehat{Q_{h,p}^{p-1}}=\left(\widehat{T_{h,p-1}^{p}}\right)^{T}. \tag{25}$$

5.4. Three-level Fourier analysis

In this section we will describe the discrete Fourier analysis of the three-level h-multigrid error transformation operator. A unified formulation for both uniform and semi-coarsening multigrid will be presented. This unified formulation makes the construction of the complete *hp-MGS* error transformation operator, which is discussed in Section 5.5, much easier. In order to simplify notation we omit in this section the subscript p in the discrete operators. It should be kept in mind, however, that all discrete operators depend on the polynomial order p of the space–time DG discretization.

In the three-level analysis the Fourier symbols $\widehat{L_h}(\theta)$, $\widehat{L_{nh}}(n\theta)$ and $\widehat{L_{mh}}(m\theta)$, with $n\in\{(2,2),(2,1),(1,2)\}, m\in\{(4,4),(4,1),(1,4)\}$ can be zero for certain values of θ. The frequencies of these Fourier harmonics are removed from $\mathcal{F}_h^3(\theta)$ through the definition of

$$\mathcal{F}_h^{3g}:=\left\{\mathcal{F}_h^3(\theta_\beta^\alpha(\theta))\mid\theta\in\Pi_{(4,4)}\setminus\Psi_{n,m},\ \forall\alpha\in\alpha_n^i,\ \forall\beta\in\beta_n^j,\ i,j\in s_n,\ \forall n\in\{(2,2),(2,1),(1,2)\},\ \forall m\in\{(4,4),(4,1),(1,4)\}\right\},$$

with

$$\Psi_{n,m}:=\left\{\theta\in\Pi_{(4,4)}\mid\det\left(\widehat{L_h}\left(\theta_\beta^\alpha(\theta)\right)\right)=0\text{ or }\det\left(\widehat{L_{nh}}\left(n\theta_\beta^{\gamma_n^i}(\theta)\right)\right)=0\text{ or }\det\left(\widehat{L_{mh}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}(\theta)\right)\right)=0,\ \forall\alpha\in\alpha_n^i,\ \forall\beta\in\beta_n^j,\ i,j\in s_n\right\} \tag{26}$$

and $\theta_\beta^\alpha=\theta_\beta^\alpha(\theta)$ given by (17). The error e_h^D after a three-level multigrid cycle is equal to

$$e_h^D=M_h^{3g}e_h^A, \tag{27}$$

with e_h^A the initial error. The three-level multigrid error transformation operator M_h^{3g} can be obtained from (9) and is equal to

$$M_h^{3g}=S_h^{\nu_2}\left(I_h-P_{nh}^h(I_{nh}-M_{nh}^m)(L_{nh})^{-1}R_h^{nh}L_h\right)S_h^{\nu_1} \tag{28}$$

with the coarse grid correction defined as

$$M_{nh}^m=S_{nh}^{\nu_2}\left(I_{nh}-P_{mh}^{nh}(L_{mh})^{-1}R_{nh}^{mh}L_{nh}\right)S_{nh}^{\nu_1}. \tag{29}$$

Here $L_h,\ L_{nh}$ and L_{mh} denote the matrices of the DG discretization (4) on the meshes G_h, G_{nh} and G_{mh}, respectively, $S_h,\ S_{nh}$ the multigrid smoothers, which can be either the semi-coarsening smoothers $HS_{nh,p}^2HS_{nh,p}^1,\ HS_{nh,p}^1HS_{nh,p}^2$, or the semi-implicit Runge–Kutta smoother $S_{nh,p}^i,\ i=1,2$. Furthermore, $R_h^{nh},\ R_{nh}^{mh}$ are the restriction operators, $P_{nh}^h,\ P_{mh}^{nh}$ the prolongation operators, and $I_h,\ I_{nh}$ the identity operators on the meshes G_h and G_{nh}, respectively. The parameters $\nu_1,\ \nu_2$ denote the number of pre- and post-smoothing iterations.

We start the three-level analysis with the coarse grid contribution (29) on the mesh G_{nh}, which can be expressed as

$$e_{nh}^D(\bar{x})=\sum_{i\in s_n}\sum_{j\in s_n}\sum_{\beta\in\beta_n^j}\int_{\theta\in\Pi_{(4,4)}}\widehat{M_{nh}^me_{nh}^A}\left(n\theta_\beta^{\gamma_n^i}\right)e^{\imath n\theta_\beta^{\gamma_n^i}\cdot\bar{x}/(nh)}d\theta,$$

with

$$\begin{aligned}\widehat{M_{nh}^me_{nh}^A}\left(n\theta_\beta^{\gamma_n^i}\right)=&\left(\widehat{S_{nh}}\left(n\theta_\beta^{\gamma_n^i}\right)\right)^{\nu_1+\nu_2}\widehat{e_{nh}^A}\left(n\theta_\beta^{\gamma_n^i}\right)-\left(\widehat{S_{nh}}\left(n\theta_\beta^{\gamma_n^i}\right)\right)^{\nu_2}\widehat{P_{mh}^{nh}}\left(n\theta_\beta^{\gamma_n^i}\right)\left(\widehat{L_{mh}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)\right)^{-1}\\&\times\sum_{\beta_2\in\beta_n^j}\widehat{R_{nh}^{mh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\widehat{L_{nh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\left(\widehat{S_{nh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\right)^{\nu_1}\widehat{e_{nh}^A}\left(n\theta_{\beta_2}^{\gamma_n^i}\right),\quad\forall\beta\in\beta_n^j,\ i,j\in s_n\end{aligned} \tag{30}$$

using the expressions for the Fourier symbols of the discrete operators given in Section 5.3. Define now the coarse grid correction operator

$$\widetilde{M}_{nh}^m=I_{nh}-M_{nh}^m. \tag{31}$$

If we introduce the matrices $\left(\widehat{\widetilde{M}_{nh}^m}\right)_\beta\in\mathbb{C}^{qr\times qr}$, with $\beta\in\beta_n^j,\ i,j\in s_n,\ q=m_p$ the size of the blocks in the space–time DG discretization, and $r=\mathrm{Car}(\alpha_n^i)=\mathrm{Car}(\beta_n^j)$, then we can write the discrete Fourier transform of $\widetilde{M}_{nh}^me_{nh}^A$ as

$$\widehat{\widetilde{M}_{nh}^me_{nh}^A}\left(n\theta_\beta^{\gamma_n^i}\right)=\sum_{\beta_2\in\beta_n^j}\left(\widehat{\widetilde{M}_{nh}^m}\right)_{\beta_2}\left(n\theta_\beta^{\gamma_n^i}\right)\widehat{e_{nh}^A}\left(n\theta_{\beta_2}^{\gamma_n^i}\right),\qquad\forall\beta\in\beta_n^j,\ i,j\in s_n, \tag{32}$$

where an explicit expression of $\left(\widehat{\widetilde{M}_{nh}^{m}}\right)_{\beta_2}\left(n\theta_{\beta}^{\gamma_n^i}\right)$ can be obtained using (30)

$$\left(\widehat{\widetilde{M}_{nh}^{m}}\right)_{\beta_2}\left(n\theta_{\beta}^{\gamma_n^i}\right)=I^{qr}-\left(\widehat{S_{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\right)^{\nu_1+\nu_2}+\left(\widehat{S_{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\right)^{\nu_2}\widehat{P_{mh}^{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\left(\widehat{L_{mh}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)\right)^{-1}\widehat{R_{nh}^{mh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\widehat{L_{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\left(\widehat{S_{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\right)^{\nu_1},\text{if } \beta_2=\beta,$$

$$=\left(\widehat{S_{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\right)^{\nu_2}\widehat{P_{mh}^{nh}}\left(n\theta_{\beta}^{\gamma_n^i}\right)\left(\widehat{L_{mh}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)\right)^{-1}\widehat{R_{nh}^{mh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\widehat{L_{nh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\left(\widehat{S_{nh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\right)^{\nu_1},\text{ if } \beta_2\neq\beta.$$

Next, we compute the Fourier symbol of the error transformation operator M_h^{3g} on the mesh G_h. Using (28) and the Fourier symbols of the individual discrete operators discussed in Section 5.3 the error in the three-level multigrid algorithm can now be expressed as

$$e_h^D(\bar{x})=\sum_{i\in s_n}\sum_{j\in s_n}\sum_{\alpha\in\alpha_n^i}\sum_{\beta\in\beta_n^j}\int_{\theta\in\Pi_{(4,4)}}\widehat{M_h^{3g}e_h^A}\left(\theta_\beta^\alpha\right)e^{\imath\theta_\beta^\alpha\cdot\bar{x}/h}d\theta,$$

with

$$\widehat{M_h^{3g}e_h^A}\left(\theta_\beta^\alpha\right)=\left(\widehat{S_h}\left(\theta_\beta^\alpha\right)\right)^{\nu_1+\nu_2}\widehat{e^A}\left(\theta_\beta^\alpha\right)-\left(\widehat{S_h}\left(\theta_\beta^\alpha\right)\right)^{\nu_2}\widehat{P_{nh}^h}\left(\theta_\beta^\alpha\right)\sum_{\beta_2\in\beta_n^j}\left(\widehat{\widetilde{M}_{nh}^{m}}\right)_{\beta_2}\left(n\theta_\beta^{\gamma_n^i}\right)\left(\widehat{L_{nh}}\left(n\theta_{\beta_2}^{\gamma_n^i}\right)\right)^{-1}\sum_{\alpha_2\in\alpha_n^i}\widehat{R_h^{nh}}\left(\theta_{\beta_2}^{\alpha_2}\right)\widehat{L_h}\left(\theta_{\beta_2}^{\alpha_2}\right)$$
$$\times\left(\widehat{S_h}\left(\theta_{\beta_2}^{\alpha_2}\right)\right)^{\nu_1}\widehat{e_h^A}\left(\theta_{\beta_2}^{\alpha_2}\right),\ \forall\alpha\in\alpha_n^i,\ \forall\beta\in\beta_n^j,\ i,j\in s_n.$$

The expressions for the discrete Fourier transform of the error transformation operator can be simplified using a matrix representation. On the mesh G_{nh} we introduce the matrices

$$\widehat{L}_{nh}^{n}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=\mathrm{bdiag}\left(\widehat{L_{nh}}\left(n\theta_{\beta_1}^{\gamma_n^i}\right),\ldots,\widehat{L_{nh}}\left(n\theta_{\beta_r}^{\gamma_n^i}\right)\right)\in\mathbb{C}^{qr\times qr},\tag{33}$$

$$\widehat{S}_{nh}^{n}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=\mathrm{bdiag}\left(\widehat{S_{nh}}\left(n\theta_{\beta_1}^{\gamma_n^i}\right),\ldots,\widehat{S_{nh}}\left(n\theta_{\beta_r}^{\gamma_n^i}\right)\right)\in\mathbb{C}^{qr\times qr},\tag{34}$$

$$\widehat{R}_{nh}^{mh}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=\left(\widehat{R_{nh}^{mh}}\left(n\theta_{\beta_1}^{\gamma_n^i}\right),\ldots,\widehat{R_{nh}^{mh}}\left(n\theta_{\beta_r}^{\gamma_n^i}\right)\right)\in\mathbb{C}^{q\times qr},\tag{35}$$

$$\widehat{P}_{mh}^{nh}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=\left(\widehat{P_{mh}^{nh}}\left(n\theta_{\beta_1}^{\gamma_n^i}\right),\ldots,\widehat{P_{mh}^{nh}}\left(n\theta_{\beta_r}^{\gamma_n^i}\right)\right)^T\in\mathbb{C}^{qr\times q},\tag{36}$$

with $\theta_{\beta_n^j}^{\gamma_n^i}=\left(\theta_{\beta_1}^{\gamma_n^i},\ldots\theta_{\beta_r}^{\gamma_n^i}\right)^T,\ \beta_1,\ldots,\beta_r\in\beta_n^j,\ r=\mathrm{Car}(\alpha_n^i)=\mathrm{Car}(\beta_n^j),\ i,j\in s_n$, and bdiag refers to a block diagonal matrix consisting of $q\times q$ blocks with $q\geqslant 1$. For each group of modes $\beta_n^j, j\in s_n$, the discrete Fourier transform of the coarse grid multigrid error transformation operator $\widehat{M}_{nh}^m$ can be directly obtained from (30) resulting in

$$\widehat{M}_{nh}^{m}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=\left(\widehat{S}_{nh}^{n}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\right)^{\nu_2}\left(I^{qr}-\widehat{P}_{mh}^{nh}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\left(\widehat{L_{mh}}\left(m\theta_{\delta_n^j}^{\gamma_n^i}\right)\right)^{-1}\widehat{R}_{nh}^{mh}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\widehat{L}_{nh}^{n}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\right)\left(\widehat{S}_{nh}^{n}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\right)^{\nu_1}\in\mathbb{C}^{qr\times qr},\quad i,j\in s_n,$$

with $I^{qr}\in\mathbb{R}^{qr\times qr}$ the identity matrix. The matrices representing the discrete Fourier transform of the coarse grid operator (31) then are equal to

$$\widehat{\widetilde{M}_{nh}^{m}}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)=I^{qr}-\widehat{M}_{nh}^{m}\left(n\theta_{\beta_n^j}^{\gamma_n^i}\right)\in\mathbb{C}^{qr\times qr},\quad i,j\in s_n.$$

Next, we introduce for each group of modes α_n^i,β_n^j, with $i,j\in s_n$, the matrices

$$\widetilde{L}_h^n\left(\theta_{\beta_k}^{\alpha_n^i}\right)=\mathrm{bdiag}\left(\widehat{L_h}\left(\theta_{\beta_k}^{\alpha_1}\right),\ldots,\widehat{L_h}\left(\theta_{\beta_k}^{\alpha_r}\right)\right)\in\mathbb{C}^{qr\times qr},\tag{37}$$

$$\overline{L}_h^n\left(\theta_{\beta_n^j}^{\alpha_n^i}\right)=\mathrm{bdiag}\left(\widetilde{L}_h^n\left(\theta_{\beta_1}^{\alpha_n^i}\right),\ldots,\widetilde{L}_h^n\left(\theta_{\beta_r}^{\alpha_n^i}\right)\right)\in\mathbb{C}^{qr^2\times qr^2},\tag{38}$$

$$\widetilde{S}_h^n\left(\theta_{\beta_k}^{\alpha_n^i}\right)=\mathrm{bdiag}\left(\widehat{S_h}\left(\theta_{\beta_k}^{\alpha_1}\right),\ldots,\widehat{S_h}\left(\theta_{\beta_k}^{\alpha_r}\right)\right)\in\mathbb{C}^{qr\times qr}\tag{39}$$

$$\overline{S}_h^n\left(\theta_{\beta_n^j}^{\alpha_n^i}\right)=\mathrm{bdiag}\left(\widetilde{S}_h^n\left(\theta_{\beta_1}^{\alpha_n^i}\right),\ldots,\widetilde{S}_h^n\left(\theta_{\beta_r}^{\alpha_n^i}\right)\right)\in\mathbb{C}^{qr^2\times qr^2},\tag{40}$$

$$\widetilde{R}_h^{nh}\left(\theta_{\beta_k}^{\alpha_n^i}\right)=\left(\widehat{R_h^{nh}}\left(\theta_{\beta_k}^{\alpha_1}\right),\ldots,\widehat{R_h^{nh}}\left(\theta_{\beta_k}^{\alpha_r}\right)\right)\in\mathbb{C}^{q\times qr},\tag{41}$$

$$\overline{R}_h^{nh}\left(\theta_{\beta_n^j}^{\alpha_n^i}\right)=\mathrm{bdiag}\left(\widetilde{R}_h^{nh}\left(\theta_{\beta_1}^{\alpha_n^i}\right),\ldots,\widetilde{R}_h^{nh}(\theta_{\beta_r}^{\alpha_n^i})\right)\in\mathbb{C}^{qr\times qr^2}\tag{42}$$

$$\widetilde{P}^h_{nh}\left(\theta^{\alpha^i_n}_{\beta_k}\right) = \left(\widehat{P^h_{nh}}\left(\theta^{\alpha_1}_{\beta_k}\right), \ldots, \widehat{P^h_{nh}}\left(\theta^{\alpha_r}_{\beta_k}\right)\right)^T \in \mathbb{C}^{qr\times q}, \tag{43}$$

$$\overline{P}^h_{nh}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right) = \text{bdiag}\left(\widetilde{P}^h_{nh}\left(\theta^{\alpha^i_n}_{\beta_1}\right), \ldots, \widetilde{P}^h_{nh}\left(\theta^{\alpha^i_n}_{\beta_r}\right)\right) \in \mathbb{C}^{qr^2\times qr}, \tag{44}$$

$$\overline{Q}^n_{nh}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right) = \left(\text{bdiag}\left(\widehat{L_{nh}}\left(n\theta^{\gamma^i_n}_{\beta_1}\right), \ldots, \widehat{L_{nh}}\left(n\theta^{\gamma^i_n}_{\beta_r}\right)\right)\right)^{-1} \in \mathbb{C}^{qr\times qr}, \tag{45}$$

with $\theta^{\alpha^i_n}_{\beta_k} = \left(\theta^{\alpha_1}_{\beta_k}, \ldots, \theta^{\alpha_r}_{\beta_k}\right)^T$, $\theta^{\alpha^i_n}_{\beta^j_n} = \left(\theta^{\alpha^i_n}_{\beta_1}, \ldots, \theta^{\alpha^i_n}_{\beta_r}\right)^T$, $\alpha_1, \ldots, \alpha_r \in \alpha^i_n$, $\beta_1, \ldots, \beta_r \in \beta^j_n$.

The discrete Fourier transform of the error transformation operator for a three-level multigrid cycle can now be expressed for each group of Fourier modes as

$$\widehat{M}^n_h\left(\theta^{\alpha^i_n}_{\beta^j_n}\right) = \left(\overline{S}^n_h\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)^{\nu_2}\left(I^{r^2q} - \overline{P}^h_{nh}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\widehat{\widetilde{M}^m_{nh}}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\overline{Q}^n_{nh}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\overline{R}^{nh}_h(\theta^{\alpha^i_n}_{\beta^j_n})\overline{L}^n_h\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)\left(\overline{S}^n_h\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)^{\nu_1} \in \mathbb{C}^{r^2q\times r^2q}, \quad i,j \in s_n. \tag{46}$$

The discrete Fourier transform of the three-level error transformation operator for different types of mesh coarsening can now be obtained by combining the contributions from the different groups of Fourier modes. For uniform coarsening the multigrid error transformation operator is equal to

$$\widehat{M}^{(2,2)}_h\left(\theta^\alpha_\beta\right) = \widehat{M}^{(2,2)}_h\left(\theta^{\alpha^1_{(2,2)}}_{\beta^1_{(2,2)}}\right) \in \mathbb{C}^{16q\times 16q}, \tag{47}$$

with $\theta^\alpha_\beta = \theta^{\alpha^1_{(2,2)}}_{\beta^1_{(2,2)}}$. The discrete Fourier transform of the error after one three-level multigrid cycle with uniform coarsening can now be expressed as

$$\widehat{e^D_h}\left(\theta^\alpha_\beta\right) = \widehat{M}^{(2,2)}_h\left(\theta^\alpha_\beta\right)\widehat{e^A_h}\left(\theta^\alpha_\beta\right),$$

with

$$\widehat{e^{A,D}_h}\left(\theta^\alpha_\beta\right) = \left(\widehat{e^D_h}\left(\theta^{\alpha_1}_{\beta_1}\right), \ldots, \widehat{e^{A,D}_h}\left(\theta^{\alpha_4}_{\beta_1}\right), \widehat{e^{A,D}_h}\left(\theta^{\alpha_1}_{\beta_2}\right), \ldots, \widehat{e^{A,D}_h}\left(\theta^{\alpha_4}_{\beta_2}\right), \ldots, \widehat{e^{A,D}_h}\left(\theta^{\alpha_1}_{\beta_4}\right), \ldots, \widehat{e^{A,D}_h}(\theta^{\alpha_4}_{\beta_4})\right)^T, \quad \alpha_1, \ldots, \alpha_4 \in \alpha^1_{(2,2)},$$
$$\beta_1, \ldots, \beta_4 \in \beta^1_{(2,2)}.$$

The discrete Fourier transform of the multigrid error transformation operator for semi-coarsening in the $\bar{x}_1$-direction is

$$\widehat{M}^{(2,1)}_h\left(\theta^{\alpha_{(2,1)}}_{\beta_{(2,1)}}\right) = \text{bdiag}\left(\widehat{M}^{(2,1)}_h\left(\theta^{\alpha^1_{(2,1)}}_{\beta^1_{(2,1)}}\right), \widehat{M}^{(2,1)}_h\left(\theta^{\alpha^2_{(2,1)}}_{\beta^1_{(2,1)}}\right), \widehat{M}^{(2,1)}_h\left(\theta^{\alpha^1_{(2,1)}}_{\beta^2_{(2,1)}}\right), \widehat{M}^{(2,1)}_h\left(\theta^{\alpha^2_{(2,1)}}_{\beta^2_{(2,1)}}\right)\right) \in \mathbb{C}^{16q\times 16q},$$

with $\theta^{\alpha_{(2,1)}}_{\beta_{(2,1)}} = \left(\theta^{\alpha^1_{(2,1)}}_{\beta^1_{(2,1)}}, \theta^{\alpha^2_{(2,1)}}_{\beta^1_{(2,1)}}, \theta^{\alpha^1_{(2,1)}}_{\beta^2_{(2,1)}}, \theta^{\alpha^2_{(2,1)}}_{\beta^2_{(2,1)}}\right)^T$. The frequencies $\theta^{\alpha^i_{(2,1)}}_{\beta^j_{(2,1)}}$, $i,j \in s_n$, are defined as

$$\theta^{\alpha^1_{(2,1)}}_{\beta^1_{(2,1)}} = \left(\theta^{00}_{00}, \theta^{10}_{00}, \theta^{00}_{\frac{1}{2}0}, \theta^{10}_{\frac{1}{2}0}\right)^T, \quad \theta^{\alpha^2_{(2,1)}}_{\beta^1_{(2,1)}} = \left(\theta^{11}_{00}, \theta^{01}_{00}, \theta^{11}_{\frac{1}{2}0}, \theta^{01}_{\frac{1}{2}0}\right)^T,$$
$$\theta^{\alpha^1_{(2,1)}}_{\beta^2_{(2,1)}} = \left(\theta^{00}_{\frac{1}{2}\frac{1}{2}}, \theta^{10}_{\frac{1}{2}\frac{1}{2}}, \theta^{00}_{0\frac{1}{2}}, \theta^{10}_{0\frac{1}{2}}\right)^T, \quad \theta^{\alpha^2_{(2,1)}}_{\beta^2_{(2,1)}} = \left(\theta^{11}_{\frac{1}{2}\frac{1}{2}}, \theta^{01}_{\frac{1}{2}\frac{1}{2}}, \theta^{11}_{0\frac{1}{2}}, \theta^{01}_{0\frac{1}{2}}\right)^T.$$

Note, however, that the Fourier modes in the error vectors for semi-coarsening in the $\bar{x}_1$ and $\bar{x}_2$-direction have a different ordering than for uniform coarsening $\widehat{e^{A,D}_h}\left(\theta^\alpha_\beta\right)$. The ordering of the Fourier modes in the error vectors is not important for the computation of the operator norms and the spectral radius of the error transformation operator when one particular type of mesh coarsening is used. For the coupling of multigrid algorithms with different types of mesh coarsening, which is required for the multilevel analysis of the *hp-MGS* algorithm, it is, however, essential that the same ordering of the Fourier modes in the error vectors is used. This can be easily accomplished using the permutation matrix $P^{(2,1)}_h \in \mathbb{R}^{16q\times 16q}$, which reorders the Fourier modes in the error vector for semi-coarsening in the $\bar{x}_1$-direction to that of $\widehat{e^{A,D}_h}(\theta^{\alpha_{(2,1)}}_{\beta_{(2,1)}})$ to the error vector for uniform coarsening $\widehat{e^{A,D}_h}\left(\theta^\alpha_\beta\right)$. The permutation matrix consists of blocks of size $q \times q$. All blocks in the permutation matrix $P^{(2,1)}_h$ are zero, except the blocks with indices

$$(1,1),(2,3),(3,9),(4,11),(5,2),(6,4),(7,10),(8,12),(9,5),(10,7),(11,13),(12,15),(13,6),(14,8),(15,14),(16,16),$$

which are equal to the identity matrix I^q. The error after one three-level multigrid cycle with semi-coarsening in the $\bar{x}_1$-direction can now be expressed as

$$\widehat{e^D_h}\left(\theta^\alpha_\beta\right) = \left(P^{(2,1)}_h\right)^{-1}\widehat{M}^{(2,1)}_h\left(\theta^{\alpha_{(2,1)}}_{\beta_{(2,1)}}\right)P^{(2,1)}_h\widehat{e^A_h}\left(\theta^\alpha_\beta\right).$$

Finally, the discrete Fourier transform of the multigrid error transformation operator for semi-coarsening in the $\bar{x}_2$-direction is

$$\widehat{M}_h^{(1,2)}\left(\theta_{\beta_{(1,2)}}^{\alpha_{(1,2)}}\right) = \mathrm{bdiag}\left(\widehat{M}_h^{(1,2)}\left(\theta_{\beta^1_{(1,2)}}^{\alpha^1_{(1,2)}}\right), \widehat{M}_h^{(1,2)}\left(\theta_{\beta^1_{(1,2)}}^{\alpha^2_{(1,2)}}\right), \widehat{M}_h^{(1,2)}\left(\theta_{\beta^2_{(1,2)}}^{\alpha^1_{(1,2)}}\right), \widehat{M}_h^{(1,2)}\left(\theta_{\beta^2_{(1,2)}}^{\alpha^2_{(1,2)}}\right)\right) \in \mathbb{C}^{16q\times 16q},$$

with $\theta_{\beta_{(1,2)}}^{\alpha_{(1,2)}} = \left(\theta_{\beta^1_{(1,2)}}^{\alpha^1_{(1,2)}}, \theta_{\beta^1_{(1,2)}}^{\alpha^2_{(1,2)}}, \theta_{\beta^2_{(1,2)}}^{\alpha^1_{(1,2)}}, \theta_{\beta^2_{(1,2)}}^{\alpha^2_{(1,2)}}\right)^T$. The frequencies $\theta_{\beta^j_{(1,2)}}^{\alpha^i_{(1,2)}}$ are defined as

$$\theta_{\beta^1_{(1,2)}}^{\alpha^1_{(1,2)}} = \left(\theta_{00}^{00}, \theta_{00}^{01}, \theta_{0\frac{1}{2}}^{00}, \theta_{0\frac{1}{2}}^{01}\right)^T, \quad \theta_{\beta^1_{(1,2)}}^{\alpha^2_{(1,2)}} = \left(\theta_{00}^{11}, \theta_{00}^{10}, \theta_{0\frac{1}{2}}^{11}, \theta_{0\frac{1}{2}}^{10}\right)^T,$$
$$\theta_{\beta^2_{(1,2)}}^{\alpha^1_{(1,2)}} = \left(\theta_{\frac{1}{2}\frac{1}{2}}^{00}, \theta_{\frac{1}{2}\frac{1}{2}}^{01}, \theta_{\frac{1}{2}0}^{00}, \theta_{\frac{1}{2}0}^{01}\right)^T, \quad \theta_{\beta^2_{(1,2)}}^{\alpha^2_{(1,2)}} = \left(\theta_{\frac{1}{2}\frac{1}{2}}^{11}, \theta_{\frac{1}{2}\frac{1}{2}}^{10}, \theta_{\frac{1}{2}0}^{11}, \theta_{\frac{1}{2}0}^{10}\right)^T.$$

The permutation matrix $P_h^{(1,2)} \in \mathbb{R}^{16q\times 16q}$, which reorders the Fourier modes in the error vector for semi-coarsening in the $\bar{x}_2$-direction to that of $\widehat{e_h^{A,D}}$ for uniform coarsening consists of blocks of size $q\times q$. All blocks in the permutation matrix $P_h^{(1,2)}$ are zero, except blocks with the indices

$$(1,1),(2,4),(3,13),(4,16),(5,2),(6,3),(7,14),(8,15),(9,5),(10,8),(11,9),(12,12),(13,6),(14,7),(15,10),(16,11),$$

which are equal to the identity matrix I^q. The discrete Fourier transform of the error after one three-level multigrid cycle with semi-coarsening in the $\bar{x}_2$-direction can now be expressed as

$$\widehat{e_h^D}\left(\theta_\beta^\alpha\right) = \left(P_h^{(1,2)}\right)^{-1} \widehat{M}_h^{(1,2)}\left(\theta_{\beta_{(1,2)}}^{\alpha_{(1,2)}}\right) P_h^{(1,2)} \widehat{e_h^A}(\theta_\beta^\alpha).$$

5.5. Discrete Fourier transform of hp-MGS multigrid error transformation operator

The discrete Fourier transform of the error transformation operator $\widehat{M}_{h,3}$ of the *hp-MGS* algorithm for a polynomial order $p=3$ and three (semi)-coarsened mesh levels can be obtained by combining the results from the previous sections. The first part of the *hp-MGS* algorithm consists of p-multigrid. Since there is no coupling in p-multigrid between modes on different meshes the discrete Fourier transform of the p-multigrid part of the *hp-MGS* algorithm can be computed straightforwardly using the Fourier symbols discussed in Section 5.3, resulting in

$$\begin{aligned}\widehat{M}_{h,3}\left(\theta_\beta^\alpha\right) = {}& \left(\widehat{HU}_{h,3}\left(\theta_\beta^\alpha\right)\right)^{\gamma_2}\left(I^{16q_3} - \overline{T}_{h,2}^3\left(\theta_\beta^\alpha\right)\left(I^{16q_2} - \widehat{M}_{h,2}\left(\theta_\beta^\alpha\right)\right)\left(\overline{L}_{h,2}^{(2,2)}\left(\theta_\beta^\alpha\right)\right)^{-1}\overline{Q}_{h,3}^2\left(\theta_\beta^\alpha\right)\overline{L}_{h,3}^{(2,2)}\left(\theta_\beta^\alpha\right)\right)\left(\widehat{HU}_{h,3}\left(\theta_\beta^\alpha\right)\right)^{\gamma_1} \\ & \in \mathbb{C}^{16q_3\times 16q_3},\end{aligned} \tag{48}$$

with the contribution from the $p=2$ level given by

$$\widehat{M}_{h,2}\left(\theta_\beta^\alpha\right) = \left(\widehat{HU}_{h,2}\left(\theta_\beta^\alpha\right)\right)^{\gamma_2}\left(I^{16q_2} - \overline{T}_{h,1}^2\left(\theta_\beta^\alpha\right)\left(I^{16q_1} - \widehat{HU}_{h,1}\left(\theta_\beta^\alpha\right)\right)\left(\overline{L}_{h,1}^{(2,2)}\left(\theta_\beta^\alpha\right)\right)^{-1}\overline{Q}_{h,2}^1\left(\theta_\beta^\alpha\right)\overline{L}_{h,2}^{(2,2)}\left(\theta_\beta^\alpha\right)\right)\left(\widehat{HU}_{h,2}\left(\theta_\beta^\alpha\right)\right)^{\gamma_1} \in \mathbb{C}^{16q_2\times 16q_2},$$

where $\theta_\beta^\alpha = \theta_{\beta^1_{(2,2)}}^{\alpha^1_{(2,2)}}$. In this section we will use the shorthand notation $\alpha = \alpha^1_{(2,2)}$ and $\beta = \beta^1_{(2,2)}$. The superscript q_p, with $p = 1,2,3$, refers to the size of the blocks in the matrices of the space–time DG discretization using polynomial basis functions of order p. Using (24) the p-multigrid prolongation matrices $\overline{T}_{h,p}^{p+1}$ are defined as

$$\overline{T}_{h,p}^{p+1}\left(\theta_\beta^\alpha\right) = \mathrm{bdiag}\left(\widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_1}^{\alpha_1}\right), \ldots, \widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_1}^{\alpha_4}\right), \widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_2}^{\alpha_1}\right), \ldots, \widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_2}^{\alpha_4}\right), \widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_4}^{\alpha_1}\right), \ldots, \widehat{T_{h,p}^{p+1}}\left(\theta_{\beta_4}^{\alpha_4}\right)\right) \in \mathbb{C}^{16q_p\times 16q_p}$$

and the restriction matrices are equal to $\overline{Q}_{h,p+1}^p = \left(\overline{T}_{h,p}^{p+1}\right)^T$. Note, frequently the p-multigrid restriction and prolongation operators are purely element based in which case their discrete Fourier transform is independent of θ_β^α. The discrete Fourier transform of the *hp-MGS* error transformation operator depends on the three-level *h-MGS* smoothers $\widehat{HU}_{h,p}\left(\theta_\beta^\alpha\right)$, $p\in\{1,2,3\}$. These operators are obtained using the three-level analysis discussed in Section 5.4. In order to describe the discrete Fourier transform we extend the matrices defined in (33)–(36) and (37)–(45) to include also the polynomial order p of the basis functions used in the space–time discretization. Using the result for the three-level error transformation operator given by (47) we obtain the discrete Fourier transform of the three-level *h-MGS* error transformation operator for each polynomial order

$$\begin{aligned}\widehat{HU}_{h,p}\left(\theta_\beta^\alpha\right) = {}& \left(\widehat{HS}_{h,p}^{(2,1)}\left(\theta_\beta^\alpha\right)\widehat{HS}_{h,p}^{(1,2)}\left(\theta_\beta^\alpha\right)\right)^{\nu_2}\left(I^{16q_p} - \overline{P}_{2h,p}^h\left(\theta_\beta^\alpha\right)\widehat{\widetilde{M}}_{2h,p}^{(4,4)}\left(2\theta_\beta^{00}\right)\overline{Q}_{2h,p}^{(2,2)}\left(2\theta_\beta^{00}\right)\overline{R}_{h,p}^{2h}\left(\theta_\beta^\alpha\right)\overline{L}_{h,p}^{(2,2)}\left(\theta_\beta^\alpha\right)\right) \\ & \times\left(\widehat{HS}_{h,p}^{(1,2)}\left(\theta_\beta^\alpha\right)\widehat{HS}_{h,p}^{(2,1)}\left(\theta_\beta^\alpha\right)\right)^{\nu_1} \in \mathbb{C}^{16q_p\times 16q_p},\end{aligned} \tag{49}$$

with $\widehat{HS}^{(2,1)}_{h,p}\left(\theta^{\alpha}_{\beta}\right)$ and $\widehat{HS}^{(1,2)}_{h,p}\left(\theta^{\alpha}_{\beta}\right)$ the discrete Fourier transform of the error transformation operator of the semi-coarsening multigrid smoothers in, respectively, the local $\bar{x}_1$- and $\bar{x}_2$-direction. The coarse grid contribution $\widehat{\widetilde{M}}^{(4,4)}_{2h,p}\left(2\theta^{00}_{\beta}\right)$ from the mesh G_{2h} in (49) is given by

$$\widehat{\widetilde{M}}^{(4,4)}_{2h,p}\left(2\theta^{00}_{\beta}\right) = I^{4q_p} - \widehat{M}^{(4,4)}_{2h,p}\left(2\theta^{00}_{\beta}\right) \in \mathbb{C}^{4q_p\times 4q_p},$$

with

$$\begin{aligned}\widehat{M}^{(4,4)}_{2h,p}(2\theta^{00}_{\beta}) = &\left(\widehat{HS}^{(2,1)}_{2h,p}(2\theta^{00}_{\beta})\widehat{HS}^{(1,2)}_{2h,p}(2\theta^{00}_{\beta})\right)^{\nu_2}\left(I^{4q_p} - \widehat{P}^{2h}_{4h,p}(2\theta^{00}_{\beta})\left(\widehat{L_{4h,p}}(4\theta^{00}_{00})\right)^{-1}\widehat{R}^{4h}_{2h,p}(2\theta^{00}_{\beta})\widehat{L}^{(2,2)}_{2h,p}(2\theta^{00}_{\beta})\right)\\ &\times\left(\widehat{HS}^{(1,2)}_{2h,p}(2\theta^{00}_{\beta})\widehat{HS}^{(2,1)}_{2h,p}(2\theta^{00}_{\beta})\right)^{\nu_1} \in \mathbb{C}^{4q_p\times 4q_p}.\end{aligned}$$

The discrete Fourier transform of the semi-coarsening smoother in the local $\bar{x}_1$-direction is given by

$$\widehat{HS}^{(2,1)}_{h,p}\left(\theta^{\alpha}_{\beta}\right) = \left(P^{(2,1)}_{h}\right)^{-1}\mathrm{bdiag}\left(\widehat{M}^{(2,1)}_{h,p}\left(\theta^{\alpha^1_{(2,1)}}_{\beta^1_{(2,1)}}\right), \widehat{M}^{(2,1)}_{h,p}\left(\theta^{\alpha^2_{(2,1)}}_{\beta^1_{(2,1)}}\right), \widehat{M}^{(2,1)}_{h,p}\left(\theta^{\alpha^1_{(2,1)}}_{\beta^2_{(2,1)}}\right), \widehat{M}^{(2,1)}_{h,p}\left(\theta^{\alpha^2_{(2,1)}}_{\beta^2_{(2,1)}}\right)\right)P^{(2,1)}_{h} \in \mathbb{C}^{16q_p\times 16q_p},$$

with the permutation matrix $P^{(2,1)}_h \in \mathbb{C}^{16q_p\times 16q_p}$ defined in Section 5.4 and

$$\begin{aligned}\theta^{\alpha^1_{(2,1)}}_{\beta^1_{(2,1)}} &= \left(\theta^{00}_{00}, \theta^{10}_{00}, \theta^{00}_{\frac{1}{2}0}, \theta^{10}_{\frac{1}{2}0}\right)^T, \quad \theta^{\alpha^2_{(2,1)}}_{\beta^1_{(2,1)}} = \left(\theta^{11}_{00}, \theta^{01}_{00}, \theta^{11}_{\frac{1}{2}0}, \theta^{01}_{\frac{1}{2}0}\right)^T,\\ \theta^{\alpha^1_{(2,1)}}_{\beta^2_{(2,1)}} &= \left(\theta^{00}_{\frac{1}{2}\frac{1}{2}}, \theta^{10}_{\frac{1}{2}\frac{1}{2}}, \theta^{00}_{0\frac{1}{2}}, \theta^{10}_{0\frac{1}{2}}\right)^T, \quad \theta^{\alpha^2_{(2,1)}}_{\beta^2_{(2,1)}} = \left(\theta^{11}_{\frac{1}{2}\frac{1}{2}}, \theta^{01}_{\frac{1}{2}\frac{1}{2}}, \theta^{11}_{0\frac{1}{2}}, \theta^{01}_{0\frac{1}{2}}\right)^T.\end{aligned}$$

The discrete Fourier transform of the semi-coarsening smoother in the local $\bar{x}_2$-direction is

$$\widehat{HS}^{(1,2)}_{h,p}\left(\theta^{\alpha}_{\beta}\right) = \left(P^{(1,2)}_{h}\right)^{-1}\mathrm{bdiag}\left(\widehat{M}^{(1,2)}_{h,p}\left(\theta^{\alpha^1_{(1,2)}}_{\beta^1_{(1,2)}}\right), \widehat{M}^{(1,2)}_{h,p}\left(\theta^{\alpha^2_{(1,2)}}_{\beta^1_{(1,2)}}\right), \widehat{M}^{(1,2)}_{h,p}\left(\theta^{\alpha^1_{(1,2)}}_{\beta^2_{(1,2)}}\right), \widehat{M}^{(1,2)}_{h,p}\left(\theta^{\alpha^2_{(1,2)}}_{\beta^2_{(1,2)}}\right)\right)P^{(1,2)}_{h} \in \mathbb{C}^{16q_p\times 16q_p},$$

with the permutation matrix $P^{(1,2)}_h \in \mathbb{C}^{16q_p\times 16q_p}$ defined in Section 5.4 and

$$\begin{aligned}\theta^{\alpha^1_{(1,2)}}_{\beta^1_{(1,2)}} &= \left(\theta^{00}_{00}, \theta^{01}_{00}, \theta^{00}_{0\frac{1}{2}}, \theta^{01}_{0\frac{1}{2}}\right)^T, \quad \theta^{\alpha^2_{(1,2)}}_{\beta^1_{(1,2)}} = \left(\theta^{11}_{00}, \theta^{10}_{00}, \theta^{11}_{0\frac{1}{2}}, \theta^{10}_{0\frac{1}{2}}\right)^T,\\ \theta^{\alpha^1_{(1,2)}}_{\beta^2_{(1,2)}} &= \left(\theta^{00}_{\frac{1}{2}\frac{1}{2}}, \theta^{01}_{\frac{1}{2}\frac{1}{2}}, \theta^{00}_{\frac{1}{2}0}, \theta^{01}_{\frac{1}{2}0}\right)^T, \quad \theta^{\alpha^2_{(1,2)}}_{\beta^2_{(1,2)}} = \left(\theta^{11}_{\frac{1}{2}\frac{1}{2}}, \theta^{10}_{\frac{1}{2}\frac{1}{2}}, \theta^{11}_{\frac{1}{2}0}, \theta^{10}_{\frac{1}{2}0}\right)^T.\end{aligned}$$

Note, the permutation matrices are necessary in order to combine the error transformation operators for the different types of mesh coarsening which use a different ordering of the Fourier modes. The contribution to the error transformation operators from the different groups of modes in the semi-coarsening smoothers $\widehat{HS}^{(2,1)}_{h,p}\left(\theta^{\alpha}_{\beta}\right)$ and $\widehat{HS}^{(1,2)}_{h,p}\left(\theta^{\alpha}_{\beta}\right)$ is now given for $i,j\in s_n$ by

$$\widehat{M}^{n}_{h,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right) = \left(\overline{S}^{n}_{h,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)^{\mu_2}\left(I^{4q_p} - \overline{P}^{h}_{nh,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\widehat{\widetilde{M}}^{m}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\overline{Q}^{n}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\overline{R}^{nh}_{h,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\overline{L}^{n}_{h,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)\left(\overline{S}^{n}_{h,p}\left(\theta^{\alpha^i_n}_{\beta^j_n}\right)\right)^{\mu_1} \in \mathbb{C}^{4q_p\times 4q_p},$$

with the coarse grid contributions

$$\widehat{\widetilde{M}}^{m}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right) = I^{2q_p} - \widehat{M}^{m}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right) \in \mathbb{C}^{2q_p\times 2q_p}$$

and

$$\begin{aligned}\widehat{M}^{m}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right) = &\left(\widehat{S}^{n}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\right)^{\mu_2}\left(I^{2q_p} - \widehat{P}^{nh}_{mh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\right)\left(I^{q_p} - \left(\widehat{S_{mh,p}}\left(m\theta^{\gamma^i_n}_{\delta^j_n}\right)\right)^{\mu_3}\right)\\ &\times\left(\widehat{L_{mh,p}}\left(m\theta^{\gamma^i_n}_{\delta^j_n}\right)\right)^{-1}\widehat{R}^{mh}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\widehat{L}^{n}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\Bigg)\left(\widehat{S}^{n}_{nh,p}\left(n\theta^{\gamma^i_n}_{\beta^j_n}\right)\right)^{\mu_1} \in \mathbb{C}^{2q_p\times 2q_p},\end{aligned}$$

where $n=(2,1)$, $m=(4,1)$ for $HS^{(2,1)}_{h,p}$ and $n=(1,2)$, $m=(1,4)$ for $HS^{(1,2)}_{h,p}$. The smoother matrices $\overline{S}^n_{h,p}$, $\widehat{S}^n_{nh,p}$ are defined in Section 5.4 and $\widehat{S_{mh,p}}$ is the discrete Fourier symbol of the semi-implicit Runge–Kutta smoother discussed in Section 5.3.2, which is implicit in, respectively, the local $\bar{x}_1$-direction for $HS^{(2,1)}_{h,p}$ and in the local $\bar{x}_2$-direction for $HS^{(1,2)}_{h,p}$.

The contribution of the semi-coarsening smoothers at the mesh level $2h$ is equal to

$$\widehat{HS}^{n}_{2h,p}\left(2\theta^{00}_{\beta}\right) = (P^n_{2h})^{-1}\mathrm{bdiag}\left(\widehat{M}^{n}_{2h,p}\left(2\theta^{00}_{\beta^1_n}\right), \widehat{M}^{n}_{2h,p}\left(2\theta^{00}_{\beta^2_n}\right)\right)P^n_{2h} \in \mathbb{C}^{4q_p\times 4q_p},$$

with

$$\widehat{M}^n_{2h,p}\left(2\theta^{00}_{\beta^j_n}\right) = \left(\widehat{S}^n_{2h,p}\left(2\theta^{00}_{\beta^j_n}\right)\right)^{\mu_2}\left(I^{2q_p} - \widehat{P}^{2h}_{2nh,p}\left(2\theta^{00}_{\beta^j_n}\right)\left(I^{q_p} - \left(\widehat{S_{2nh,p}}\left(2n\theta^{00}_{\delta^j_n}\right)\right)^{\mu_3}\right)\left(\widehat{L_{2nh,p}}\left(2n\theta^{00}_{\delta^j_n}\right)\right)^{-1}\right.$$
$$\left.\times\, \widehat{R}^{2nh}_{2h,p}\left(2\theta^{00}_{\beta^j_n}\right)\widehat{L}^n_{2h,p}\left(2\theta^{00}_{\beta^j_n}\right)\right)\left(\widehat{S}^n_{2h,p}\left(2\theta^{00}_{\beta^j_n}\right)\right)^{\mu_1} \in \mathbb{C}^{2q_p\times 2q_p},$$

where $n=(2,1)$ for $HS^{(2,1)}_{2h,p}$ and $n=(1,2)$ for $HS^{(1,2)}_{2h,p}$. The permutation matrices are defined as

$$P^{(2,1)}_{2h} = \begin{pmatrix} I^q & 0 & 0 & 0 \\ 0 & 0 & I^q & 0 \\ 0 & I^q & 0 & 0 \\ 0 & 0 & 0 & I^q \end{pmatrix} \quad P^{(1,2)}_{2h} = \begin{pmatrix} I^q & 0 & 0 & 0 \\ 0 & 0 & 0 & I^q \\ 0 & I^q & 0 & 0 \\ 0 & 0 & I^q & 0 \end{pmatrix}.$$

6. Multilevel analysis of higher order space–time DG discretization

The multilevel Fourier analysis of the *hp-MGS* algorithm will be demonstrated for the advection–diffusion equation in two space-dimensions. The optimization of the Runge–Kutta smoother, including details about the Runge–Kutta coefficients, and the performance of the *hp-MGS* algorithm on a number of test cases is discussed in Part II [32].

The space–time discretization is made dimensionless by introducing the following dimensionless numbers, viz. the CFL number and the cell Reynolds number, defined as

$$CFL = \frac{|\,a\,|\,\Delta t}{|\,h\,|}, \quad Re_h = \frac{|\,a\,||\,h\,|}{\nu}, \tag{50}$$

with $|\,h\,|$ the diameter of the element and time step $\triangle t = t_{n+1} - t_n$.

For the multigrid analysis we assume a uniform mesh with mesh sizes h_1 and h_2 in the $\bar{x}_1$- and $\bar{x}_2$-direction, respectively. Furthermore, we introduce the mesh aspect ratio $A_h = \frac{h_2}{h_1}$, which implies that $|\,h\,| = h_1\sqrt{1+A_h^2}$, and the flow angle α with respect to the $\bar{x}_1$-axis, hence $a_1 = |\,a\,|\cos\alpha$ and $a_2 = |\,a\,|\sin\alpha$. On meshes with $h_1 \neq h_2$ we use two cell Reynolds numbers, viz. Re_{h_1} and Re_{h_2}, with h replaced by h_1 and h_2 in (50). The space–time DG discretization for the advection–diffusion Eq. (3) on a uniform mesh can be represented using the following dimensionless stencil notation:

$$L_{h,p} := L^a_{h,1} + L^a_{h,2} + L^d_{h,1} + L^d_{h,2} + L^t_h,$$
$$F_{h,p} := L^{t-1}_h U^{n-1}_h.$$

The inviscid part of the stencil only depends on the *CFL* number and can be written as:

$$L^a_{h,1} = CFL\sqrt{1+A_h^2}\cos\alpha \begin{bmatrix} & 0 & \\ L^a_1 & D^a_1 & 0 \\ & 0 & \end{bmatrix},$$
$$L^a_{h,2} = CFL\frac{\sqrt{1+A_h^2}}{A_h}\sin\alpha \begin{bmatrix} & 0 & \\ 0 & D^a_2 & 0 \\ & L^a_2 & . \end{bmatrix}$$

Note, we assume here that both components of the advection velocity a are positive. In case one or both coefficients are negative then the upwind direction in the stencil has to be adjusted accordingly. The viscous part of the stencil depends on the cell Reynolds and *CFL* numbers:

$$L^d_{h,1} := \frac{CFL}{Re_h}\begin{bmatrix} & 0 & \\ L^d_1 & D^d_1 & U^d_1 \\ & 0 & \end{bmatrix}, \quad L^d_{h,2} := \frac{CFL}{Re_h}\begin{bmatrix} & U^d_2 & \\ 0 & D^d_2 & 0 \\ & L^d_2 & \end{bmatrix}.$$

The stencils related to the time discretization are given by

$$L^t_h := \begin{bmatrix} & 0 & \\ 0 & D^t & 0 \\ & 0 & \end{bmatrix}, \quad L^{t-1}_h := \begin{bmatrix} & 0 & \\ 0 & D^{t-1} & 0 \\ & 0 & \end{bmatrix}.$$

The size m_p of the blocks $L^{a,d}_{1,2},\ D^{a,d}_{1,2},\ U^d_{1,2}, D^{t,t-1} \in \mathbb{R}^{m_p\times m_p}$ depends on the order of the polynomial basis functions used in the space–time DG discretization.

In the multilevel analysis we will first consider steady state problems. In general it is much harder to get good multigrid performance for steady state problems than for time-accurate problems. At steady state the contribution of L^t_h to the matrix

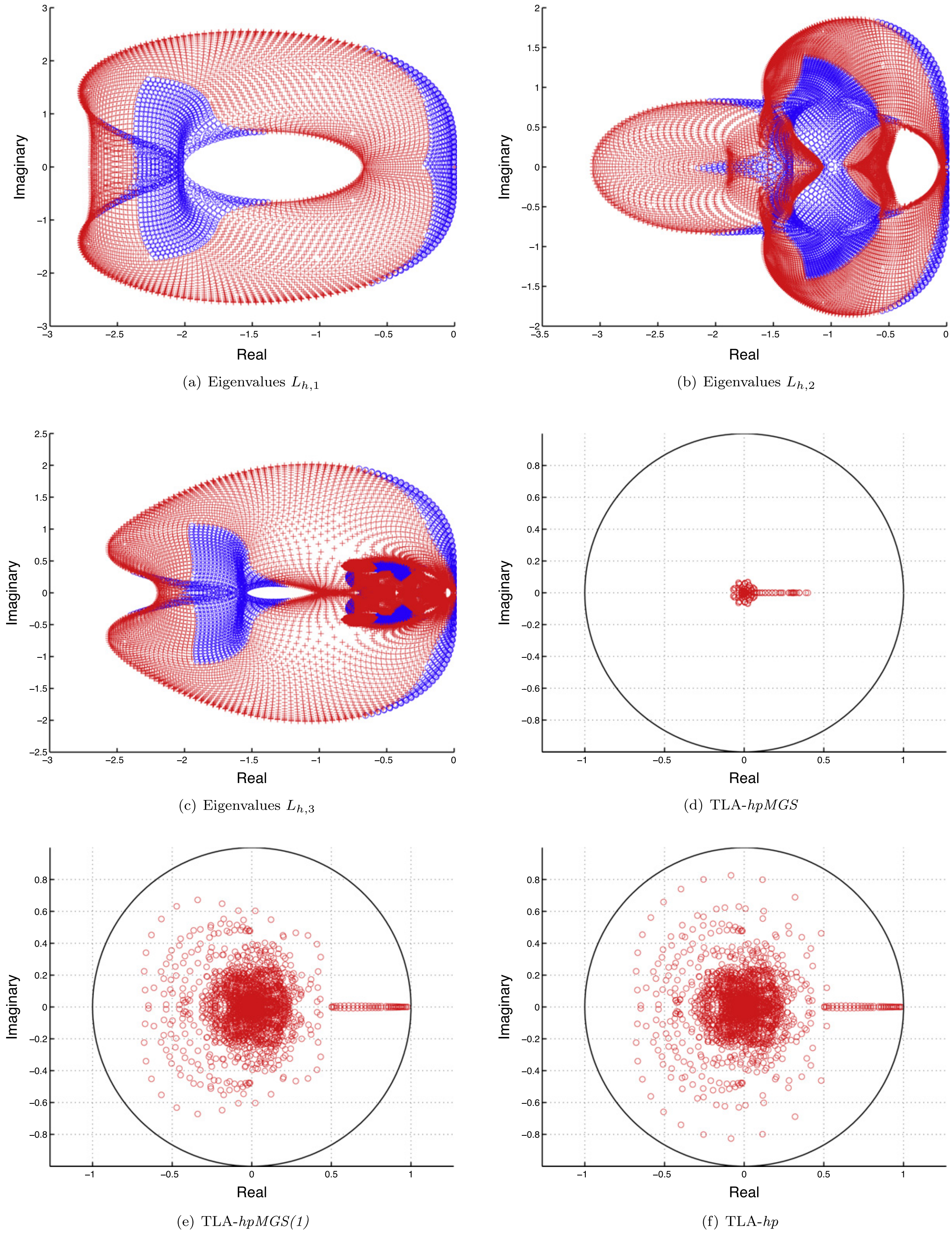
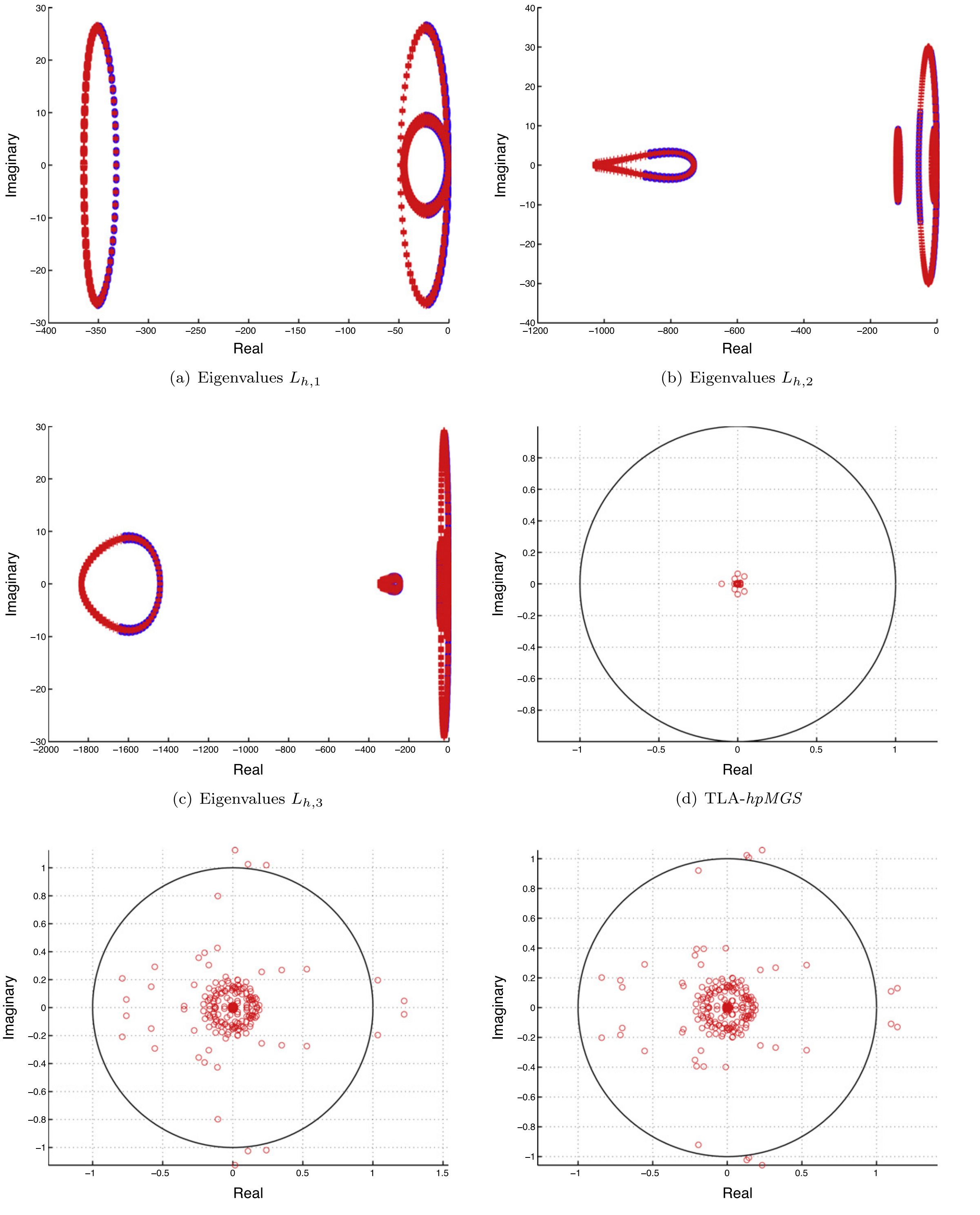


Fig. 6. Spectra of the DG matrices $L_{h,p}$ for polynomial orders $p = 1$, 2 and 3, and spectra of the error transformation operator of the full *hp-MGS* algorithm, and two simplifications, viz. the *hp-MGS*(1) and the *hp*-multigrid algorithms. Cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 10^3$, mesh aspect ratio $A_h = 1$, flow angle 45°, steady state.

L_h drops out and the matrix L_h does not depend on the CFL-number anymore. The parameters in the *hp-MGS* algorithm are $\gamma_1 = \gamma_2 = v_1 = v_2 = \mu_1 = \mu_2 = \mu_3 = 1$. The polynomial levels in the *hp-MGS* algorithm are $p = 1$, 2 and 3, and three uniformly



(a) Eigenvalues $L_{h,1}$

(b) Eigenvalues $L_{h,2}$

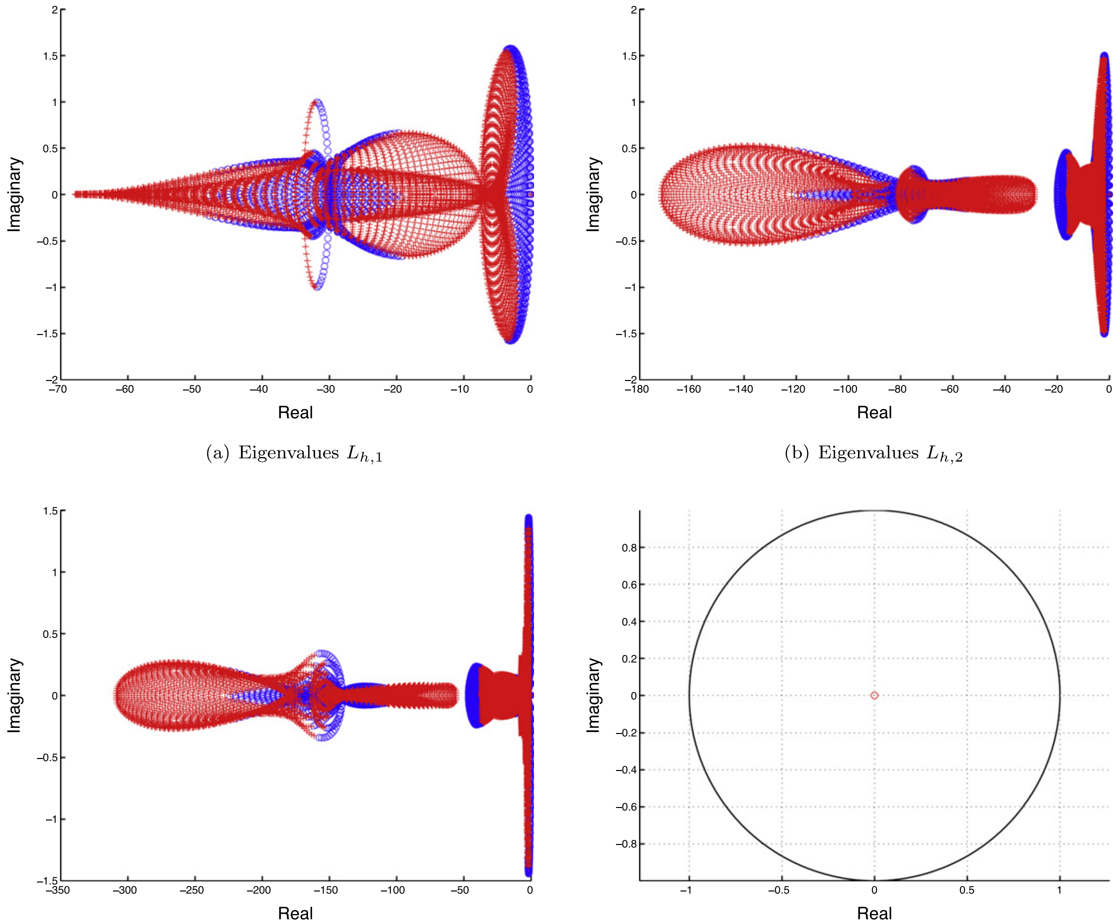
(c) Eigenvalues $L_{h,3}$

(d) TLA-$hpMGS$

(e) TLA-$hpMGS(1)$

(f) TLA-hp

Fig. 7. Spectra of the DG matrices $L_{h,p}$ for polynomial orders $p = 1$, 2 and 3, and spectra of the error transformation operator of the full *hp-MGS* algorithm, and two simplifications, viz. the *hp-MGS*(1) and the *hp*-multigrid algorithms. Cell Reynolds numbers $Re_{h_1} = 10^{-1}$, $Re_{h_2} = 10^{3}$, mesh aspect ratio $A_h = 100$, flow angle 75°, steady state.



(a) Eigenvalues $L_{h,1}$ (b) Eigenvalues $L_{h,2}$

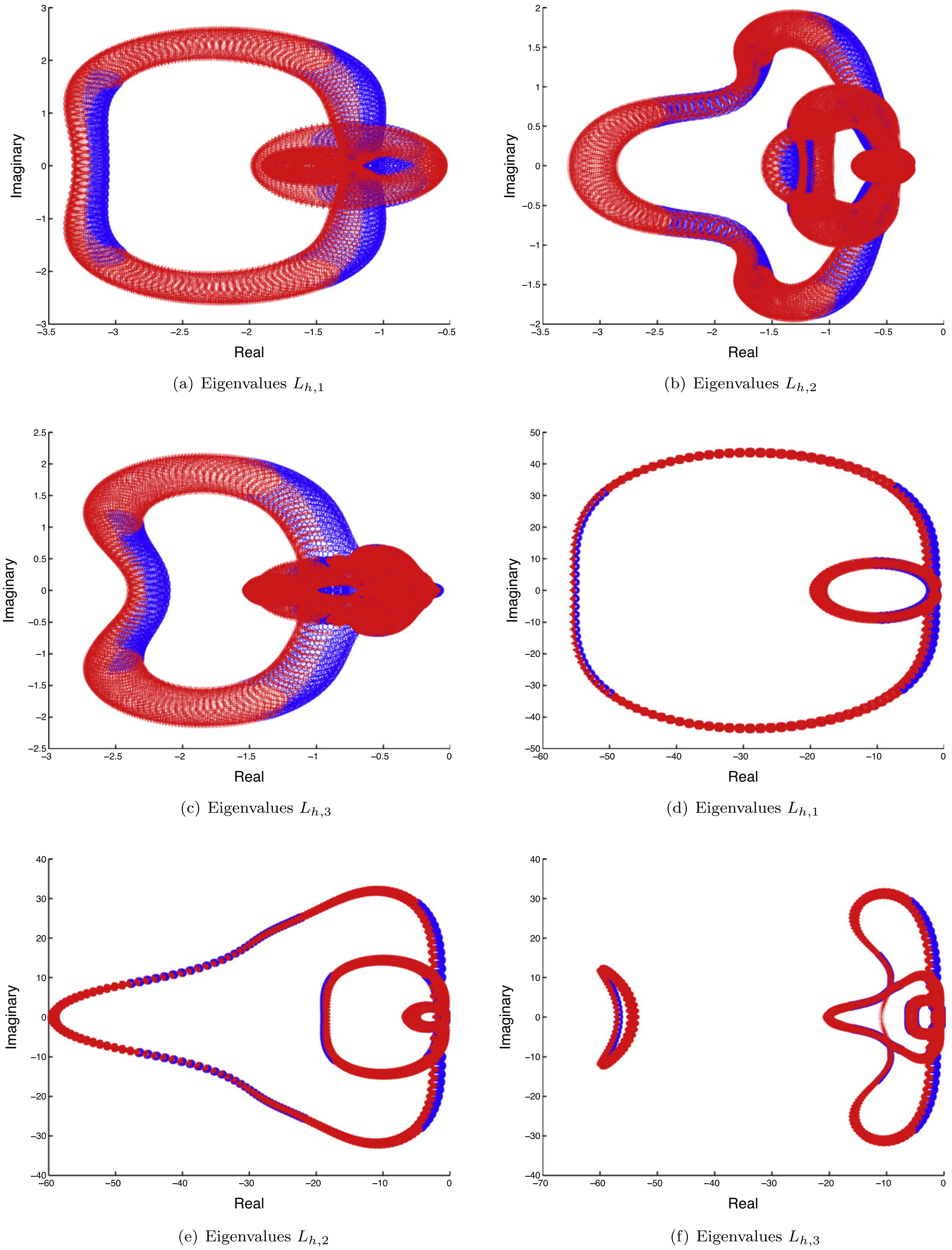
(c) Eigenvalues $L_{h,3}$ (d) TLA-$hpMGS$

Fig. 8. Spectra of the DG matrices $L_{h,p}$ for polynomial orders $p = 1$, 2 and 3, and spectra of the error transformation operator of the full *hp-MGS* algorithm. Cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 1$, mesh aspect ratio $A_h = 1$, flow angle 45°, steady state.

and three semi-coarsened meshes are used. In the multilevel analysis 64×64 Fourier modes are used. The Fourier code was also verified with a matrix analysis and both methods agreed up to machine accuracy. The matrix analysis is, however, computationally much more expensive.

The first test case we consider has cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 10^3$ on a mesh with aspect ratio $A_h = 1$. The flow angle is 45° with respect to the $\bar{x}_1$-axis. In Fig. 6(a)–(c) the spectrum is shown of the matrices $L_{h,1},\ L_{h,2}$ and $L_{h,3}$ representing, respectively, DG discretizations with polynomial orders $p = 1$, 2 and 3. The blue color refers to low frequency eigenvalues and the red color to high frequency eigenvalues relative to a uniformly coarsened mesh. The eigenvalues of the full *hp-MGS* algorithm are shown in Fig. 6(d). The spectral radius of the error transformation operator $M_{h,3}$ is 0.4000 and the operator norm $\|M_{h,3}\| = 0.7379$, which shows that the *hp-MGS* algorithm is an efficient algorithm for higher order accurate DG discretizations of advection dominated flows. We also consider two simplifications of the *hp-MGS* algorithm. The first simplification, denoted *hp-MGS*(1), is to use the semi-implicit Runge–Kutta smoother at the $p = 2$ and 3 levels and only use the *h-MGS* algorithm at the $p = 1$ level. The eigenvalues of this algorithm are shown in Fig. 6(e). The spectral radius of the error transformation operator is 0.9781 and the operator norm 1.645. The second simplification is standard *hp*-multigrid, with *h*-multigrid with uniform coarsening at the $p = 1$ level and the semi-implicit Runge–Kutta method as smoother. The eigenvalues of this algorithm are shown in Fig. 6(f). The spectral radius of the error transformation operator is 0.9856 and the operator norm 1.645. The multigrid performance of the *hp-MGS*(1) and the *hp*-multigrid algorithm is very poor and the algorithms are not suitable for higher order accurate DG discretizations, despite extensive optimization of the semi-implicit Runge–Kutta smoother. A more detailed comparison of the different multigrid algorithms and their computational cost will be given in Part II [32].

Next, we consider the multigrid performance on stretched meshes, which are essential to deal with thin boundary layers. We consider cell Reynolds numbers $Re_{h_1} = 10^{-1}$ and $Re_{h_2} = 10^3$ on a mesh with aspect ratio $A_h = 100$. The same Runge–Kutta coefficients as in the previous test case with $Re_h = 10^3$ and $A_h = 1$ were used. The flow angle is 75° with respect to the $\bar{x}_1$-axis. In Fig. 7(a)–(c) the spectrum is shown of the DG matrices $L_{h,1},\ L_{h,2}$ and $L_{h,3}$. The spectra have a distinctly different pattern



(a) Eigenvalues $L_{h,1}$

(b) Eigenvalues $L_{h,2}$

(c) Eigenvalues $L_{h,3}$

(d) Eigenvalues $L_{h,1}$

(e) Eigenvalues $L_{h,2}$

(f) Eigenvalues $L_{h,3}$

Fig. 9. Spectra of the DG matrices $L_{h,p}$ for polynomial orders $p = 1, 2$ and 3. Mesh aspect ratio $A_h = 1$ and cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 10^5$ in (a)–(c). Mesh aspect ratio $A_h = 100$ and cell Reynolds numbers $Re_{h_1} = 10^{-1},\ Re_{h_2} = 10^3$ in (d)–(f). Flow angle 75°, CFL=1.

than the spectra in Fig. 6(a)–(c), which are for a mesh aspect ratio $A_h = 1$. The spectra in Fig. 7(a)–(c) split into two or three parts and the largest eigenvalue is close to -2000. These strongly negative eigenvalues are hard to deal with for explicit smoothers. The multigrid performance of the *hp-MGS* algorithm, shown in Fig. 7(d), is excellent, with a spectral radius of the error transformation operator $M_{h,3}$ of 0.1021 and the operator norm 0.3093. Both the *hp-MGS*(1) and the *hp*-multigrid algorithms are, however, unstable as can be seen in Fig. 7(e)–(f). The spectral radius of the error transformation operator of the *hp-MGS*(1) algorithm is 1.224 and the operator norm 1.950. The spectral radius of the error transformation operator of the *hp*-multigrid algorithm is 1.148 and the operator norm 1.950. This shows that the use of a semi-implicit Runge–Kutta smoother is not sufficient to ensure good multigrid performance on highly stretched meshes and that the semi-coarsening multigrid smoother is also important. Also, increasing the number of smoother iterations does not improve the convergence rate for the *hp-MGS*(1) and *hp*-multigrid algorithms.

At low Reynolds numbers the difference between the different multigrid algorithms is small. We consider the cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 1$ on a mesh with aspect ratio $A_h = 1$. The flow angle is 45° with respect to the $\bar{x}_1$-axis. In Fig. 8(a)–(c) the spectrum is shown of the matrices $L_{h,1}, L_{h,2}$ and $L_{h,3}$. Compared to the cell Reynolds number $Re_h = 10^3$, shown in Fig. 6(a)–(c), the spectra look very different. The multigrid performance of all three algorithms for this low cell Reynolds number is, however, excellent. The spectral radius for the *hp-MGS, hp-MGS*(1) and *hp*-multigrid algorithms is, respectively, 2.537×10^{-7}, 3.743×10^{-2} and 8.715×10^{-2}. The operator norms are 2.976×10^{-7}, 9.469×10^{-2} and 1.247×10^{-1}. The excellent convergence rate at this low cell Reynolds number can be attributed to the use of the semi-implicit Runge–Kutta smoother.

Finally, we consider time dependent problems. The space–time discretization now uses cubic polynomials, both in space and time. We select a CFL number of 1 since this is a natural choice for time-accurate simulations. The first test case has cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 10^5$ on a mesh with aspect ratio $A_h = 1$. The spectra are shown in Fig. 9(a)–(c). The second test case has cell Reynolds numbers $Re_{h_1} = 10^1$ and $Re_{h_2} = 10^5$ on a mesh with aspect ratio $A_h = 100$. The flow angle for both cases is 75°. In the computations the same Runge–Kutta coefficients are used as for the steady state problem with mesh aspect ratio $A_h = 1$ and $Re_{h_1} = Re_{h_2} = 10^5$. The spectra are shown in Fig. 9(d)–(f). The main feature of the spectra for time-accurate problems is that the eigenvalues are in the left half of the complex plane, away from the imaginary axis. This makes it much easier to get good multigrid performance, than when the eigenvalues are close to zero. The spectral radius and operator norm for the cell Reynolds numbers $Re_{h_1} = Re_{h_2} = 10^5$ are, respectively, 2.121×10^{-18} and 2.268×10^{-18}. For the cell Reynolds numbers are $Re_{h_1} = 10^1$ and $Re_{h_2} = 10^5$ the spectral radius is 2.950×10^{-17} and the operator norm 1.575×10^{-16}. Both cases show an extremely fast convergence, which is also confirmed by multigrid computations. This very fast convergence for time-accurate problems is observed for a very large range of cell Reynolds numbers, ranging from 1 to 10^9.

7. Conclusions and outlook

The *hp*-Multigrid as Smoother algorithm for the efficient solution of higher order accurate discretizations of advection dominated flows was presented. This algorithm uses a V-cycle *p*-multigrid algorithm with *h*-multigrid as smoother at all polynomial levels. The performance of the *h*-multigrid algorithm is further enhanced using semi-coarsening and a semi-implicit Runge–Kutta smoother. The discrete Fourier multilevel analysis of the complete *hp-MGS* algorithm was discussed for three *p*-levels and three uniformly and three semi-coarsened meshes. This analysis provides the operator norm and spectral radius of the error transformation operator of the *hp-MGS* algorithm. This information is important both to obtain realistic estimates of the multigrid performance and to optimize the multigrid algorithm, which will be discussed in Part II [32]. The analysis of the *hp-MGS* algorithm is demonstrated for algebraic systems resulting from a fourth order accurate space–time DG discretization of the two-dimensional advection–diffusion equation for various cell Reynolds numbers and mesh aspect ratios. The multilevel analysis shows that the new *hp-MGS* algorithm has excellent convergence rates for a wide range of cell Reynolds numbers, both on uniform and stretched meshes and for steady and time-dependent problems. The *hp-MGS*(1) and the standard *hp*-multigrid algorithm do not perform well for high cell Reynolds numbers despite extensive optimization conducted in Part II [32]. At low cell Reynolds numbers, the *hp-MGS*, *hp-MGS*(1) and the *hp*-multigrid algorithms converge well.

In Part II [32] the multilevel analysis discussed in this article will be used to optimize the semi-implicit Runge–Kutta smoother and extensive tables with Runge–Kutta coefficients and multigrid convergence rates will be presented. In addition, an analysis of the computational complexity of the *hp-MGS* algorithm and several simplifications will be discussed. Also, the *hp-MGS* algorithm will be tested on a number of model problems, including thin boundary layers and highly stretched meshes.

Acknowledgments

Part of this research was conducted while van der Vegt was visiting Professor at the Division of Applied Mathematics of Brown University, Providence, RI, USA. The kind hospitality of Prof. Shu is greatly appreciated. The research of Rhebergen was partly funded by the EU sixth framework ADIGMA project and a Rubicon fellowship from the Netherlands Organisation for Scientific Research (NWO) and the Marie Curie Cofund Action.

References

[1] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, J. Comput. Phys. 131 (1997) 267–279.

[2] F. Bassi, A. Ghidoni, S. Rebay, P. Tesini, High-order accurate *p*-multigrid discontinuous Galerkin solution of the Euler equations, Int. J. Numer. Meth. Fluids 60 (2009) 847–865.
[3] F. Bassi, A. Ghidoni, S. Rebay, Optimal Runge–Kutta smoother for the *p*-multigrid discontinuous Galerkin solution of the 1D Euler equations, J. Comput. Phys. 230 (2011) 4153–4175.
[4] A. Brandt, Rigorous quantitative analysis of multigrid I: constant coefficients two-level cycle with L_2-norm, SIAM J. Numer. Anal. 31 (1994) 1695–1730.
[5] F. Brezzi, G. Manzini, D. Marini, P. Pietra, A. Russo, Discontinuous Galerkin approximations for elliptic problems, Numer. Meth. Part. Diff. Eq. 16 (2000) 365–378.
[6] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (3) (2001) 173–261.
[7] L.T. Diosady, D.L. Darmofal, Preconditioning methods for discontinuous Galerkin solutions of the Navier–Stokes equations, J. Comput. Phys. 228 (2009) 3917–3935.
[8] K.J. Fidkowski, T.A. Oliver, J. Lu, L. Darmofal, *p*-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 207 (2005) 92–113.
[9] J. Gopalakrishnan, G. Kanschat, A multilevel discontinuous Galerkin method, Numer. Math. 95 (2003) 527–550.
[10] W. Hackbusch, Multigrid Methods and Applications, Springer Verlag, Berlin, 1985.
[11] W. Hackbusch, U. Trottenberg, Multigrid methods: proceedings of the conference held at Köln-Porz, November 23–27, 1981, Springer-Verlag, Berlin, 1982.
[12] J.S. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods. Algorithms, Analysis and Applications, Springer Verlag, New York, 2008.
[13] A. Jameson, Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings, AIAA paper (1991) 91–1596.
[14] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin method for the compressible Navier–Stokes equations, J. Comput. Phys. 217 (2006) 589–611.
[15] C.M. Klaij, J.J.W. van der Vegt, H. van der Ven, Pseudo-time stepping methods for space–time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 219 (2006) 622–643.
[16] C.M. Klaij, M.H. van Raalte, H. van der Ven, J.J.W. van der Vegt, *h*-multigrid for space–time discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 227 (2007) 1024–1045.
[17] H. Luo, J.D. Baum, R. Löhner, A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, J. Comput. Phys. 211 (2006) 767–783.
[18] B.S. Mascarenhas, B.T. Helenbrook, H.L. Atkins, Application of *p*-multigrid to discontinuous Galerkin formulations of the Euler equations, AIAA J. 47 (2009) 1200–1208.
[19] B.S. Mascarenhas, B.T. Helenbrook, H.L. Atkins, Coupling *p*-multigrid to geometric multigrid for discontinuous Galerkin formulations of the convection–diffusion equation, J. Comput. Phys. 229 (2010) 3664–3674.
[20] N.D. Melson, M.D. Sanetrik, H.L. Atkins, Time-accurate Navier–Stokes calculations with multigrid acceleration, in: Proc. 6th Copper Mountain Confer. on Multigrid Methods, pp. 423–437, 1993 (NASA Langley Research Center, 1993).
[21] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an *hp*-multigrid approach, J. Comput. Phys. 213 (2006) 330–357.
[22] C.W. Oosterlee, The convergence of parallel multiblock multigrid methods, Appl. Numer. Math. 19 (1995) 115–128.
[23] P.-O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier–Stokes equations, SIAM J. Sci. Comput. 30 (6) (2008) 2709–2733.
[24] M.H. van Raalte, P.W. Hemker, Two-level multigrid analysis for the convection–diffusion equation discretized by a discontinuous Galerkin method, Numer. Linear Algebra Appl. 12 (2005) 563–584.
[25] S. Rhebergen, J.J.W. van der Vegt, H. van der Ven, Multigrid optimization for space–time discontinuous Galerkin discretizations of advection dominated flows, in: ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications. Notes on Numerical Fluid Mechanics and Multidisciplinary Design 113 Springer-Verlag, Berlin, (2010) pp. 257–269.
[26] K. Shahbazi, D.J. Mavriplis, N.K. Burgess, Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, J. Comput. Phys. 228 (2009) 7917–7940.
[27] J.J. Sudirham, J.J.W. van der Vegt, R.M.J. van Damme, Space–time discontinuous Galerkin method for advection–diffusion problems on time-dependent domains, Appl. Numer. Math. 56 (2006) 1491–1518.
[28] U. Trottenberg, C.W. Oosterlee, A. Schüller, Multigrid, Academic Press, London, 2001.
[29] J.J.W. van der Vegt, H. van der Ven, Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows I. general formulation, J. Comput. Phys. 182 (2002) 546–585.
[30] J.J.W. van der Vegt, Y. Xu, Space–time discontinuous Galerkin method for nonlinear water waves, J. Comput. Phys. 224 (1) (2007) 17–39.
[31] J.J.W. van der Vegt, S. Rhebergen, Discrete fourier analysis of multigrid algorithms, Memorandum Department of Applied Mathematics, University of Twente, Enschede, The Netherlands, (2011), <http://eprints.eemcs.utwente.nl/>.
[32] J.J.W. van der Vegt, S. Rhebergen, *HP*-Multigrid as Smoother algorithm for higher order discontinuous Galerkin discretizations of advection dominated flows. Part II, optimization of the Runge–Kutta smoother, J. Comput. Phys., 2012, http://dx.doi.org/10.1016/j.jcp.2012.05.037
[33] H. van der Ven, An adaptive multitime multigrid algorithm for time-periodic flow simulations, J. Comput. Phys. 227 (2008) 5286–5303.
[34] T. Washio, C.W. Oosterlee, Flexible multiple semicoarsening for three-dimensional singularly perturbed problems, SIAM J. Sci. Comput. 19 (5) (1998) 1646–1666.
[35] P. Wesseling, An Introduction to Multigrid Methods, Wiley, Chicester, 1991.
[36] R. Wienands, C.W. Oosterlee, On three-grid Fourier analysis for multigrid, SIAM J. Sci. Comput. 23 (2001) 651–671.
[37] R. Wienands, W. Joppich, Practical Fourier Analysis for Multigrid Methods, Chapman & Hall, CRC, 2005.