HP-MULTIGRID AS SMOOTHER ALGORITHM FOR HIGHER ORDER DISCONTINUOUS GALERKIN DISCRETIZATIONS OF ADVECTION DOMINATED FLOWS

PART II. OPTIMIZATION OF THE RUNGE-KUTTA SMOOTHER

J.J.W. VAN DER VEGT* AND S. RHEBERGEN†

AMS subject classifications. Primary 65M55; Secondary 65M60, 76M10.

Abstract. Using a detailed multilevel analysis of the complete *hp*-Multigrid as Smoother algorithm accurate predictions are obtained of the spectral radius and operator norms of the multigrid error transformation operator. This multilevel analysis is used to optimize the coefficients in the semi-implicit Runge-Kutta smoother, such that the spectral radius of the multigrid error transformation operator is minimal under properly chosen constraints. The Runge-Kutta coefficients for a wide range of cell Reynolds numbers and a detailed analysis of the performance of the *hp-MGS* algorithm are presented. In addition, the computational complexity of the *hp-MGS* algorithm is investigated. The *hp-MGS* algorithm is tested on a fourth order accurate space-time discontinuous Galerkin finite element discretization of the advection-diffusion equation for a number of model problems, which include thin boundary layers and highly stretched meshes, and a non-constant advection velocity. For all test cases excellent multigrid convergence is obtained.

Key words. multigrid algorithms, discontinuous Galerkin methods, higher order accurate discretizations, space-time methods, Runge-Kutta methods, Fourier analysis, multilevel analysis

1. Introduction. In [16], subsequently called Part I, we introduced the *hp*-Multigrid as Smoother algorithm as a new multigrid method for the solution of algebraic systems resulting from higher order accurate finite element discretizations of partial differential equations. Using discrete Fourier multilevel analysis the multigrid performance of the full *hp-MGS* algorithm was analyzed for two-dimensional problems. An important component of the *hp-MGS* algorithm is the semi-implicit Runge-Kutta smoother, which contains a number of free coefficients. The multilevel analysis of the *hp-MGS* algorithm, discussed in Part I, gives the opportunity to optimize the multigrid efficiency by computing Runge-Kutta coefficients such that the spectral radius and operator norms of the full *hp-MGS* algorithm are minimal for a given class of problems.

In this article we are particularly interested in improving the multigrid performance for higher order accurate space-time discontinuous Galerkin discretizations of advection dominated flows. Discontinuous Galerkin finite element methods have received significant attention during the past decade and provide stable and robust discretizations for large classes of partial differential equations. In particular, due to their local structure they are well suited for parallel computing and *hp*-mesh adaptation, where p refers to adjustment of the polynomial order of the basis functions and h to local mesh refinement and coarsening. For an overview of various aspects of DG methods, see e.g. [2, 6]. During the past decade also extensive research into efficient multigrid algorithms for discontinuous Galerkin discretizations has been conducted, see e.g. [1, 3, 8, 9, 10, 11]. This research is motivated by the strong need in industrial applications to reduce the computational cost of DG methods. Both p-, h-, and

*Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands (j.j.w.vandervegt@math.utwente.nl)

†Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands. Current address School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA (srheberg@umn.edu)

hp-multigrid methods have been investigated. In *p*-multigrid the coarser levels are obtained using a sequence of lower order discretizations, whereas in *h*-multigrid coarser meshes are used. Combinations of both methods result in *hp*-multigrid. For a more detailed overview of multigrid methods for discontinuous Galerkin discretizations, see Part I and the references therein.

For the multigrid optimization we consider the advection-diffusion equation as model problem. Using the discrete Fourier multilevel analysis discussed in Part I, we can compute the operator norms and spectral radius of the full *hp-MGS* algorithm for a space-time discontinuous Galerkin discretization with periodic boundary conditions. The operator norms give an upper bound for the reduction of the error and residual after one application of the *hp-MGS* algorithm and the spectral radius the asymptotic convergence rate of the multigrid algorithm. The use of discrete Fourier multilevel analysis to optimize multigrid performance is very old and can be found in many multigrid textbooks, [5, 13, 17, 18]. Frequently the optimization is done using analytical tools, but the computation of the Fourier symbol of the full *hp-MGS* algorithm is too complicated for a purely analytic approach. We therefore wrote a Matlab code to conduct this analysis and use the fmincon Matlab optimization function for the optimization of the Runge-Kutta smoother. This approach gives optimal multigrid algorithms for large classes of problems which could not be obtained with other techniques. Since a number of important simplifications have to be made in the discrete Fourier multilevel analysis, such as periodic boundary conditions and constant coefficients, we also extensively test the new multigrid algorithms on problems with very thin boundary layers, which require locally highly stretched meshes, and problems with non-constant coefficients.

For an efficient multigrid algorithm not only a fast convergence rate is important, but also its computational cost. In order to assess this we consider the computational complexity of the *hp-MGS* algorithm and compare this with the computational complexity of several simplifications. The first simplified scheme uses only the semi-implicit Runge-Kutta method as smoother at the $p = 2$ and 3 levels and semi-coarsening *h*-multigrid at the $p = 1$ level. The second simplification is to replace also the semi-coarsening multigrid at the $p = 1$ level with standard *h*-multigrid using uniformly coarsened meshes.

The outline of this article is as follows. First, we briefly discuss in Section 2 the advection-diffusion model problem used for the multigrid optimization. Next, we summarize in Section 3 the *hp-MGS* algorithm, the semi-implicit Runge-Kutta smoother and the multigrid error transformation operator. The computational complexity of the *hp-MGS* algorithm is analyzed in Section 4 and the multigrid smoothers are optimized in Section 5 using the multilevel Fourier analysis discussed in Part I. This analysis provides the theoretical performance of the *hp-MGS* algorithm and the two simplifications thereof discussed earlier. The practical performance of the *hp-MGS* algorithm on realistic model problems for advection dominated flows, which include thin boundary layers and a non-constant advection velocity, will be discussed in Section 6. Finally, conclusions are drawn in Section 7.

2. Space-time DG discretization of the advection-diffusion equation. As model problem we consider a space-time discontinuous Galerkin discretization of the advection-diffusion equation in two space dimensions. In a space-time DG formulation, the space and time variables are discretized simultaneously. A point at time $t = x_0$, with position vector $\bar{x} = (\bar{x}_1, \bar{x}_2) \in \mathbb{R}^2$, has Cartesian coordinates $x = (x_0, \bar{x})$ in the open domain $\mathcal{E} = \Omega \times (t_0, T) \subset \mathbb{R}^3$, with t_0 and T the initial and

final time of the solution and $\Omega \subset \mathbb{R}^2$ the spatial domain. 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, $a \in \mathbb{R}^2$ the advection velocity, and $\nabla = (\frac{\partial}{\partial \bar{x}_1}, \frac{\partial}{\partial \bar{x}_2}) \in \mathbb{R}^2$ 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 details of the space-time discontinuous Galerkin discretization for the advection-diffusion equation can be found in Part I [16].

The space-time DG discretization for the advection-diffusion equation results at each time level t_n in a linear system

$$L_h U_h^n = f_h, \tag{2.1}$$

for the DG coefficients U_h^n in each element, with L_h the discretization matrix and f_h the righthand side, which depends on the known DG coefficients U_h^{n-1} from the previous time level.

For the multigrid analysis we will assume a uniform mesh with mesh sizes h_1 and h_2 in the $\bar{x}_1$- and $\bar{x}_2$-direction, respectively, and periodic boundary conditions. Furthermore, we introduce the mesh aspect ratio $A_h = \frac{h_2}{h_1}$, which implies that element diameter is equal to $|h| = h_1\sqrt{1 + A_h^2}$, and the flow angle α with respect to the $\bar{x}_1$-axis, hence $a_1 = |a| \cos\alpha$ and $a_2 = |a| \sin\alpha$. The space-time discretization is made dimensionless by introducing the following dimensionless numbers, viz. the CFL number and the cell Reynolds number, defined as

$$CFL = \frac{|a|\Delta t}{|h|}, \quad Re_h = \frac{|a||h|}{\nu}, \tag{2.2}$$

with time step $\triangle t = t_{n+1} - t_n$. 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 (2.2).

3. Multigrid Algorithm.

3.1. *hp*-Multigrid as Smoother Algorithm. In this section we summarize the *hp*-Multigrid as Smoother algorithm, which we presented in Part I [16] as a new multigrid algorithm for the solution of algebraic systems resulting from higher order accurate finite element discretizations of partial differential equations. The *hp-MGS* algorithm consists of three steps. First, a V-cycle *p*-multigrid algorithm is combined with *h*-multigrid, which acts as smoother in the *p*-multigrid at each polynomial level *p*, see Figure 3.1. Next, the *h*-multigrid algorithm uses a semi-coarsening multigrid algorithm as smoother at each uniformly coarsened mesh, see Figure 3.2. Finally, the semi-coarsening multigrid algorithm uses a semi-implicit Runge-Kutta method as smoother.

The *hp-MGS* algorithm is defined in Algorithms 1, 2 and 3. 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* smoother $HU_{nh,p}$ defined in Algorithm 2. In Algorithm 1 the linear system at each grid and polynomial level is

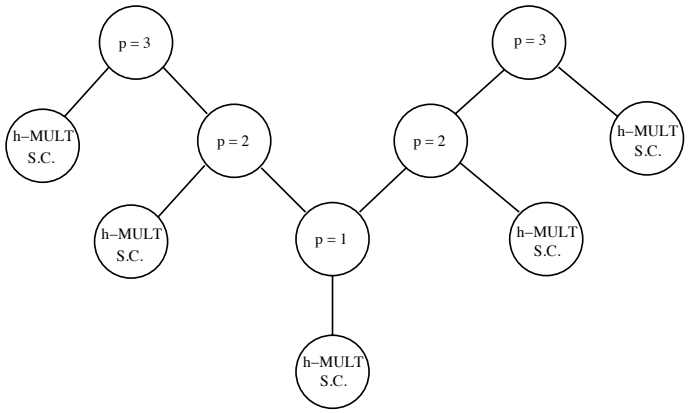


FIG. 3.1. *hp-MGS algorithm combining p-multigrid and the h-Multigrid as Smoother algorithm at each polynomial level. The h-Multigrid as Smoother algorithm uses semi-coarsening in the local* $\bar{x}_1$*- and* $\bar{x}_2$*-directions and a semi-implicit Runge-Kutta method.*

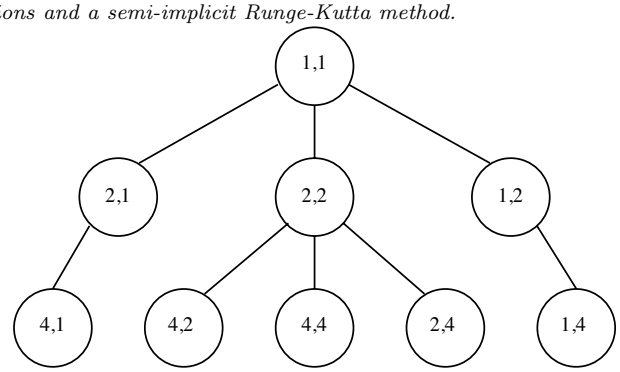


FIG. 3.2. *h-Multigrid as Smoother algorithm used at each polynomial level p as smoother in the hp-MGS algorithm. The indices refer to grid coarsening. Mesh* $(1,1)$ *is the fine mesh and e.g. Mesh* $(4,1)$ *has mesh size* $(4h_1, h_2)$*.*

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 $h = (h_1, h_2)$, with h_1 and h_2 the mesh size in the different local coordinate directions. The mesh coarsening is indicated by the integer $n = (n_1, n_2)$, hence $nh := (n_1 h_1, n_2 h_2)$. 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-relaxations 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 transposed of the prolongation operators, viz. $Q^{p-1}_{nh,p} = (T^p_{nh,p-1})^T$.

In the $HU_{nh,p}$-multigrid algorithm, defined recursively in Algorithm 2, the semi-coarsening multigrid algorithm $HS^i_{nh,p}$, $i = 1, 2$, is used as smoother in the local i-direction. The restriction of the data from the mesh $\mathcal{M}_{nh}$ to the mesh $\mathcal{M}_{mh}$, with $m_1 \geq n_1$ and $m_2 \geq n_2$, is indicated by the restriction operators $R^{mh}_{nh,p}$. The

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 $it = 1, \cdots, \gamma_1$ **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
$r_{nh,p} := f_{nh,p} - L_{nh,p} v_{nh,p}$;
$f_{nh,p-1} := Q^{p-1}_{nh,p} r_{nh,p}$;
$v_{nh,p-1} := 0$;
$v_{nh,p-1} := HP_{nh,p}(L_{nh,p-1}, f_{nh,p-1}, v_{nh,p-1}, n, p-1, \gamma_1, \gamma_2, \nu_1, \nu_2, \mu_1, \mu_2, \mu_3)$;
// lower order polynomial correction
$v_{nh,p} := v_{nh,p} + T^p_{nh,p-1} v_{nh,p-1}$;
// post-smoothing with *h-MGS* algorithm
for $it = 1, \cdots, \gamma_2$ **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
}

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 $R^{mh}_{nh,p} = (P^{nh}_{mh,p})^T/(n_1 n_2)$.

The semi-coarsening h-multigrid smoothers $HS^i_{nh,p}$, $i = 1, 2$, 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 discussed in detail in Section 3.2. At the coarsest levels in the semi-coarsened meshes we use μ_3 smoother iterations.

Different 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 Runge-Kutta smoother. As multigrid smoother we use in Algorithm 3 at each polynomial level a semi-implicit Runge-Kutta pseudo-time integration method, which we briefly summarize. More details can be found in Part I. First, the linear system is augmented with a pseudo-time derivative, which is integrated to steady-state in pseudo-time

$$\frac{\partial v^*_{nh,p}}{\partial \sigma} = -\frac{1}{\triangle t}(L_{nh,p} v^*_{nh,p} - f_{nh,p}). \tag{3.1}$$

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 $it = 1, \cdots, \nu_1$ **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 $it = 1, \cdots, \nu_2$ **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
}

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.

The system of ordinary differential equations (3.1) is now solved using a semi-implicit Runge-Kutta method. Since, the *hp-MGS* algorithm uses semi-coarsening in the local i_1- and i_2-directions of each element we split the matrix $L_{nh,p}$ for sweeps in the i_1-direction, as

$$L_{nh,p} = L^{i_{11}}_{nh,p} + L^{i_{12}}_{nh,p}, \tag{3.2}$$

and for sweeps in the i_2-direction as

$$L_{nh,p} = L^{i_{21}}_{nh,p} + L^{i_{22}}_{nh,p}. \tag{3.3}$$

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

Algorithm 3 Semi-coarsening Multigrid Algorithm ($HS^i_{nh,p}$)

$v_{nh,p} := HS^i_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p}, i, n, p, \mu_1, \mu_2, \mu_3)$
{
if ($i == 1$ **and** coarsest mesh in local i_1-direction) **or** ($i == 2$ **and** coarsest mesh in local i_2-direction) **then**
  for $it = 1, \cdots, \mu_3$ **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 $it = 1, \cdots, \mu_1$ **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 $it = 1, \cdots, \mu_2$ **do**
  $v_{nh,p} := S^i_{nh,p}(L_{nh,p}, f_{nh,p}, v_{nh,p})$;
end for
}

can be defined for the $l+1$ pseudo-time step as

$$
\begin{aligned}
&v_0 = v^l_{nh,p} \\
&v_k = \big(I_{nh,p} + \beta_k \lambda_\sigma L^{i_{11}}_{nh,p}\big)^{-1} \big(v_0 - \lambda_\sigma \sum_{j=0}^{k-1} \alpha_{kj} (L^{i_{12}}_{nh,p} v_j - f_{nh,p})\big), \quad k = 1, \cdots, 5, \\
&v^{l+1}_{nh,p} = S^i_{nh,p} v^l_{nh,p} = v_5,
\end{aligned}
\tag{3.4}
$$

with a similar relation for sweeps in the i_2-direction. Here, α_{kj} are the Runge-Kutta coefficients, $\beta_k = \sum_{j=0}^{k-1} \alpha_{kj}$ for $k = 1, \cdots 5$, $\lambda_\sigma = \triangle\sigma / \triangle t$, and $\triangle\sigma$ the pseudo-time step. At steady state of the σ-pseudo-time integration we obtain the solution of

$$
L_{nh,p} v_{nh,p} = f_{nh,p}. \tag{3.5}
$$

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 (3.5). 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 p the remaining fifteen undefined Runge-Kutta coefficients will be computed by optimizing the convergence rate of the *hp-MGS* algorithm using the multilevel analysis discussed in Part I. In addition, the optimal value of λ_σ is determined for each polynomial level.

3.3. Multigrid error transformation operator. In the multigrid optimization we will search for Runge-Kutta coefficients which minimize the spectral radius of the *hp-MGS* error transformation operator $M_{h,3}$. For completeness, we summarize in this section the general form of the *hp-MGS* error transformation operator $M_{nh,p}$, which relates the initial and multigrid error, viz.

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

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

$$\begin{aligned} M_{nh,p} = &\big(HU_{nh,p}\big)^{\gamma_2}\big(I_{nh,p} - T^p_{nh,p-1}(I_{nh,p-1} - M_{nh,p-1})(L_{nh,p-1})^{-1} \\ &Q^{p-1}_{nh,p} L_{nh,p}\big)\big(HU_{nh,p}\big)^{\gamma_1} && \text{if } p > 1, \quad (3.6)\\ = &\, HU_{nh,1} && \text{if } p = 1. \end{aligned}$$

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

$$\begin{aligned} HU_{nh,p} = &\big(HS^1_{nh,p} HS^2_{nh,p}\big)^{\nu_2}\big(I_{nh,p} - P^{nh}_{2nh,p}(I_{2nh,p} - HU_{2nh,p}) \\ &(L_{2nh,p})^{-1} R^{2nh}_{nh,p} L_{nh,p}\big)\big(HS^2_{nh,p} HS^1_{nh,p}\big)^{\nu_1}, && \text{if } n < m, \quad (3.7)\\ = &\, 0, && \text{if } n = m. \end{aligned}$$

The $HU_{nh,p}$ error transformation operator (3.7) 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

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

Finally, the error after one semi-implicit Runge-Kutta step can be defined recursively as

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

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 $M_{nh,p}$.

4. Computational Complexity. The computational complexity of the *hp-MGS* algorithm in combination with its convergence rate determine its efficiency. Since the semi-implicit Runge-Kutta smoother, discussed in Section 3.2, is by far the computationally most expensive part of the *hp-MGS* algorithm we will focus on the number of operations of this smoother. In order to prevent unnecessarily complex estimates of the computational complexity we assume a structured mesh with $N \times M$ elements in, respectively, the $\bar{x}_1$- and $\bar{x}_2$-direction. In the semi-implicit Runge-Kutta smoother we need to solve then M block tri-diagonal matrices with N blocks on the main diagonal. The blocks in the space-time DG discretization have size m_p, with p the polynomial order of the basis functions. Three polynomial levels will be considered in the analysis. For steady state problems we have $m_1 = 3$, $m_2 = 6$ and $m_3 = 10$. We also assume that L multigrid levels are used, both in the uniform and semi-coarsening multigrid steps of the *hp-MGS* algorithm. The multigrid parameters in Algorithms 1 - 3 are set equal to $\gamma_1 = \gamma_2 = \nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 1$.

The computational cost of solving these linear systems contains two components, viz. the construction of the LU-decomposition of each matrix used in the smoother and the back solution using forward and backward substitutions. In [14] estimates are given for the number of operations for both steps, which are essentially the same if we either consider a block or band solve of the linear system. The LU-decompositions require approximately $\frac{7}{3} NM m_p^3$ operations and the back-solve $3NMm_p^2$ operations. The number of operations in the semi-implicit Runge-Kutta smoother during the semi-coarsening multigrid sweeps in the $\bar{x}_1$- and $\bar{x}_2$-direction in Algorithm 3 is then for the LU-decomposition equal to

$$2 \cdot \left(\frac{7}{3} NM m_p^3 + \frac{7}{3} \frac{NM}{2} m_p^3 + \cdots + \frac{7}{3} \frac{NM}{2^{L-1}} m_p^3 \right) = \frac{14}{3} NM m_p^3 \cdot 2 \left(1 - (\frac{1}{2})^L \right), \quad (4.1)$$

where the factor 2 on the left hand side accounts for the two semi-coarsening directions. The number of operations for the back-solve, assuming a V-cycle multigrid in both the $\bar{x}_1$- and $\bar{x}_2$-direction, is

$$2 \cdot 2 \cdot \left(3NMm_p^2 + 3 \frac{NM}{2} m_p^2 + \cdots + 3 \frac{NM}{2^{L-1}} m_p^2 \right) = 12 NM m_p^2 \cdot 2 \left(1 - (\frac{1}{2})^L \right), \quad (4.2)$$

where the factors 2 on the left hand side account for the V-cycle and the two semi-coarsening directions. The semi-coarsening multigrid is used as smoother in the *h-MGS*-multigrid in Algorithm 2. Using (4.1) - (4.2) and assuming that $L \geq 2$ we obtain the following estimate for the number of operations in the LU-decompositions

in Algorithm 2

$$\frac{14}{3}NMm_p^3 \cdot 2\left(1-(\frac{1}{2})^L\right)+\frac{14}{3}\frac{NM}{4}m_p^3\cdot 2\left(1-(\frac{1}{2})^{L-1}\right)+\cdots \\ +\frac{14}{3}\frac{NM}{4^{L-2}}m_p^3\cdot 2(1-(\frac{1}{2})^2)=\frac{14}{3}NMm_p^3\cdot 2\sum_{k=0}^{L-2}\frac{1}{4^k}\left(1-(\frac{1}{2})^{L-k}\right).$$

Note, the coarsest uniformly refined mesh is not included here since a direct solver is used at this level. If the number of multigrid levels L is sufficiently large this cost is negligible. Analogously, we obtain for the number of operations in the back-solve in Algorithm 2

$$2\cdot 12NMm_p^2\cdot 2\sum_{k=0}^{L-2}\frac{1}{4^k}\left(1-(\frac{1}{2})^{L-k}\right),$$

where the first factor 2 accounts for the V-cycle multigrid. In order to simplify notation we introduce

$$T(L)=2\sum_{k=0}^{L-2}\frac{1}{4^k}\left(1-(\frac{1}{2})^{L-k}\right)=\frac{2}{3}\left(4-(\frac{1}{4})^{L-1}-3(\frac{1}{2})^{L-1}\right),\quad L\geq 2,$$

hence $T(2)=3/2$, $T(3)=17/8$ and $T(4)=77/32$. The cost of the LU-decompositions in the full *hp-MGS* algorithm can now be estimated as

$$LU^1_{cost}=\frac{14}{3}NMT(L)(m_1^3+m_2^3+m_3^3),$$

and the cost of the back-solve is

$$B^1_{cost}=24NMT(L)(m_1^2+2m_2^2+2m_3^2). \tag{4.3}$$

The factors 2 in (4.3) are due to the fact that the $p=2$ and 3 levels are visited twice in the p-multigrid cycle.

We also consider two simplifications of the *hp-MGS* algorithm, viz. the *hp-MGS(1)* and the *hp*-multigrid algorithms, which are defined in Section 3.1. The computational cost of the LU-decomposition in the *hp-MGS(1)* algorithm at the polynomial levels $p=2$ and 3 is $\frac{14}{3}NMm_p^3$, since we need to generate an LU-decomposition for both coordinate directions. The cost of the back-solve at these levels is $12NM(m_2^2+m_3^2)$ using the fact that the $p=2$ and 3 levels are visited twice in the p-multigrid cycle and there are two semi-coarsening directions. Combining all contributions then gives for the LU-decomposition in the *hp-MGS(1)* algorithm the number of operations

$$LU^2_{cost}=\frac{14}{3}NM(m_2^3+m_3^3)+\frac{14}{3}NMT(L)m_1^3,$$

and the cost of the back-solve is

$$B^2_{cost}=12NM(m_2^2+m_3^2)+24NMT(L)m_1^2.$$

TABLE 4.1

Overview of the computational complexity consisting of LU-decomposition cost (LU_{cost}) and back-solver cost (B_{cost}). Superscripts 1, 2 and 3 refer to the hp-MGS, hp-MGS(1) and the hp-multigrid algorithms, respectively.

LU^1_{cost}	LU^2_{cost}	LU^3_{cost}	LU^1_{cost}/LU^2_{cost}	LU^1_{cost}/LU^3_{cost}
$12326\,NM$	$5942\,NM$	$5840\,NM$	2.1	2.1

B^1_{cost}	B^2_{cost}	B^3_{cost}	B^1_{cost}/B^2_{cost}	B^1_{cost}/B^3_{cost}
$14331\,NM$	$2091\,NM$	$1773\,NM$	6.9	8.1

TABLE 4.2

Overview of the memory necessary to store the LU-decomposition matrices (Mem). Superscripts 1, 2 and 3 refer to the hp-MGS, hp-MGS(1) and the hp-multigrid algorithms, respectively.

Mem^1	Mem^2	Mem^3	Mem^1/Mem^2	Mem^1/Mem^3
$3081\,NM$	$1551\,NM$	$1478\,NM$	2.0	2.1

Finally, we consider *hp*-multigrid which uses the semi-implicit Runge-Kutta smoother in both local coordinate directions at all polynomial levels in combination with standard *h*-multigrid with uniform coarsening at the $p = 1$ level. The cost of the LU-decomposition is then equal to

$$LU^3_{cost} = \frac{14}{3} NM \left(\frac{4}{3} m_1^3 (1 - (\frac{1}{4})^L) + m_2^3 + m_3^3 \right),$$

and the cost of the back-solve

$$B^3_{cost} = 2 \cdot 3NM \left(2 \cdot \frac{4}{3} m_1^2 (1 - (\frac{1}{4})^L) + 2m_2^2 + 2m_3^2 \right).$$

Here, the first factor 2 accounts for the fact that the smoother acts in two directions. The other factors 2 are for, respectively, the V-cycle in the *h*- and *p*-multigrid. In Table 4.1 the estimates for the LU-decomposition and the back-solve costs for the different algorithms are summarized. Also, the ratio of the computational cost of the *hp-MGS* algorithm to the simplified algorithms is given.

The results in Table 4.1 indicate that the cost of the LU-decomposition is substantial for all three versions of the multigrid algorithm. Since for linear problems the LU-decomposition only needs to be computed once, and also can be reused many times for nonlinear problems, it is beneficial to store these matrices. The required amount of memory to store these matrices and also the matrices used in the explicit part of the semi-implicit Runge-Kutta algorithm at each grid level is for the *hp-MGS* algorithm proportional to

$$Mem^1 = 10NMT(L)(m_1^2 + m_2^2 + m_3^2),$$

where we accounted for the five block diagonals in the discretization matrix and the two semi-coarsening directions. For the *hp-MGS(1)* algorithm the approximate memory use for the matrices is

$$Mem^2 = 10NM \left(T(L) m_1^2 + m_2^2 + m_3^2 \right),$$

and for standard *hp*-multigrid with uniform coarsening at the $p = 1$ level, the approximate memory use is

$$Mem^3 = 10NM\left(\frac{4}{3}m_1^2(1-(\frac{1}{4})^L) + m_2^2 + m_3^2\right).$$

An overview of the memory use for the different algorithms is given in Table 4.2.

5. Multigrid Optimization. The pseudo-time Runge-Kutta smoother, discussed in Section 3.2, is an important part of the *hp-MGS* algorithm and has significant influence on the multigrid performance. Since time-accuracy is not important for pseudo-time smoothers we only require that the Runge-Kutta smoothers are consistent and first order accurate. The remaining coefficients can be chosen such that the multigrid performance for a selected class of problems is optimal. In [7, 12, 15] we performed this optimization for explicit Runge-Kutta methods. In this section we will discuss the optimization of the semi-implicit Runge-Kutta smoother used in the *hp-MGS* multigrid algorithm. We consider a fourth order accurate space-time discontinuous Galerkin discretization of the two-dimensional advection-diffusion equation. The *hp-MGS* multigrid algorithm used in the optimization process has three polynomial levels ($p = 1, 2, 3$) in the p-multigrid and three mesh levels in the h-multigrid part, both for the uniformly and semi-coarsened meshes, see Figures 3.1 and 3.2. In all computations the multigrid parameters in Algorithms 1 - 3 are $\gamma_1 = \gamma_2 = \nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 1$. Only, for the cell Reynolds number $Re_h = 10^4$, we use the parameters $\nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 2$.

The error $e^1_{h,3}$ after one full *hp-MGS* iteration is determined by the multigrid error transformation operator $M_{h,3}$ as

$$e^1_{h,3} = M_{h,3}e^0_{h,3}.$$

The multigrid error transformation operator $M_{h,3}$ is defined in Section 3.3. The optimization of the Runge-Kutta smoother in the *hp-MGS* algorithm is performed simultaneously for all polynomial levels, since there is a strong interaction between these multigrid levels in the *hp-MGS* algorithm. In the optimization process we search for Runge-Kutta coefficients which minimize the spectral radius $\rho(M_{h,3})$ of the *hp-MGS* error transformation operator. In addition, we require that the spectral radii of the Runge-Kutta smoothers $S^i_{nh,p}$, $i = 1, 2$, in Algorithm 3 are less than one for all polynomial levels p. These constraints on the smoothers are essential to obtain a robust multigrid algorithm. Unstable smoothers for some p-levels can give a better multigrid performance, but are not reliable. In addition, we require that each of the semi-coarsening multigrid algorithms $HS^i_{nh,p}$, $i = 1, 2$, given by Algorithm 3, have a spectral radius less than one. The operator norms and spectral radii of the multigrid error transformation operator and the Runge-Kutta smoothers are computed for the two-dimensional advection-diffusion equation using the discrete Fourier multilevel analysis discussed in Part I, [16].

The operator norm $\|M_{h,3}\|$ of the *hp-MGS* error transformation operator provides an upper bound for the reduction of the error after one iteration of the full *hp-MGS* algorithm. We also compute the norm $\|M_{h,3}\|_L := \|L^{-1}_{h,3}M_{h,3}L_{h,3}\|$, which gives an upper bound for the reduction of the residual after one *hp-MGS* iteration, and the spectral radius $\rho(M_{h,3})$, which gives the asymptotic convergence rate. See [4] for an explanation of the different convergence measures.

For the computation of the operator norms and spectral radius an extensive Matlab program was written, which performs the full multilevel analysis of the *hp-MGS*

TABLE 5.1

Optimized Runge-Kutta smoother coefficients α_{ij} and pseudo-time step λ_σ for the hp-MGS algorithm for a fourth order space-time DG discretization of the steady advection-diffusion equation ($\gamma_1 = \gamma_2 = \nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 1$).

Runge-Kutta coefficients $Re_h = 1$			
	P^{10}-level	P^{20}-level	P^{30}-level
λ_σ	4.064e+01	8.239e+01	1.149e+02
$\alpha_{1,0}$	1.066209267521e+00	3.350894806084e-01	2.727948283871e-01
$\alpha_{2,0}$	7.180492865411e-01	2.951566984408e-02	3.819041499688e-01
$\alpha_{2,1}$	6.163816243989e-01	3.756895602864e-01	2.044998728477e-01
$\alpha_{3,0}$	-4.527958372691e-01	-4.535683794919e-01	-4.562349170706e-01
$\alpha_{3,1}$	-5.583612017693e-01	-2.831700768733e-01	9.511751654662e-03
$\alpha_{3,2}$	1.967631764281e+00	-4.908716265626e-01	7.156119819888e-01
$\alpha_{4,0}$	-8.888763218402e-01	-4.238398891230e-01	-8.897482288601e-02
$\alpha_{4,1}$	-1.401525685401e+00	1.831110282381e-01	-1.046204349088e-01
$\alpha_{4,2}$	4.844557217353e-01	1.676572696271e-01	-6.168593216634e-01
$\alpha_{4,3}$	2.048020891903e-01	7.775979258990e-01	-1.041586085909e-01
$\alpha_{5,0}$	-1.652446826064e+00	-2.221934676001e+00	-1.053917760604e+00
$\alpha_{5,1}$	-5.258533708371e-01	2.531223130879e-01	2.184090410378e-01
$\alpha_{5,2}$	8.008864716245e-02	5.872585947323e-01	4.911479000314e-02
$\alpha_{5,3}$	1.281626589298e+00	5.554657058114e-01	5.641596413156e-01
$\alpha_{5,4}$	1.816584960441e+00	1.826088062369e+00	1.222234288247e+00

Runge-Kutta coefficients $Re_h = 10$			
	P^{10}-level	P^{20}-level	P^{30}-level
λ_σ	4.064e+01	8.239e+01	1.149e+02
$\alpha_{1,0}$	1.067076553707e+00	3.354038200637e-01	2.698898366842e-01
$\alpha_{2,0}$	7.475044908338e-01	4.364450123197e-01	3.630028474154e-01
$\alpha_{2,1}$	5.782869546967e-01	3.471346464581e-01	2.370748120992e-01
$\alpha_{3,0}$	-2.321022121346e-01	-6.045448400458e-01	-4.677183793622e-01
$\alpha_{3,1}$	-4.629530713053e-01	-2.724698584165e-01	1.462815397238e-04
$\alpha_{3,2}$	2.039046812330e+00	-3.566993131079e-01	7.155518835259e-01
$\alpha_{4,0}$	-9.398870654851e-01	3.920819132938e-01	-7.797329349952e-02
$\alpha_{4,1}$	-1.325186004004e+00	6.260665126301e-01	-9.457888411948e-02
$\alpha_{4,2}$	5.762255233657e-01	4.600411796511e-01	-6.119099535840e-01
$\alpha_{4,3}$	3.082732977044e-01	1.056866241204e+00	-1.530418886823e-01
$\alpha_{5,0}$	-5.389654416153e-01	-8.815339257398e-01	-9.984400648089e-01
$\alpha_{5,1}$	-7.330159070872e-01	5.791100087170e-02	2.011573494938e-01
$\alpha_{5,2}$	-1.783450185086e-01	2.596544690805e-01	3.175641184709e-02
$\alpha_{5,3}$	8.890378339131e-01	2.467434890736e-01	5.530581503410e-01
$\alpha_{5,4}$	1.561288533298e+00	1.317224966714e+00	1.212468153127e+00

algorithm discussed in Part I using a finite number of Fourier modes. Since the discrete Fourier multilevel analysis is quite intricate it is verified by comparing the results with a matrix analysis of the *hp-MGS* algorithm. This analysis computes the operator norms and spectral radius directly from the matrix representation of the error transformation operator. For all cases the operator norms and spectral radius computed with the discrete Fourier transform and the matrix analysis agree up to machine precision. The matrix analysis of the full *hp-MGS* algorithm is, however, computationally

TABLE 5.2
Optimized Runge-Kutta smoother coefficients α_{ij} and pseudo-time step λ_σ for the hp-MGS algorithm for a fourth order space-time DG discretization of the steady advection-diffusion equation ($\gamma_1 = \gamma_2 = \nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 1$).

Runge-Kutta coefficients $Re_h = 10^2$			
	P^{10}-level	P^{20}-level	P^{30}-level
λ_σ	4.059e+01	8.238e+01	1.148e+02
$\alpha_{1,0}$	1.875168099757e-01	2.042989931395e-01	4.772611338845e-01
$\alpha_{2,0}$	1.788557210400e-02	7.606498423253e-03	-8.312761869675e-02
$\alpha_{2,1}$	2.619682251192e-01	2.522228252442e-01	4.664398891082e-01
$\alpha_{3,0}$	2.081705840618e-03	-1.379499687713e-01	-1.344442798725e-01
$\alpha_{3,1}$	1.840377842528e-02	-1.456043068852e-02	5.512529454746e-02
$\alpha_{3,2}$	3.526505428668e-01	3.377211127332e-01	5.032484753505e-01
$\alpha_{4,0}$	3.478615954498e-02	-8.318763590321e-03	2.399183961118e-03
$\alpha_{4,1}$	-6.473522267297e-03	-6.202015368083e-02	-1.424155037565e-01
$\alpha_{4,2}$	4.761322957430e-03	-2.108372207376e-02	1.499062564444e-01
$\alpha_{4,3}$	5.364376367676e-01	5.389061802536e-01	5.259982962831e-01
$\alpha_{5,0}$	7.030542918979e-02	6.293201561569e-02	-6.467621539816e-02
$\alpha_{5,1}$	-1.536408672031e-02	-1.300641145400e-02	3.438390839114e-02
$\alpha_{5,2}$	-2.724347090973e-02	-3.031293146299e-02	-1.243092975350e-01
$\alpha_{5,3}$	-2.496762906715e-02	-4.067726799470e-02	7.797864129102e-02
$\alpha_{5,4}$	9.972697575074e-01	1.021064595296e+00	1.076622963251e+00

Runge-Kutta coefficients $Re_h = 10^3$			
	P^{10}-level	P^{20}-level	P^{30}-level
λ_σ	4.059e+01	8.239e+01	1.149e+02
$\alpha_{1,0}$	1.230191320983e-01	3.787544077773e-01	6.834476146419e-01
$\alpha_{2,0}$	-1.911011219608e-02	-2.442927468845e-02	-8.067765104029e-02
$\alpha_{2,1}$	1.791974565594e-01	4.789662134943e-01	8.442823072522e-01
$\alpha_{3,0}$	6.607895783461e-02	-1.211319344924e-02	-1.736668864009e-02
$\alpha_{3,1}$	1.041899359346e-02	2.965031660220e-02	-1.180087757507e-02
$\alpha_{3,2}$	3.147642224444e-01	5.936409581635e-01	9.361622777963e-01
$\alpha_{4,0}$	5.102610692669e-02	1.127839537195e-02	6.423732366912e-02
$\alpha_{4,1}$	4.620074200916e-02	-3.156972133469e-02	3.983702905524e-03
$\alpha_{4,2}$	1.821584235271e-02	-6.209853413418e-03	1.220654565895e-01
$\alpha_{4,3}$	5.394650992184e-01	7.388196708596e-01	1.027157194092e+00
$\alpha_{5,0}$	-5.542549811339e-02	-3.278940681468e-02	-8.604435919780e-03
$\alpha_{5,1}$	1.097845276121e-02	-3.578745418390e-02	-9.592188900718e-02
$\alpha_{5,2}$	2.057300337444e-02	-5.121999888593e-02	-1.510905772327e-01
$\alpha_{5,3}$	6.855657560740e-03	-1.101851473049e-02	-1.462692551634e-02
$\alpha_{5,4}$	1.017018384417e+00	1.130815374615e+00	1.270243827676e+00

far too expensive to be used in an optimization process. The constraint optimization for the Runge-Kutta coefficients is conducted with the Matlab function *fmincon*. In the optimization process 32×32 Fourier modes were used. Increasing the number of modes has minor influence on the results.

In this article we only consider steady state problems, since it is considerably more difficult to obtain good multigrid performance for steady state than for time-accurate problems with a space-time DG discretization. The steady state solution

TABLE 5.3
Optimized Runge-Kutta smoother coefficients α_{ij} and pseudo-time step λ_σ for the hp-MGS algorithm for a fourth order space-time DG discretization of the steady advection-diffusion equation ($\gamma_1 = \gamma_2 = \mu_3 = 1,\ \nu_1 = \nu_2 = \mu_1 = \mu_2 = 2$).

Runge-Kutta coefficients $Re_h = 10^4$			
	P^{10}-level	P^{20}-level	P^{30}-level
λ_σ	4.059e+01	8.239e+01	1.149e+02
$\alpha_{1,0}$	1.196700090440e-01	3.521851754420e-01	6.714970808381e-01
$\alpha_{2,0}$	-8.730243741498e-04	-1.946937702667e-02	-5.354183172912e-02
$\alpha_{2,1}$	1.734472033543e-01	4.629686954431e-01	8.249862430844e-01
$\alpha_{3,0}$	8.084957377579e-02	8.308777440605e-03	3.299173830984e-02
$\alpha_{3,1}$	1.844866881966e-02	2.836597896229e-02	2.334203182750e-04
$\alpha_{3,2}$	3.117604893605e-01	5.668150586102e-01	9.060620264916e-01
$\alpha_{4,0}$	6.465279866316e-02	9.966464841345e-03	7.808773075933e-02
$\alpha_{4,1}$	5.010267429506e-02	-9.842761904063e-03	4.934112313665e-02
$\alpha_{4,2}$	2.262216358794e-02	-3.620009174632e-03	9.768424174771e-02
$\alpha_{4,3}$	5.300143933421e-01	7.154247867145e-01	9.962375617257e-01
$\alpha_{5,0}$	-6.438852672615e-03	-3.986306405263e-02	-2.930006213412e-02
$\alpha_{5,1}$	6.915694851553e-03	-2.880173428246e-02	-7.612015487044e-02
$\alpha_{5,2}$	3.719052958574e-03	-3.386419857743e-02	-1.134426465864e-01
$\alpha_{5,3}$	-5.460166654512e-03	-1.611563144248e-02	-1.854202477204e-02
$\alpha_{5,4}$	1.001264271517e+00	1.118644628355e+00	1.237404888363e+00

of the advection-diffusion equation depends on the cell Reynolds number Re_h, the mesh aspect ratio and the flow angle, see Section 2. The Runge-Kutta coefficients are optimized for a flow angle of 45° and a mesh aspect ratio of one. The spectra of the space-time DG discretization, however, strongly depend on the cell Reynolds number Re_h. For this purpose the optimization was performed for a range of cell Reynolds numbers, from very viscous to nearly inviscid problems. After optimization the effect of flow angle and mesh aspect ratio were investigated.

The Runge-Kutta coefficients obtained from the multigrid optimization are given for a wide range of cell Reynolds numbers in Tables 5.1–5.3. Since we consider steady state problems, the polynomial basis functions are constant in time and linear, quadratic and cubic in space, indicated respectively, as P^{10}, P^{20} and P^{30}. For $Re_h \leq 1$ the Runge-Kutta coefficients do not depend on the cell Reynolds number, but for larger values of Re_h the smoother coefficients change, however, significantly see, Tables 5.1–5.3,

The performance of the optimized *hp-MGS* algorithm is investigated for a number of test cases using the multilevel analysis described in Part I. In all computations 64×64 Fourier modes are used, which is sufficient to obtain an accurate prediction of the multigrid performance. In Table 5.4 the spectral radius $\rho(M_{h,3})$ and the operator norms $\|M_{h,3}\|$ and $\|M_{h,3}\|_L$ are presented for two flow angles, viz. $\alpha = 15°$ and $\alpha = 45°$ with respect to the $\bar{x}_1$-axis, as a function of the cell Reynolds number on a mesh with aspect ratio $A_h = 1$. This are representative flow angles since the flow field for flow angles in the range $(90°, 360°)$ can be obtained using symmetry considerations. For values of $Re_h \leq 10^2$ the convergence rate is extremely good, but for Re_h values beyond 10^3 the spectral radius increases to approximately 0.7. The influence of the flow angle is small.

Next, we investigated the influence of a number of simplifications of the *hp-MGS*

TABLE 5.4
Spectral radius and operator norms of the hp-MGS error transformation operator for flow angles $\alpha = 15^\circ$ and 45° on a mesh with aspect ratio $A_h = 1$.

	$\alpha = 15^\circ$			$\alpha = 45^\circ$		
Re_h	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-1}	4.854e-08	6.471e-08	6.472e-08	5.108e-08	6.420e-08	6.420e-08
10^{0}	3.963e-07	5.067e-07	5.068e-07	2.537e-07	2.976e-07	2.977e-07
10^{1}	2.283e-05	2.789e-05	2.792e-05	1.118e-03	1.431e-03	1.433e-03
10^{2}	2.239e-05	2.797e-05	2.915e-05	5.464e-04	7.662e-04	7.898e-04
10^{3}	2.722e-01	4.355e-01	4.368e-01	4.000e-01	7.379e-01	7.364e-01
10^{4}	4.547e-01	8.378e-01	8.536e-01	7.687e-01	1.282e+00	1.287e+00

TABLE 5.5
Spectral radius and operator norms of the hp-MGS(1) error transformation operator for flow angles $\alpha = 15^\circ$ and $\alpha = 45^\circ$ on a mesh with aspect ratio $A_h = 1$.

	$\alpha = 15^\circ$			$\alpha = 45^\circ$		
Re_h	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-1}	2.950e-02	8.204e-02	8.204e-02	2.932e-02	9.137e-02	9.137e-02
10^{0}	2.886e-02	5.311e-02	5.310e-02	3.743e-02	9.469e-02	9.469e-02
10^{1}	2.044e-01	2.647e-01	2.652e-01	3.057e-01	3.670e-01	3.678e-01
10^{2}	4.361e-01	5.084e-01	5.182e-01	8.106e-01	1.049e+00	1.067e+00
10^{3}	8.597e-01	1.293e+00	1.306e+00	9.781e-01	1.645e+00	1.668e+00
10^{4}	9.522e-01	1.730e+00	1.762e+00	9.953e-01	1.981e+00	2.007e+00

TABLE 5.6
Spectral radius and operator norms of the hp-multigrid error transformation operator for flow angles $\alpha = 15^\circ$ and $\alpha = 45^\circ$ on a mesh with aspect ratio $A_h = 1$.

	$\alpha = 15^\circ$			$\alpha = 45^\circ$		
Re_h	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-1}	5.750e-02	9.355e-02	9.355e-02	5.550e-02	9.654e-02	9.654e-02
10^{0}	6.802e-02	8.896e-02	8.897e-02	8.715e-02	1.247e-01	1.247e-01
10^{1}	2.691e-01	3.114e-01	3.125e-01	4.342e-01	5.057e-01	5.065e-01
10^{2}	6.215e-01	7.391e-01	7.552e-01	8.715e-01	1.127e+00	1.147e+00
10^{3}	9.177e-01	1.353e+00	1.364e+00	9.856e-01	1.645e+00	1.668e+00
10^{4}	9.761e-01	1.729e+00	1.762e+00	9.980e-01	1.981e+00	2.007e+00

algorithm. The first simplification is the *hp-MGS(1)* algorithm, which uses the semi-implicit Runge-Kutta smoother at the $p = 2$ and 3 levels and the *h-MGS* algorithm only at the $p = 1$ level, see Section 3.1. The multigrid performance of this algorithm is summarized in Table 5.5 . The second simplification is to use a standard *hp*-multigrid algorithm, which uses *h*-multigrid with uniformly coarsened meshes at the $p = 1$ level and the semi-implicit Runge-Kutta smoother at all *p*-levels, see Table 5.6.

The effect of these simplifications is very large if one compares the results in Tables 5.5 and 5.6 with Table 5.4. For cell Reynolds numbers $Re_h \leq 10^1$ the multigrid convergence rate of the simplified algorithms is still very good, although the full *hp-MGS* algorithm converges much faster, even if one takes into account that the computational cost of the *hp-MGS* algorithm is about a factor 8 larger based on the results of the analysis of the computational complexity given in Table 4.1. For

TABLE 5.7
Spectral radius and operator norms of the hp-MGS error transformation operator on a mesh with aspect ratio $A_h = 100$ for flow angles of 15°, 45°, and 75°.

		$\alpha = 15°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	3.944e-10	1.039e-09	1.039e-09
10^{-4}	10^{0}	1.355e-09	3.077e-09	3.076e-09
10^{-3}	10^{1}	1.049e-05	1.418e-05	1.420e-05
10^{-2}	10^{2}	1.132e-04	1.413e-04	1.426e-04
10^{-1}	10^{3}	7.716e-04	2.063e-03	2.080e-03
10^{0}	10^{4}	1.758e-07	2.596e-07	2.624e-07
		$\alpha = 45°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	3.965e-10	1.045e-09	1.045e-09
10^{-4}	10^{0}	1.426e-09	3.242e-09	3.242e-09
10^{-3}	10^{1}	5.924e-04	7.142e-04	7.151e-04
10^{-2}	10^{2}	9.435e-06	2.664e-05	2.793e-05
10^{-1}	10^{3}	1.765e-03	1.906e-02	1.914e-02
10^{0}	10^{4}	1.884e-05	3.724e-05	3.741e-05
		$\alpha = 75°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	3.983e-10	1.049e-09	1.049e-09
10^{-4}	10^{0}	1.485e-09	3.375e-09	3.374e-09
10^{-3}	10^{1}	8.241e-04	9.918e-04	9.931e-04
10^{-2}	10^{2}	7.153e-05	1.607e-04	1.664e-04
10^{-1}	10^{3}	1.021e-01	3.093e-01	3.103e-01
10^{0}	10^{4}	6.278e-03	1.428e-02	1.441e-02

cell Reynolds numbers $Re_h \geq 10^3$ the convergence rate of the simplified multigrid algorithms is very poor. Since the Runge-Kutta coefficients were optimized for the full *hp-MGS* algorithm one might wonder if a direct optimization of the simplified multigrid algorithms would improve the convergence rate. This, however, is not the case, which shows the importance of using the full *h-MGS* algorithm at all *p*-levels.

Since the Runge-Kutta coefficients were optimized for a uniform mesh it is important to investigate the multigrid performance on highly stretched meshes. For this purpose we consider a mesh aspect ratio $A_h = 100$ and various flow angles. The mesh aspect ratio $A_h = 100$ results in a factor 10^4 difference in the cell Reynolds numbers Re_{h_1} and Re_{h_2} in, respectively, the $\bar{x}_1$- and $\bar{x}_2$-coordinate directions. On non-uniform meshes the Runge-Kutta coefficients of the smoother are selected using the maximum cell Reynolds number $Re_{h,max} := \max(Re_{h_1}, Re_{h_2})$ and the coefficients are selected from the class with the lowest Re_h value in Tables 5.1 – 5.3, such that $Re_{h,max} \leq Re_h$. This approach is also used in Section 6 for problems which require a large mesh stretching to account for thin boundary layers. The multigrid performance of the *hp-MGS* algorithm for flow angles of 15°, 45° and 75° is summarized in Table 5.7. On these highly stretched meshes the convergence rate is extremely good, despite the fact that the Runge-Kutta smoother coefficients were optimized for a uniform mesh with $A_h = 1$. If we simplify the *hp-MGS* algorithm and use either the *hp-MGS(1)* algorithm or the *hp*-multigrid method then the convergence rate is still

TABLE 5.8
*Spectral radius and operator norms of the hp-MGS(1) error transformation operator on a mesh with aspect ratio $A_h = 100$ and flow angles 15°, 45° and 75° (**$\nu_1 = \nu_2 = \mu_1 = \mu_2 = 2$).*

		$\alpha = 15°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.794e-02	8.925e-02	8.925e-02
10^{-4}	10^{0}	6.569e-02	1.012e-01	1.012e-01
10^{-3}	10^{1}	8.647e-02	1.339e-01	1.351e-01
10^{-2}	10^{2}	1.442e-01	3.285e-01	3.142e-01
10^{-1}	10^{3}	1.105e-01	1.688e-01	1.694e-01
10^{0}	10^{4}	2.747e-02	4.444e-02	4.461e-02

		$\alpha = 45°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.797e-02	8.930e-02	8.930e-02
10^{-4}	10^{0}	6.610e-02	1.018e-01	1.018e-01
10^{-3}	10^{1}	1.368e-01	1.871e-01	1.852e-01
10^{-2}	10^{2}	1.666e-01	2.835e-01	2.804e-01
10^{-1}	10^{3}	2.924e-01	4.645e-01	4.654e-01
10^{0}	10^{4}	4.428e-01	7.206e-01	7.207e-01

		$\alpha = 75°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.799e-02	8.933e-02	8.933e-02
10^{-4}	10^{0}	6.628e-02	1.021e-01	1.021e-01
10^{-3}	10^{1}	1.917e-01	2.870e-01	2.891e-01
10^{-2}	10^{2}	2.159e-01	3.601e-01	3.585e-01
10^{-1}	10^{3}	1.224e+00	1.950e+00	1.953e+00
10^{-1}	10^{3} **	1.816e+00	2.919e+00	2.922e+00
10^{0}	10^{4}	7.941e-01	1.600e+00	1.615e+00

good for $Re_{h_2} \leq 10^2$, but much slower than for the *hp-MGS* algorithm, see Tables 5.8 and 5.9. For a flow angle of 75° the simplified multigrid schemes become, however, unstable when $Re_{h_2} \geq 10^3$. Increasing the number of pre- and post-relaxations does not improve the convergence rate. These results show that the *hp-MGS* is much more efficient and robust on highly stretched meshes than the *hp-MGS(1)* and *hp*-multigrid algorithms.

6. Multigrid performance. In order to demonstrate the performance of the *hp-MGS* algorithm we consider the 2D dimensionless advection-diffusion equation

$$\begin{aligned} \partial_t u(t,\bar{x}) + \nabla \cdot (a u(t,\bar{x})) = \tfrac{1}{Re}\Delta u(t,\bar{x}) = 0, \quad & (t,\bar{x}) \in (t_0,T) \times \Omega, \\ u(t,\bar{x}) = u_D, \quad & (t,\bar{x}) \in (t_0,T) \times \Gamma_D, \\ u(t_0,\bar{x}) = u_0, \quad & \bar{x} \in \Omega, \end{aligned} \tag{6.1}$$

with domain $\Omega = [0,1]^2$, advection velocity $a = (\cos\alpha, \sin\alpha)$, where α is the flow angle with respect to the $\bar{x}_1$-axis, Re the global Reynolds number, defined as $Re = \frac{|a|L}{\nu}$, with L a reference length for the domain Ω. Note, the global Reynolds number Re is generally much larger than the cell Reynolds number Re_h. The boundary data u_D

TABLE 5.9

*Spectral radius and operator norms of the hp-multigrid error transformation operator on a mesh with aspect ratio $A_h = 100$ and flow angles 15°, 45° and 75° (**$\nu_1 = \nu_2 = \mu_1 = \mu_2 = 2$).*

		$\alpha = 15°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.794e-02	8.925e-02	8.925e-02
10^{-4}	10^{0}	6.569e-02	1.012e-01	1.012e-01
10^{-3}	10^{1}	1.180e-01	3.087e-01	3.076e-01
10^{-2}	10^{2}	1.469e-01	3.315e-01	3.173e-01
10^{-1}	10^{3}	1.120e-01	1.717e-01	1.744e-01
10^{0}	10^{4}	2.744e-02	4.441e-02	4.460e-02

		$\alpha = 45°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.797e-02	8.930e-02	8.930e-02
10^{-4}	10^{0}	6.610e-02	1.018e-01	1.018e-01
10^{-3}	10^{1}	1.580e-01	2.597e-01	2.563e-01
10^{-2}	10^{2}	1.592e-01	2.736e-01	2.717e-01
10^{-1}	10^{3}	3.082e-01	5.275e-01	5.456e-01
10^{0}	10^{4}	4.440e-01	7.224e-01	7.227e-01

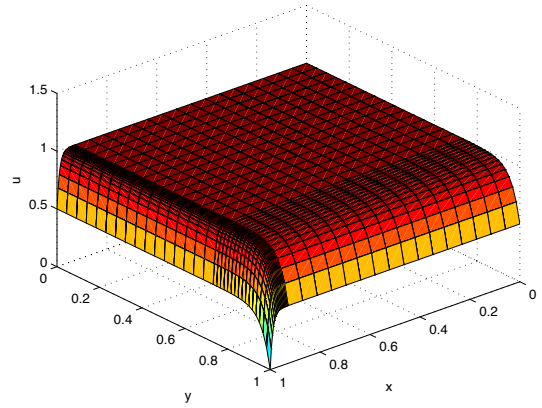
		$\alpha = 75°$		
Re_{h_1}	Re_{h_2}	$\rho(M_{h,3})$	$\|M_{h,3}\|$	$\|M_{h,3}\|_L$
10^{-5}	10^{-1}	5.799e-02	8.933e-02	8.933e-02
10^{-4}	10^{0}	6.628e-02	1.021e-01	1.021e-01
10^{-3}	10^{1}	1.957e-01	3.146e-01	3.147e-01
10^{-2}	10^{2}	2.062e-01	3.647e-01	3.652e-01
10^{-1}	10^{3}	1.148e+00	1.950e+00	1.953e+00
10^{-1}	10^{3} **	1.748e+00	2.812e+00	2.816e+00
10^{0}	10^{4}	7.907e-01	1.599e+00	1.615e+00

equal the exact steady state solution at the domain boundary, given by

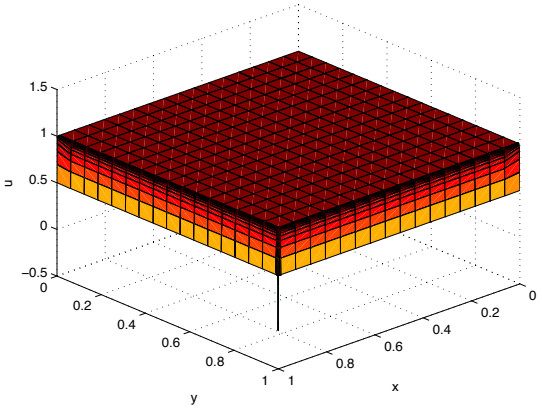
$$u(\bar{x}_1, \bar{x}_2) = \frac{1}{2}\left(\frac{\exp(a_1 Re) - \exp(a_1 Re\,\bar{x}_1)}{\exp(a_1 Re) - 1} + \frac{\exp(a_2 Re) - \exp(a_2 Re\,\bar{x}_2)}{\exp(a_2 Re) - 1}\right). \quad (6.2)$$

The exact solution has a thin boundary layer, with a thickness proportional to $1/Re$, see Figure 6.1. Note, this boundary layer is considerably thinner than the laminar boundary on a flat plate, which is proportional to $1/\sqrt{Re}$. For $Re = 1000$ we used an asymptotic expansion in terms of $1/Re$ to prescribe the boundary data, because otherwise serious underflow would occur. The extremely thin boundary layer for $Re = 1000$ is clearly visible at the point $(1, 1)$ in Figure 6.1(b). The thin boundary layer poses serious problems for Reynolds numbers larger than 10 when the algebraic system resulting from the fourth order accurate space-time DG discretization is solved with a multigrid algorithm. Even after extensive optimization for the *hp-MGS* algorithm explicit smoothers were not suitable to obtain a converged solution. We also evaluated smoothers based on various incomplete LU-decompositions of the matrix. These ILU-smoothers required, however, so much fill-in that they were essentially a direct solver, which does not make them attractive as a multigrid smoother.

In order to deal with the thin boundary layer a so-called Shishkin mesh was used. In this mesh the coordinates $(\bar{x}_1^u, \bar{x}_2^u)$ of a uniform mesh in Ω are mapped onto a mesh



(a) $Re = 100$, $\alpha = 15°$.



(b) $Re = 1000$, $\alpha = 45°$.

FIG. 6.1. *Solution of advection-diffusion equation on a* 32×32 *Shishkin mesh.*

suitable for dealing with boundary layers. The mapping is given by:

$$\bar{x}_i = \begin{cases} 2(1-\sigma_i)\bar{x}_i^u, & \text{for } \bar{x}_i^u < 0.5 \\ 1 + 2\sigma_i(\bar{x}_i^u - 1), & \text{for } \bar{x}_i^u \geq 0.5 \end{cases}, \quad i = 1, 2,$$

where $\sigma_i = \min(\frac{1}{2}, 2/(|a|Re)\ln(N_i))$, and where N_i is the number of elements in the $\bar{x}_i$-direction.

The advection-diffusion equation is solved with a fourth order accurate space-time DG discretization. For more details, see Part I. The resulting system of algebraic equations is solved with the *hp-MGS* algorithm defined in Algorithms 1 - 3 using three polynomial levels ($p = 1, 2, 3$) and three uniformly and semi-coarsened meshes. The multigrid parameters are $\gamma_1 = \gamma_2 = \nu_1 = \nu_2 = \mu_1 = \mu_2 = \mu_3 = 1$. The Runge-Kutta coefficients of the smoother (3.4) are selected in each element from Tables 5.1 – 5.3 using the maximum cell Reynolds number $Re_{h,max} = \max(Re_{h_1}, Re_{h_2})$. In each element the coefficients from the class with the lowest Re_h are used, such

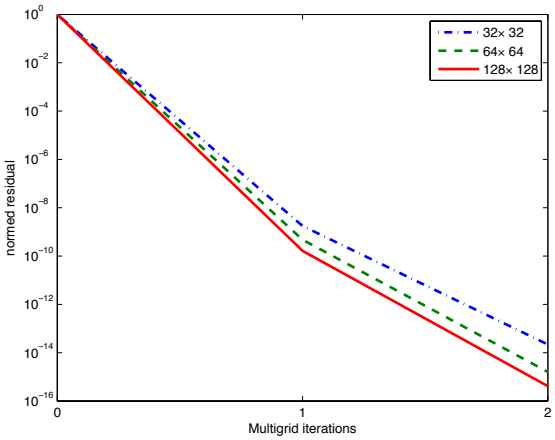


FIG. 6.2. *Mesh size dependance of the convergence rate of hp-MGS algorithm for a 4th order space-time DG discretization of the advection-diffusion equation. ($Re = 100$, $\alpha = 45°$).*

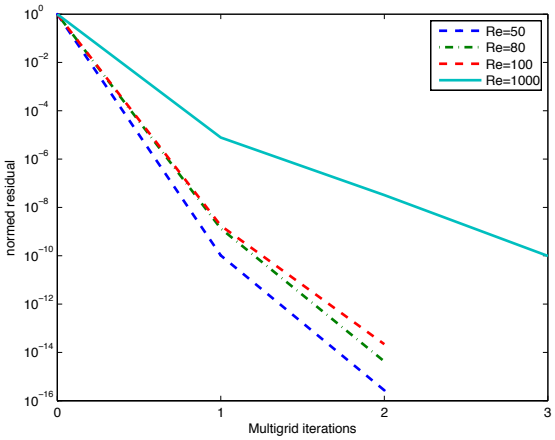


FIG. 6.3. *Reynolds number dependence of the convergence rate of hp-MGS algorithm for a 4th order space-time DG discretization of the advection-diffusion equation. (32×32 mesh, $\alpha = 45°$).*

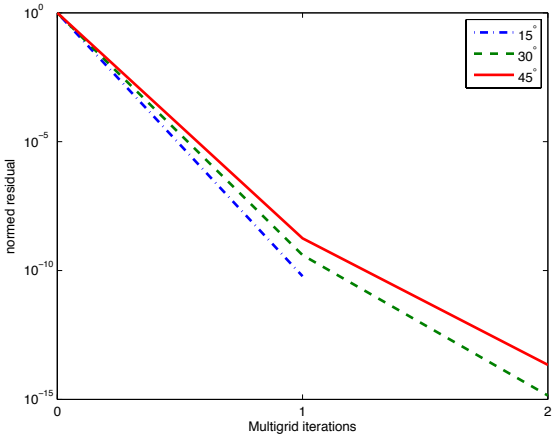


FIG. 6.4. *Flow angle dependance of the convergence rate of hp-MGS algorithm for a 4th order space-time DG discretization of the advection-diffusion equation. (32×32 mesh, $Re = 100$).*

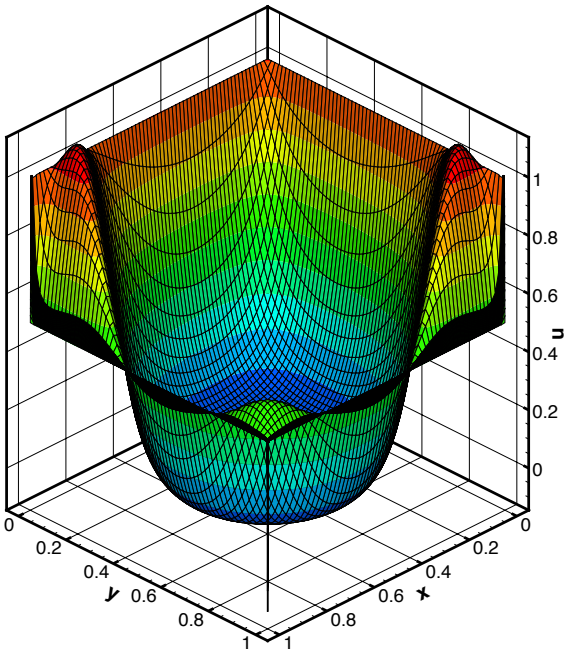


FIG. 6.5. *Solution of the advection-diffusion equation at* $Re = 1000$ *on a* 128×128 *Shishkin mesh for a rotating advective velocity field.*

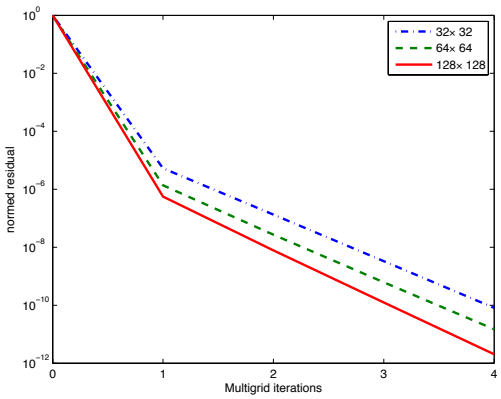


FIG. 6.6. *Grid dependance of the convergence rate of the hp-MGS algorithm for a 4th order accurate space-time DG discretization of the advection-diffusion equation at* $Re = 1000$ *for a rotating advective velocity field.*

that $Re_{h,max} \leq Re_h$. Note, this results in very different smoother coefficients in the boundary layer than in the central part of the domain. The stopping criterium for the *hp-MGS* algorithm is that the residual should decrease by 10 orders.

In the first set of computations we investigated the dependence of the convergence rate of the *hp-MGS* multigrid algorithm on the mesh size. In Figure 6.2 the convergence rates are shown for a Reynolds number $Re = 100$ and flow angle $\alpha = 45^\circ$ on meshes with 32×32, 64×64 and 128×128 elements. The *hp-MGS* algorithm shows

an excellent convergence rate on the non-uniform Shiskin mesh, independent of the mesh size, despite the fact that the optimization of the semi-implicit Runge-Kutta smoother was performed for a uniform mesh with periodic boundary conditions.

Next, we investigate the dependence of the multigrid convergence rate on the global Reynolds number. If the global Reynolds number increases then the boundary layer becomes thinner and the Shiskin mesh contains highly stretched elements near the wall, see Figure 6.1. This makes it in general more difficult to solve the algebraic system resulting from the higher order accurate space-time DG discretization. In Figure 6.3 the convergence rate of the *hp-MGS* multigrid algorithm is shown for a 32×32 mesh, flow angle $\alpha = 45^\circ$ and global Reynolds numbers $Re = 50$, 80, 100 and 1000. The multigrid convergence is excellent for all cases.

Finally, we consider the dependence on the flow angle. In Figure 6.4 the convergence rate of the *hp-MGS* multigrid algorithm for a Reynolds number $Re = 100$ on a 32×32 mesh is shown for flow angles $\alpha = 15^\circ$, 30° and 45°. Note, the coefficients in the Runge-Kutta smoothers were optimized for a flow angle $\alpha = 45^\circ$. The effect of the flow angle on the convergence rate is, however, minimal.

In order to investigate the effect of a non-constant advection velocity, we consider on the domain $\Omega = [0,1]^2$ the rotating advection velocity-field

$$a(\bar{x}) = (a_1, a_2) = c\big(\tfrac{1}{\sqrt{2}} - r\big)^n \big(\cos(\theta), \sin(\theta)\big),$$

where $r = \sqrt{(\bar{x}_1 - \frac{1}{2})^2 + (\bar{x}_2 - \frac{1}{2})^2}$ and c is such that $\max_{\bar{x}\in\Omega} |a(\bar{x})| = 1$. Taking $n = 3$, we find $c = 2.828427124746190735$. The Dirichlet boundary condition at $\partial\Omega$ is given by (6.2) with $\bar{x}$ restricted to $\partial\Omega$. The global Reynolds number is $Re = 1000$. The cell Reynolds numbers in the mesh vary between a minimum value $Re_h = 7.5 \times 10^{-13}$ and a maximum value $Re_h = 21.5$. The solution of the fourth order space-time DG discretization for this test case on a 128×128 Shishkin mesh is given in Figure 6.5. The solution has a thin boundary layer and also two discontinuities at the boundary, viz. at $(\bar{x}_1, \bar{x}_2) = (0,1)$ and $(1,0)$. The multigrid convergence is shown in Figure 6.6, which shows that also for this test case the *hp-MGS* algorithm has a nearly mesh independent convergence rate despite the thin boundary layers, non-constant advection velocity and singularities.

7. Conclusions and Outlook. The *hp-MGS* multigrid algorithm with an optimized semi-implicit Runge-Kutta smoother shows an excellent convergence rate for both advection and diffusion dominated solutions of the advection-diffusion equation, including problems with thin boundary layers and non-constant advection velocity. The larger computational complexity of the *hp-MGS* algorithm compared to simplified versions of the algorithm, including standard *hp*-multigrid, is more than compensated by its faster convergence rate. In addition, for cell Reynolds numbers $Re_h \geq 10^3$ the simplified algorithms diverge on non-uniform meshes. The convergence results were obtained both with a multilevel discrete Fourier analysis and actual computations. The *hp-MGS* algorithm combines a number of innovations, viz. the use of the *h-MGS* algorithm as smoother at all polynomial levels, which significantly improves the multigrid convergence rate and robustness for higher order accurate discretizations, the use of a new semi-implicit Runge-Kutta smoother, and the optimization of the multigrid smoother using multilevel analysis of the complete *hp-MGS* algorithm in two-space dimensions.

Currently, the *hp-MGS* multigrid algorithm is being investigated for a fourth order accurate space-time DG discretization of the Euler equations describing inviscid

compressible flows. The Runge-Kutta smoother coefficients will be optimized for the linearized Euler equations as a function of the Mach number and tested on several aerodynamic problems.

Acknowledgments. Part of this research was conducted while J.V. was visiting Professor at the Division of Applied Mathematics of Brown University, Providence, RI, USA. The kind hospitality of Prof. C.-W. Shu is greatly appreciated. The research of S.R. is partly funded by 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, A. Ghidoni, S. Rebay and P. Tesini, *High-order accurate p-multigrid discontinuous Galerkin solution of the Euler equations*, Int. J. Numer. Meth. Fluids, 60, pp. 847–865 (2009).

[2] B. Cockburn and C.-W. Shu, *RungeKutta Discontinuous Galerkin Methods for Convection-Dominated Problems*, J.Sci. Comput., 16(3), pp. 173-261 (2001).

[3] K.J. Fidkowski, T.A. Oliver, J. Lu and L. Darmofal, *p-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations*, J. Comput. Phys., 207, pp. 92–113 (2005).

[4] W. Hackbusch, *Multi-grid methods and applications*, Springer-Verlag, Berlin, 1985.

[5] W. Hackbusch and U. Trottenberg, editors, Multigrid methods : proceedings of the conference held at Köln-Porz, November 23–27, 1981, Springer-Verlag, Berlin, 1982.

[6] J.S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods. Algorithms, Analysis, and Applications*, Springer-Verlag, New York, 2008.

[7] C.M. Klaij, M.H. van Raalte, H. van der Ven and J.J.W. van der Vegt, *h-Multigrid for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations*, J. Comput. Phys., 227, pp. 1024–1045 (2007).

[8] H. Luo, J.D. Baum and R. Löhner, *A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids*, J. Comput. Phys. 211, pp. 767–783 (2006).

[9] B.S. Mascarenhas, B.T. Helenbrook and H.L. Atkins, *Application of p-multigrid to discontinuous Galerkin formulations of the Euler equations*, AIAA J., 47, pp. 1200–1208 (2009).

[10] B.S. Mascarenhas, B.T. Helenbrook and H.L. Atkins, *Coupling p-multigrid to geometric multigrid for discontinuous Galerkin formulations of the convection-diffusion equation*, J. Comput. Phys., 229, pp. 3664–3674 (2010).

[11] C.R. Nastase and D.J. Mavriplis, *High-order discontinuous Galerkin methods using an hp-multigrid approach*, J. Comput. Phys. 213, pp. 330–357 (2006).

[12] S. Rhebergen, J.J.W. van der Vegt, J.J.W. and 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, pp. 257-269 (2010).

[13] U. Trottenberg, C.W. Oosterlee and A. Schüller, *Multigrid*, Academic Press, London, 2001.

[14] J.M. Varah, *On the solution of block-tridiagonal systems arising from certain finite difference equations*, Mat. Comp., 26(120), pp. 859868 (1972).

[15] J.J.W. van der Vegt and 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, pp. 546–585 (2002).

[16] J.J.W. van der Vegt and S. Rhebergen, HP-Multigrid as Smoother Algorithm for Higher Order Discontinuous Galerkin Discretizations of Advection Dominated Flows. Part I. Multilevel Analysis, `http://eprints.eemcs.utwente.nl/`.

[17] P. Wesseling, *An introduction to multigrid methods.* Wiley, Chicester, 1991.

[18] R. Wienands and W. Joppich, *Practical Fourier analysis for multigrid methods*, Chapman & Hall/CRC, 2005.