Centrum voor Wiskunde en Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

Runge-Kutta multigrid analysis for space-time discontinuous Galerkin discretization of an advection-diffusion equation

M.H. van Raalte, C.M. Klaij

REPORT MAS-R0610 JUNE 2006

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Runge-Kutta multigrid analysis for space-time discontinuous Galerkin discretization of an advection-diffusion equation

ABSTRACT

In this article, we analyse the convergence of multigrid (MG) iteration for solving the algebraic equations arising from a space-time discontinuous Galerkin (DG) discretization of the advection-diffusion equation. To keep the MG method fully explicit, we consider Runge-Kutta smoothers that solve the algebraic equations by marching in pseudo-time to steady state. Depending on the Péclet number, we find multigrid convergence factors between 0.50 and 0.74 with Fourier two-level analysis. We illustrate the analysis with a numerical example.

2000 Mathematics Subject Classification: 65M12, 65M55

Keywords and Phrases: space-time discontinuous Galerkin method, pseudo time stepping, multigrid

Runge-Kutta Multigrid Analysis for Space-time Discontinuous Galerkin Discretization of an Advection-Diffusion Equation

M. H. van Raalte *   C. M. Klaij †

9th June 2006

Abstract

In this article, we analyse the convergence of multigrid (MG) iteration for solving the algebraic equations arising from a space-time discontinuous Galerkin (DG) discretization of the advection-diffusion equation. To keep the MG method fully explicit, we consider Runge-Kutta smoothers that solve the algebraic equations by marching in pseudo-time to steady state. Depending on the Péclet number, we find multigrid convergence factors between 0.50 and 0.74 with Fourier two-level analysis. We illustrate the analysis with a numerical example.

1 Introduction

In this analysis we study the convergence of multigrid (MG) iteration for solving a time-dependent advection-diffusion equation that is discretized by a second order accurate discontinuous Galerkin (DG) method both in the spatial direction and in the time direction. In this technique we discretize the time variable as a spatial variable and hence decoupling in space and time is avoided. A consequence of such a space-time approach is that the resulting methods can handle problems with moving and/or deforming meshes [8, 16].

*Centrum voor Wiskunde en Informatica, P.O. Box 94079, NL-1090 GB Amsterdam, The Netherlands

†University of Twente, Department of Applied Mathematics, P.O. Box 217, 7500 AE Enschede, The Netherlands

Moreover, the space-time approach applied to the incompressible Navier-Stokes equations does not suffer from the second order barrier in time as is the case in the classical pressure-correction methods. In theory, solutions of arbitrary order of accuracy can be computed. In this framework and with in mind the flexibility in mesh (h) refinement techniques and flexibility of polynomial order (p) adaptation of the approximation, we study the application of the DG method where the second order term is based on the method by Brezzi e.a [2, 14].

Discontinuous Galerkin methods, although originally developed for hyperbolic equations, are nowadays becoming the methods of interest for solving second order partial differential equations. These were unified and analysed for the Laplace equation by Arnold e.a. [1] and, among the techniques which proved consistent, adjoint consistent and stable with optimal error bounds is the internal penalty method and the method by Brezzi e.a [2]. The advantage of the latter is that the penalty term does not need a grid-dependent parameter as is the case for the internal penalty method. This is important for non-uniform grids. Its disadvantage is that the penalty term requires expensive lifting operators.

For solving the underlying system of algebraic equation we want to rely on multigrid iteration because of its expected optimal efficiency [15, 18]. However, in contrast to the block iterative schemes considered in [5, 7, 4], we study the application of explicit Runge-Kutta smoothing. For computationally demanding boundary value problems, the use of explicit multigrid schemes is preferable, in order to minimise the amount of data storage. To compute the MG convergence rates we introduce a similar two-level local mode Fourier analysis as described in [5, 17]. The difference in this analysis is that we avoid the cell-staggering problem of transferring cell data from coarse to fine cells, by associating the data in a cell with a nodal point. The resulting analysis can be used for arbitrary polynomial basis and is directly extendable to higher-dimensional problems by the tensor product principle [6]. For various cell Péclet numbers we compute MG convergence rates and we find that explicit Runge-Kutta iteration is efficient for solving time-dependent advection-diffusion equation.

The outline of this paper is as follows. In Section 2, we will describe the space-time DG discretization of the advection-diffusion equation by giving the weak form, the system of algebraic equations and the Runge-Kutta smoothers for the multigrid algorithm. In Section 3, we present the Fourier analysis needed to determine the convergence behaviour of the multigrid algorithm and we show the results for various situations in Section 4.

2 Space-time discontinuous Galerkin method

2.1 Weak form

In this section we introduce the space-time discretization for which we study multigrid convergence. For that purpose we introduce the open space-time domain

$$\Omega = \left\{(x,t) \mid x \in \mathbb{R},\ t \in \mathbb{R}^+\right\},$$

with boundary $\partial\Omega = \{(x,t) \mid t = 0,\ x \in \mathbb{R}\}$ and consider the advection-diffusion equation in generic form

$$-\nabla \cdot \mathrm{A}\nabla u + \mathrm{b} \cdot \nabla u = 0, \quad \text{in } \Omega, \quad u = u_0 \text{ on } \partial\Omega, \tag{1}$$

where

$$\nabla = \begin{bmatrix}\partial_x & \partial_t\end{bmatrix}^T, \quad \mathrm{b} = \begin{bmatrix}a & 1\end{bmatrix}, \quad \mathrm{A} = \begin{bmatrix}d & 0\\ 0 & 0\end{bmatrix}, \tag{2}$$

with $a > 0$ and $d > 0$. Next, to arrive at a DG discretization, we partition the domain Ω in regular rectangular cells of identical shape

$$\Omega_j^n = \{(x,t) \mid jh < x < (j+1)h,\ n\Delta t < t < (n+1)\Delta t\}, \tag{3}$$

with $h > 0$, $\Delta t > 0$, $j \in \mathbb{Z}$ and $n \in \{0, \mathbb{N}\}$. Then $\Omega^n = \bigcup_{j\in\mathbb{Z}} \overline{\Omega_j^n}$ is a space-time slab. To find the discrete algebraic equations we introduce the discrete function space

$$\mathcal{S}_h^n = \left\{ v_h^n \in L^2(\Omega) \;\middle|\; v_h^n|_{\Omega_j^n} \in \mathcal{P}^k(\Omega_j^n),\ \forall j \in \mathbb{Z},\ n \in \mathbb{N} \right\}, \tag{4}$$

the space of piecewise polynomials of degree at most k in the coordinate directions. Then the discrete weak DG form of (1) reads [2, 9, 12]: find $u_h^n \in \mathcal{S}_h^n$ such that

$$B(u_h^n, v_h^n) = 0, \quad \forall v_h \in \mathcal{S}_h^n,\ \forall n \in \mathbb{N}, \tag{5}$$

with

$$\begin{aligned}
&B(u_h^n, v_h^n) = \\
&\sum_{j\in\mathbb{Z}} \int_{\Omega_j^n} (\mathrm{A}\nabla u_h^n) \cdot \nabla v_h^n \, d\Omega - \sum_{j\in\mathbb{Z}} \int_{\Gamma_j^n} \langle \mathrm{A}\nabla u_h^n \rangle \cdot [v_h^n] \, ds \\
&- \sum_{j\in\mathbb{Z}} \int_{\Gamma_j^n} \langle \mathrm{A}\nabla v_h^n \rangle \cdot [u_h^n] \, ds + \sum_{e\in\mathbb{Z}} \sum_{j\in\mathbb{Z}} \int_{\Omega_j^n} \eta_e (\mathrm{r}_e(u_h^n)\,\mathrm{A}) \cdot \mathrm{r}_e(v_h^n) \, d\Omega \\
&- \sum_{j\in\mathbb{Z}} \int_{\Omega_j^n} \nabla v_h^n \cdot \mathrm{b}\, u_h^n \, d\Omega + \sum_{j\in\mathbb{Z}} \int_{\Gamma_j^{n-}} u_h^{n-} \mathrm{n}_j \cdot \mathrm{b}\, v_h^{n+} \, ds \\
&+ \sum_{j\in\mathbb{Z}} \int_{\Gamma_j^{n+}} u_h^{n+} \mathrm{n}_j \cdot \mathrm{b}\, v_h^{n+} \, ds,
\end{aligned} \tag{6}$$

and A and b defined in (2). Here, the first four terms in the righthand side are associated with the diffusion part of (1), where the term with stabilisation parameter $\eta_e > 0$ prevents the discrete system of being indefinite. In the next section, we show how to compute this stabilisation term. For a complete overview of DG methods for elliptic problems we refer to [1]. The common cell interface between two adjacent cells Ω^n_{j-1} and Ω^n_j in the time slab Ω^n is $\Gamma^n_j = \partial\Omega^n_{j-1} \cap \partial\Omega^n_j$. On this interface the jump operator $[\,\cdot\,]$ and the average operator $\langle\,\cdot\,\rangle$ are defined by

$$\begin{aligned}
[u^n_h(x,t)] &= u^n_h(x,t)|_{\partial\Omega^n_{j-1}}\mathrm{n}_{j-1} + u^n_h(x,t)|_{\partial\Omega^n_j}\mathrm{n}_j, \quad \text{for } u^n_h \in \mathcal{S}^n_h, \\
\langle\tau^n_h(x,t)\rangle &= \frac{1}{2}\left(\tau^n_h(x,t)|_{\partial\Omega^n_{j-1}} + \tau^n_h(x,t)|_{\partial\Omega^n_j}\right), \quad \text{for } \tau^n_h \in [\mathcal{S}^n_h]^2,
\end{aligned} \tag{7}$$

with $x \in \Gamma^n_j$ and with n_j the unit outward normal of cell Ω^n_j. Furthermore we distinguish between inflow and outflow boundaries of $\partial\Omega^n_j = \Gamma^{n-}_j \cup \Gamma^{n+}_j$. With Γ^{n-}_j we denote the inflow boundary part. Here is $\mathrm{n}_j \cdot \mathrm{b} < 0$. The outflow boundary is denoted by Γ^{n+}_j, i.e., $\mathrm{n}_j \cdot \mathrm{b} \geq 0$. The traces $u^{n\pm}_h$ at $\partial\Omega^n_j$ are defined by

$$u^{n\pm}_h = \lim_{\varepsilon\uparrow 0} u^n_h(x \pm \varepsilon \mathrm{n}_x, t \pm \varepsilon n_t), \tag{8}$$

with $\mathrm{n}_j = \begin{bmatrix} n_x & n_t \end{bmatrix}^T$. Notice that, because of the causality in time, $u^{0-}_h|_{\partial\Omega} = u_0$ and that $u^{n-}_h(x, n\Delta t) = u^{n-1}_h(x, n\Delta t)$. So for each time slab Ω^n we have to solve a system of algebraic equations. To explicitly describe the iterative methods studied in this paper, in the next section, we provide $\mathcal{S}^n_h$ with a polynomial space and we give the discrete stencils associated with (6).

2.2 Discrete system

Here, we describe the linear system that must be solved for each time slab. For sake of clarity, in this presentation we restrict ourselves to a second order discretization although the analysis can be extended to higher order and multiple dimensions.

On the unit square $(\xi,\ \eta) \in (0,\ 1) \times (0,\ 1)$ we take the following polynomial space

$$\phi_0(\xi,\eta) = 1, \quad \phi_1(\xi,\eta) = 2\xi - 1, \quad \phi_2(\xi,\eta) = 2(\eta - 1), \tag{9}$$

yielding the approximation

$$u^n_h = \sum_{j\in\mathbb{Z}}\sum_{k=0}^{2} c^n_{j,k}\phi^n_{j,k}(x,t) \equiv \sum_{j\in\mathbb{Z}}\sum_{k=0}^{2} c^n_{j,k}\phi_k\left(\frac{x - jh}{h}, \frac{t - n\Delta t}{\Delta t}\right). \tag{10}$$

This polynomial basis is of interest because of two reasons: the basis functions are chosen such that the test and trial function can be split into an element mean $\bar{u}_h$ at $t = t_{n+1}$ and a fluctuating part $\tilde{u}_h$ [16]:

$$u_h(x,t) = \bar{u}_h + \tilde{u}_h(x,t), \quad \forall x, t \in \Omega_j^n$$

with $\bar{u}_h = c_{j,0}$ and

$$\int_{x\in\Omega_j^n} \tilde{u}_h(x, t_{n+1})\, dx = 0.$$

As a consequence the relation between DG and finite volume discretizations is exposed: the equations for the element mean in the space-time DG discretization are the same as those of a finite volume discretization. The second reason is that it suits the definition of the artificial dissipation operator used in [16] as an alternative for slope limiters to guarantee monotone solutions around discontinuities and sharp gradients.

To compute the penalty term in (6), we consider its definition in variational form [2]: find $\mathrm{r}_e(v_h^n) \in [\mathcal{S}_h^n]^2$ such that

$$\sum_{j\in\mathbb{Z}} \int_{\Omega_j} \mathrm{r}_e(v_h^n) \cdot \tau_h^n \, d\Omega = \int_{\Gamma_e^n} [v_h^n] \cdot \langle \tau_h^n \rangle \, ds, \quad \forall \tau_h^n \in [\mathcal{S}_h^n]^2, \quad e \in \mathbb{Z}. \tag{11}$$

Since $[\mathcal{S}_h^n]^2 = \left[\mathrm{Span}\left\{\phi_{j,k}^n\right\}\right]^2$ and because $\mathrm{r}_e = \left[(\mathrm{r}_e)_x \quad (\mathrm{r}_e)_t\right]^T$ is a polynomial expansion we take

$$(\mathrm{r}_e)_* = \sum_{j\in\mathbb{Z}} \sum_{k=0}^{2} \left(a_{j,k}^n\right)_* \phi_{j,k}^n \,, \quad * = x, t, \tag{12}$$

with $2 \times 3\mathbb{Z}$ unknowns $\left(a_{j,k}^n\right)_*$. Taking the same number of test functions

$$\tau_h^n \in \left\{ \begin{bmatrix} \phi_{j,k}^n \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \phi_{j,k}^n \end{bmatrix} \right\}, \quad j \in \mathbb{Z},\ 0 \le k \le 2, \tag{13}$$

we find $\mathrm{r}_e(\phi_{\tilde{j},\tilde{k}}^n)$ and hence $\mathrm{r}_e(u_h^n)$ by solving the small linear system for the unknowns $\left(a_{j,k}^n\right)_x$

$$\sum_{j=e-1}^{e} \sum_{k=0}^{2} \left(a_{j,k}^n\right)_x \int_{\Omega_j} \phi_{j,k}^n \phi_{j,l}^n \, d\Omega = \int_{\Gamma_e^n} \left[\phi_{\tilde{j},\tilde{k}}^n\right] \cdot \left\langle \begin{bmatrix} \phi_{j,k}^n \\ 0 \end{bmatrix} \right\rangle ds, \tag{14}$$

with $\tilde{j} \in \{e-1, e\}$ and $l, \tilde{k} \in \{0,1,2\}$, while $(a_{j,k}^n)_x = 0$ for all $j \in \mathbb{Z}/\{e-1, e\}$ and $(a_{j,k}^n)_t = 0$ for all $j \in \mathbb{Z}$. So with the approximation (10) and with the definition of the lifting functions (12) and (14), the discrete system (5) is $3\mathbb{Z}$ block-Toeplitz. The corresponding operator is given by the associated stencils

$$\mathcal{L}^n_{d_h} \cong \frac{d\Delta t}{h}\left[\begin{array}{ccc|ccc|ccc} -2\eta & 1-2\eta & 2\eta & 4\eta & 0 & -4\eta & -2\eta & -1+2\eta & 2\eta \\ -1+2\eta & -2+2\eta & 1-2\eta & 0 & 4\eta & 0 & 1-2\eta & -2+2\eta & -1+2\eta \\ 2\eta & -1+2\eta & -\frac{13}{6}\eta & -4\eta & 0 & \frac{13}{3}\eta & 2\eta & 1-2\eta & -\frac{13}{6}\eta \end{array}\right] \tag{15}$$

for the diffusion part with $\eta_e = \eta$. The space-time advection stencil is given by

$$\mathcal{L}^n_{a_h} \cong \left[\begin{array}{ccc|ccc|ccc} -\sigma & -\sigma & \sigma & \sigma+h & \sigma & -\sigma & 0 & 0 & 0 \\ \sigma & \sigma & -\sigma & -\sigma & \sigma+\frac{1}{3}h & \sigma & 0 & 0 & 0 \\ \sigma & \sigma & -\frac{4}{3}\sigma & -\sigma-2h & -\sigma & \frac{4}{3}\sigma+2h & 0 & 0 & 0 \end{array}\right]. \tag{16}$$

with $\sigma = a\Delta t$. The stencil containing data of the previous time slab is given by

$$\mathcal{L}^{n-1}_{a_h} \cong \left[\begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & -h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{3}h & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2h & 0 & 0 & 0 & 0 & 0 \end{array}\right]. \tag{17}$$

With $\mathcal{L}^n_h = \mathcal{L}^n_{d_h} + \mathcal{L}^n_{a_h}$ and $f^n_h = \mathcal{L}^{n-1}_{a_h} u^{n-1}_h$ we have to solve for each time slab Ω^n the $3\mathbb{Z} \times 3\mathbb{Z}$ linear system

$$\mathcal{L}^n_h u^n_h = f^n_h. \tag{18}$$

This system will be solved by multigrid iteration combined with Runge-Kutta smoothers and the resulting convergence behaviour will be analysed.

2.3 Runge-Kutta smoothers

In order to reduce the computational costs when handling complex higher dimensional problems, we are interested in fully explicit iterative solvers. For that purpose, we write the system of equations (18) as a system of ordinary differential equations that we want to iterate towards steady state. Hence we consider the problem

$$\frac{dc^n_h}{d\tau} = f^n_h - A^n_h c^n_h, \tag{19}$$

for expansion coefficients c^n_h of u^n_h with A^n_h the $3\mathbb{Z} \times 3\mathbb{Z}$ block-Toeplitz matrix associated with the operator $\mathcal{L}^n_h$ in (18). The first Runge-Kutta method used for this purpose is

Algorithm 1 (EXI). Explicit Runge-Kutta method for inviscid flow with Melson [11] correction and pseudo-time step $\Delta\tau$.

1. Set $(c_h^n)_0^k = (c_h^n)^k$.

2. For all stages $s = 1$ to 5 compute $(c_h^n)_s^k$ as

$$(c_h^n)_s^k = \frac{1}{1+\alpha_s\Delta\tau}\left((c_h^n)_0^k + \alpha_s\Delta\tau\left((c_h^n)_{s-1}^k + f_h^n - A_h^n(c_h^n)_{s-1}^k\right)\right).$$

3. Return $(c_h^n)^{k+1} = (c_h^n)_5^k$.

Here the Runge-Kutta coefficients are $\alpha_1 = 0.0791451$, $\alpha_2 = 0.163551$, $\alpha_3 = 0.283663$, $\alpha_4 = 0.5$ and $\alpha_5 = 1.0$. In [8], the performance of the EXI method was analysed for the space-time DG discretization of the advection-diffusion equation. When diffusion dominates, the stability condition proved quite restrictive. To alleviate this restriction, a member of a family of Runge-Kutta methods proposed by Kleb e.a. [10] was used. It has a stability domain which stretches much further along the negative real axis than classical Runge-Kutta schemes, making it ideal for diffusion dominated flow problems. The method is given by:

Algorithm 2 (EXV). Explicit Runge-Kutta method for viscous flows with pseudo-time step $\Delta\tau$.

1. Set $(c_h^n)_0^k = (c_h^n)^k$.

2. For all stages $s = 1$ to 4 compute $(c_h^n)_s^k$ as

$$(c_h^n)_s^k = (c_h^n)_0^k + \alpha_s\Delta\tau\left(f_h^n - A_h^n(c_h^n)_{s-1}^k\right).$$

3. Return $(c_h^n)^{k+1} = (c_h^n)_4^k$.

For this iteration scheme the Runge-Kutta coefficients are $\alpha_1 = 0.0178571$, $\alpha_2 = 0.0568106$, $\alpha_3 = 0.1745130$, $\alpha_4 = 1.0$. In [8], the EXI and EXV method were combined based on the cell Reynolds number: the EXV method is used for elements with low cell Reynolds numbers (i.e. boundary layers) and the EXI method for high cell Reynolds numbers (i.e. the far-field). This approach proved very effective for the 3D compressible Navier-Stokes equations. Therefore, we study the smoothing properties of these two explicit Runge-Kutta methods. The error amplification operators associated with these Runge-Kutta methods are needed to assess the smoothing property. These operators for the error $(e_h^n)^{k+1} \equiv (c_h^n)^{k+1} - (c_h^n)^k$ are given by:

$$\begin{aligned} M_h^{\mathrm{EXI}} = {} & \frac{I_h}{1+\alpha_5\Delta\tau} + \frac{\alpha_5\Delta\tau\left(I_h - A_h^n\right)}{\left(1+\alpha_4\Delta\tau\right)\left(1+\alpha_5\Delta\tau\right)} + \cdots \\ & + \frac{\alpha_2\alpha_3\cdots\alpha_5\left(\Delta\tau\left(I_h - A_h^n\right)\right)^4}{\left(1+\alpha_1\Delta\tau\right)\left(\cdots\right)\left(1+\alpha_5\Delta\tau\right)} + \frac{\alpha_1\alpha_2\cdots\alpha_5\left(\Delta\tau\left(I_h - A_h^n\right)\right)^5}{\left(1+\alpha_1\Delta\tau\right)\left(\cdots\right)\left(1+\alpha_5\Delta\tau\right)}. \end{aligned} \tag{20}$$

and

$$M_h^{\mathrm{EXV}} = I_h - \alpha_4 \Delta\tau A_h^n + \alpha_3\alpha_4 \left(\Delta\tau A_h^n\right)^2 - \cdots + \alpha_1\alpha_2\alpha_3\alpha_4 \left(\Delta\tau A_h^n\right)^4 . \quad (21)$$

2.4 Multigrid

At the core of any multigrid algorithm is the two-level scheme. Multilevel algorithms are obtained by recursively applying the two-level scheme in, for example, a V-cycle. Therefore, we study the error amplification operator of the two-level algorithm M_h^{TLA}, which is given by [5, 17]:

$$M_h^{\mathrm{TLA}} = M_h^{\mathrm{CGC}} M_h^{\mathrm{REL}},$$

with M_h^{REL} the error amplification operator associated with either the EXV or the EXI scheme, given in (21) and (20). The coarse grid correction operator is defined as

$$M_h^{\mathrm{CGC}} = I_h - P_{hH} \left(\mathcal{L}_H^n\right)^{-1} \overline{R}_{Hh} \mathcal{L}_h^n, \quad (22)$$

with $\mathcal{L}_H^n$ the system obtained by the space-time DG discretization for the time slab Ω^n on the coarse grid with $H = 2h$. The prolongation and restriction operators P_{hH} and $\overline{R}_{Hh}$ are based on the embedding of the spaces $S_H^n \subset S_h^n$ and will be given in the next section.

Remark. Contrary to the internal penalty method, the discretization of the second order term (based on the method by Brezzi e.a. [1, 2]) only satisfies the Galerkin property ($\mathcal{L}_H = \overline{R}_{Hh} \mathcal{L}_h P_{hH}$) if the stabilisation parameter η_e on the coarse mesh is a factor H/h larger than on the fine mesh (3). In general this property does not hold, e.g., on non-uniform meshes. Therefore we take the same stabilisation parameter on the fine and coarse mesh. For stability of the discretization we take $\eta_e = 2$; equal to the number of spatial faces per cell [14].

The convergence behaviour of the two-level algorithm for the space-time DG discretization is given by the spectral radius of the error amplification operator $\rho\left(M_h^{\mathrm{TLA}}\right)$ which represents the expected convergence factor per iteration. In the next section, we will apply Fourier analysis to compute the eigenvalue spectra of the two-level algorithm.

3 Fourier analysis

3.1 Grid functions and the space-time block Toeplitz operator

To study the convergence of the various iterative methods we introduce two-level Fourier analysis tools for the unknowns in the cells Ω_j^n. The key part in this analysis is to associate the coefficients $\left\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\right\}_{j\in\mathbb{Z}}$ of the approximation (10) in the system (18) with the nodal points jh. In this way we avoid the staggering problem of transferring cell data from coarse to fine cells [5, 17], while we keep the data in cell wise ordering. Hence, we introduce an elementary mode $e_h(\omega) = e^{\mathrm{i}jh\omega}$, with $\omega \in \mathbb{T}_h = [-\pi/h, \pi/h)$ on the space-time grid

$$\mathbb{Z}_h^n = \{(jh, n\Delta t) \mid j \in \mathbb{Z},\ n \in \mathbb{N},\ h > 0,\ \Delta t > 0\}. \tag{23}$$

If we decompose $\mathcal{L}_h^n$ into a strict block-lower, a block-diagonal and a strict block-upper matrix, where

$$\mathcal{L}_h^n \cong \begin{bmatrix} L_h^n & D_h^n & U_h^n \end{bmatrix}, \tag{24}$$

we compute the Fourier transform by

$$\widehat{\mathcal{L}_h^n}(\omega) = L_h^n e^{-\mathrm{i}\omega h} + D_h^n + U_h^n e^{+\mathrm{i}\omega h}. \tag{25}$$

Then, following [5, 17], we find in the eigenvalue-eigenvector decomposition

$$\widehat{\mathcal{L}_h^n}(\omega)\mathrm{v} = \mathrm{v}\Lambda_h^n(\omega), \quad \omega \in \mathbb{T}_h, \tag{26}$$

that $\Lambda_h^n(\omega)$ is a 3×3 diagonal matrix with the eigenvalues $\lambda_i(\omega)$ of $\mathcal{L}_h^n$ as function of $\omega \in \mathbb{T}_h$. The columns of $\mathrm{v} = [\mathrm{v}_0, \mathrm{v}_1, \mathrm{v}_2]$ are the coefficients of the eigenvector $v_i e^{\mathrm{i}jh\omega}$ of $\mathcal{L}_h^n$. We see that this eigenvector is a three-valued grid function on the grid (23) in the coefficient ordering $\left\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\right\}_{j\in\mathbb{Z}}$.

In the next section we introduce the grid transfer operators that are needed to construct the two-level algorithm.

3.2 Prolongation and restriction

Important ingredients in the two-level analysis are the flat prolongation and the flat restriction operator. Any constant coefficient grid transfer operator is a combination of a Toeplitz and a flat grid transfer operator. In this section we introduce the grid transfer operators and the Fourier transforms for grid functions $\left\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\right\}_{j\in\mathbb{Z}}$ that are needed in the two-level analysis.

To avoid the data staggering problem related to the grid transfer operators acting on cell-wise data [5], it is necessary for this analysis to associate the cell data with the nodal points x_j. In this way, we can obtain vector valued grid functions $\mathbf{u}_h^n(jh) = \left\{c_{j,0}^n, c_{j,1}^n, c_{j,2}^n\right\}_{j\in\mathbb{Z}}$ in the Hilbert space $\left[l^2(\mathbb{Z}_h^n)\right]^3$ for which the grid transfer operators are easily defined. For such a grid function the flat prolongation $P_{hH}^0 : \left[l^2(\mathbb{Z}_H^n)\right]^3 \to \left[l^2(\mathbb{Z}_h^n)\right]^3$ is defined by

$$P_{hH}^0 \mathbf{u}_H^n(jH) = \begin{cases} \mathbf{u}_H^n(\frac{j}{2}H) & \text{if } j \text{ even}, \\ \mathbf{0} & \text{if } j \text{ odd}. \end{cases} \tag{27}$$

The flat restriction operator $R^0_{hH} : \left[l^2(\mathbb{Z}^n_h)\right]^3 \to \left[l^2(\mathbb{Z}^n_H)\right]^3$ is defined by

$$(R^0_{Hh}\mathbf{u}^n_h)(jH) = \mathbf{u}^n_h(2jh). \tag{28}$$

Then the prolongation $P_{hH} : \mathcal{S}^n_H \to \mathcal{S}^n_h$ so that $P_{hH}u^n_H(x) = u^n_H(x)$ for all $x \in \mathbb{R} \setminus \mathbb{Z}^n_h$ is uniquely defined by $P_{hH} = P_h P^0_{hH}$, where

$$P_h \cong \left[\begin{array}{ccc|ccc|ccc} 1 & \frac{1}{2} & 0 & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right]. \tag{29}$$

Another unique operator that is needed in the two-level analysis is the restriction on the residual. It is the adjoint of the prolongation operator and is given by

$$\overline{R}_{Hh} = R^0_{Hh} P^T_h. \tag{30}$$

Then the Fourier transforms are computed from [5, 17]

$$\widehat{P_{hH}\mathbf{u}^n_H}(\omega) = \left(P_h \widehat{P^0_{hH}\mathbf{u}^n_H}\right)(\omega) = \frac{1}{2}\begin{bmatrix} \widehat{P_h}\,(\omega) \\ \widehat{P_h}\left(\omega + \frac{\pi}{h}\right) \end{bmatrix} \widehat{\mathbf{u}^n_H}(\omega), \tag{31}$$

and with $\overline{R}_h = P^T_h$

$$\begin{aligned} \widehat{\overline{R}_{Hh}\mathbf{u}^n_h}(\omega) &= R^0_{Hh}\widehat{\overline{R}_h\mathbf{u}^n_h}(\omega) \\ &= \begin{bmatrix} \widehat{\overline{R}_h}\,(\omega) & \widehat{\overline{R}_h}\left(\omega + \frac{\pi}{h}\right) \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{u}^n_h}\,(\omega) \\ \widehat{\mathbf{u}^n_h}\left(\omega + \frac{\pi}{h}\right) \end{bmatrix}, \end{aligned} \tag{32}$$

with $\omega \in \mathbb{T}_H = [-\pi/H, \pi/H)$ and

$$\widehat{\mathbf{u}^n_h}(\omega) = \frac{h}{\sqrt{2\pi}} \sum_{j\in\mathbb{Z}} e^{-\mathrm{i}jh\omega}\mathbf{u}^n_h(jh).$$

A sketch of the basic two-level set up is shown in Figure 1. Here we see the grid function $\mathbf{u}^n_h(jh)$ on the space-time grid; a part of the coarse grid and pictorially the action of the grid transfer operators. With these tools we construct the Runge-Kutta two-level analysis in the next section.

3.3 Two-level algorithm

The eigenvalue spectra of the two-level algorithm M^{TLA}_h is shown [17] to be $\{\lambda_i(\omega)\}$ with $i = 1, \ldots, 6$ and $\lambda_i(\omega)$ the eigenvalues of the Fourier transform $\widehat{M^{\mathrm{TLA}}_h}$ for $\omega \in \mathbb{T}_H$. The Fourier transform of the two-level operator reads:

$$\widehat{M^{\mathrm{TLA}}_h}(\omega) = \widehat{M^{\mathrm{CGC}}_h}\widehat{M^{\mathrm{REL}}_h}(\omega),$$

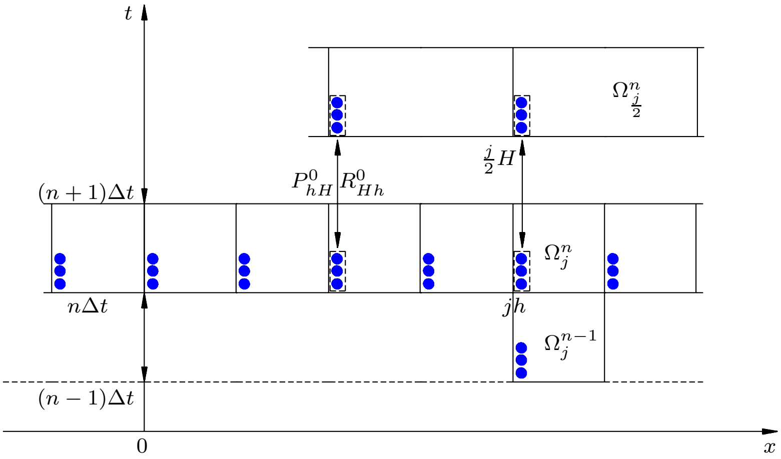


Figure 1: A sketch of the grid function $\mathbf{u}_h^n(jh)$ on the space-time grid (23) and a part of the coarse mesh. Pictorially the action (31) of the flat prolongation and the flat restriction (32) is shown.

with $\widehat{M_h^{\mathrm{REL}}}$ the Fourier transform of the EXI (or EXV) Runge-Kutta method and $\widehat{M_h^{\mathrm{CGC}}}$ of the coarse grid correction. For the operator $\widehat{\mathcal{L}_h^n}$ (see (25)), the Fourier transforms of the EXI and EXV error amplification operators are

$$\widehat{M_h^{\mathrm{EXI}}}(\omega) = \frac{I_h}{1+\alpha_5\Delta\tau} + \frac{\alpha_5\Delta\tau\left(I_h - \widehat{\mathcal{L}_h^n}(\omega)\right)}{(1+\alpha_4\Delta\tau)\,(1+\alpha_5\Delta\tau)} + \cdots$$
$$+ \frac{\alpha_2\alpha_3\cdots\alpha_5\left(\Delta\tau\left(I_h - \widehat{\mathcal{L}_h^n}(\omega)\right)\right)^4}{(1+\alpha_1\Delta\tau)\,(\cdots)\,(1+\alpha_5\Delta\tau)} + \frac{\alpha_1\alpha_2\cdots\alpha_5\left(\Delta\tau\left(I_h - \widehat{\mathcal{L}_h^n}(\omega)\right)\right)^5}{(1+\alpha_1\Delta\tau)\,(\cdots)\,(1+\alpha_5\Delta\tau)},$$

and

$$\widehat{M_h^{\mathrm{EXV}}}(\omega) = I_h - \alpha_4\Delta\tau\widehat{\mathcal{L}_h^n}(\omega) + \alpha_3\alpha_4\left(\Delta\tau\widehat{\mathcal{L}_h^n}(\omega)\right)^2 - \cdots + \alpha_1\alpha_2\alpha_3\alpha_4\left(\Delta\tau\widehat{\mathcal{L}_h^n}(\omega)\right)^4,$$

with $\omega \in \mathbb{T}_h$. Using the Fourier transform of the block Toeplitz operator and of the prolongation and restriction operators the Fourier transform of the two-level error amplification operator is given by (see (22))

$$\widehat{M_h^{\mathrm{TLA}}}(\omega) = \begin{bmatrix} I_h & 0 \\ 0 & I_h \end{bmatrix} - \begin{bmatrix} \widehat{P_h}\,(\omega) \\ \widehat{P_h}\,(\omega + \frac{\pi}{h}) \end{bmatrix} \left[\widehat{\mathcal{L}_H}(\omega)^{-1}\right]$$
$$\left[\widehat{R_h}\,(\omega) \quad \widehat{R_h}\,(\omega + \tfrac{\pi}{h})\right] \begin{bmatrix} \widehat{\mathcal{L}_h}(\omega) & 0 \\ 0 & \widehat{\mathcal{L}_h}(\omega + \frac{\pi}{h}) \end{bmatrix} \begin{bmatrix} \widehat{M_h^{\mathrm{REL}}}(\omega) & 0 \\ 0 & \widehat{M_h^{\mathrm{REL}}}(\omega + \frac{\pi}{h}) \end{bmatrix},$$

with I_h the 3×3 identity matrix and $\omega \in \mathbb{T}_H$.

Table 1: TLA convergence with EXI smoothing is preferable in the advection dominated case (denoted by $*$).

physics		stability	convergence	
$\mathrm{C}_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho\left(M_h^{\mathrm{EXI}}\right)$	$\rho\left(M_h^{\mathrm{TLA}}\right)$
100	100	$1.8 \cdot 10^{-4}$	0.991	0.623^*
100	0.01	$1.0 \cdot 10^{-7}$	0.999	0.959
1	100	1.6	0.796	0.479^*
1	0.01	$1.0 \cdot 10^{-3}$	0.999	0.957

Table 2: TLA convergence with EXV smoothing is preferable in the diffusion dominated case (denoted by $*$).

physics		stability	convergence	
$\mathrm{C}_{\Delta t}$	Pe_h	$\Delta\tau/\Delta t$	$\rho\left(M_h^{\mathrm{EXV}}\right)$	$\rho\left(M_h^{\mathrm{TLA}}\right)$
100.0	100.0	$2.0 \cdot 10^{-5}$	0.999	0.914
100.0	0.01	$8.0 \cdot 10^{-7}$	0.999	0.744^*
1.0	100.0	1.0	0.924	0.660
1.0	0.01	$8.0 \cdot 10^{-3}$	0.993	0.744^*

4 Results

4.1 Convergence of the two-level algorithm

In this section, the eigenvalue spectra and radii of the two-level algorithm are given for various situations, described by the Courant and Péclet numbers:

$$\mathrm{C}_{\Delta t} = \frac{a\Delta t}{h} \quad \text{and} \quad \mathrm{Pe}_h = \frac{ah}{d}.$$

The Courant number expresses the time-accuracy of the discretization and the Péclet number the importance of diffusion relative to advection. Since the space-time DG discretization is *implicit* in physical time, the method is unconditionally stable [14] for any physical time step. This allows us to take the Courant number $\mathrm{C}_{\Delta t} = 100$ for steady-state cases and $\mathrm{C}_{\Delta t} = 1$ for time-dependent cases. We will consider Péclet numbers $\mathrm{Pe}_h = 0.01$ and $\mathrm{Pe}_h = 100$, which represent the diffusion and advection dominated cases, respectively. This defines the following four flow regimes:

- steady, advection dominated: $\mathrm{C}_{\Delta t} = 100$, $\mathrm{Pe}_h = 100$
- steady, diffusion dominated: $\mathrm{C}_{\Delta t} = 100$, $\mathrm{Pe}_h = 0.01$
- unsteady, advection dominated: $\mathrm{C}_{\Delta t} = 1$, $\mathrm{Pe}_h = 100$

- unsteady, diffusion dominated: $\mathrm{C}_{\Delta t} = 1$, $\mathrm{Pe}_h = 0.01$

The Runge-Kutta methods are *explicit* in pseudo time and their stability is governed by the ratio between the pseudo and physical time step $\Delta\tau/\Delta t$.

In Tables 1 and 2, we give the spectral radii of the smoothers and the two-level algorithm for these cases. Both the EXI and EXV smoother are stable but hardly converge (except in the unsteady advection dominated case) which shows the necessity of multigrid iteration. With the two-level algorithm, the situation is considerably improved. Clearly, the EXI method is preferable in the advection dominated case. We find convergence factors of 0.62 and 0.48 for the steady and unsteady case respectively (Table 1). The EXV method is preferable in the diffusion dominated case, where we find convergence factors of 0.74 both in the steady and unsteady situation (Table 2).

In Figures 2, 3, 4 and 5, we show the eigenvalue spectra of the preferable smoother and the two-level algorithm for each case. For the smoothers we have plotted the eigenvalues corresponding to a discrete series of low frequencies $\omega_i = -\pi/2h, -0.96\pi/2h, \dots, \pi/2h$ and associated high frequencies $\omega_i + \pi/h$. The eigenvalues corresponding to low frequencies are denoted by $\circ$; those corresponding to high frequencies by $+$. The eigenvalue spectra of two-level algorithms are plotted for $\omega_i = -\pi/H, -0.96\pi/H, \dots, \pi/H$. Here we do not distinguish between low and high frequencies. The two-level algorithm must damp all frequencies.

Moreover, from these figures we see that the Runge-Kutta methods have the smoothing property, i.e., the high frequencies are damped. The observed smoothing factor of approximately 0.8 (which is often used as an estimate for the multigrid convergence [3]) is rather inaccurate in comparison to the true smoothing factor obtained with two-level analysis.

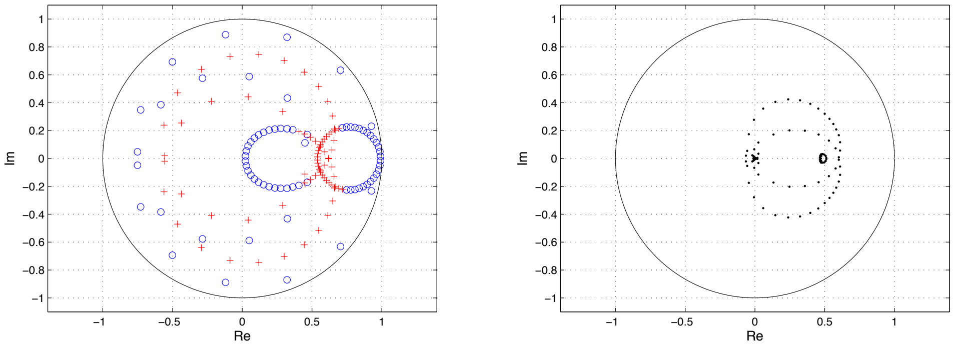
4.2 Numerical illustration

To illustrate the results of the multigrid analysis, we consider the space-time discretization of the scalar advection-diffusion equation for the following simple initial boundary value problem:

$$\begin{cases} u_t + au_x = du_{xx}, & x \in (0,1), \quad t \in \mathbb{R}^+, \\ u(0,t) = 1, \quad u(1,t) = 0, & t \in \mathbb{R}^+, \\ u(x,0) = 1 - x, & x \in (0,1). \end{cases}$$

The exact (steady state) solution is given by:

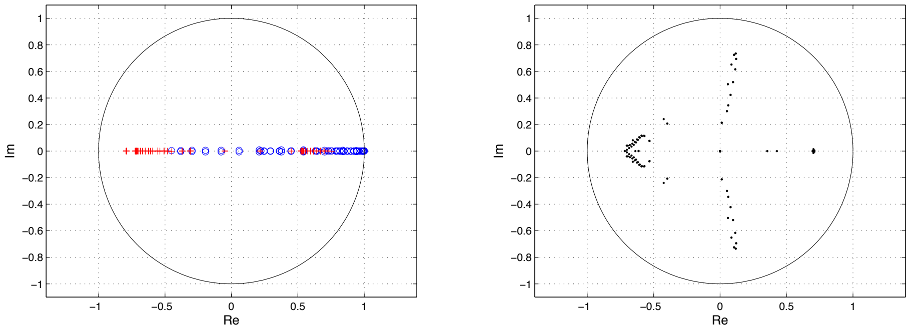
$$u(x) = \frac{e^{a/d} - e^{ax/d}}{e^{a/d} - 1},$$



(a) EXI

(b) TLA with EXI

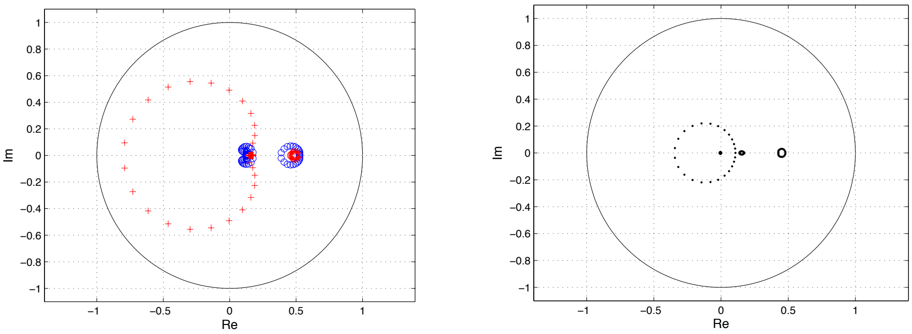
Figure 2: Eigenvalue spectra of the EXI smoother and two-level algorithm in the steady advection dominated case ($\mathrm{C}_{\Delta\tau} = 100$ and $\mathrm{Pe}_h = 100$).



(a) EXV

(b) TLA with EXV

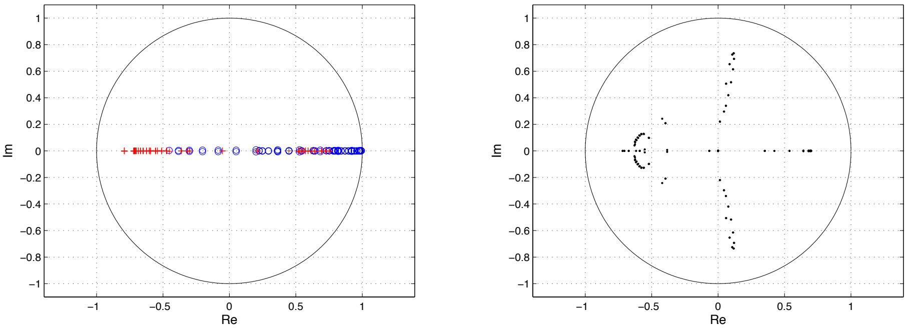
Figure 3: Eigenvalue spectra of the EXV smoother and two-level algorithm in the steady diffusion dominated case ($\mathrm{C}_{\Delta\tau} = 100$ and $\mathrm{Pe}_h = 0.01$).



(a) EXI

(b) TLA with EXI

Figure 4: Eigenvalue spectra of the EXI smoother and two-level algorithm in the unsteady advection dominated case ($C_{\Delta\tau} = 1$ and $Pe_h = 100$).



(a) EXV

(b) TLA with EXV

Figure 5: Eigenvalue spectra of the EXV smoother and two-level algorithm in the unsteady diffusion dominated case ($C_{\Delta\tau} = 1$ and $Pe_h = 0.01$).

and features an exponential boundary layer near $x = 1$. Such a case is best solved on a so-called Shishkin mesh [13]. With N elements, this mesh is piecewise equidistant with nodes x_j given by:

$$x_j = \begin{cases} 2(1-c)j/N & \text{for } j = 0, 1, \ldots, N/2, \\ 1 - c + 2c/N(j - N/2) & \text{for } j = N/2, N/2+1, \ldots, N, \end{cases}$$

where $c = (2/a)d\ln(N)$. For our example, we take $a = 1$, $d = 0.025$ and $N = 32$. Advection dominates in the first part, so we use the EXI scheme there and the EXV scheme in the second part. We use three level multigrid in a V-cycle with two pre- and post-relaxations. The coarse grid problem is solved approximately with four relaxations, which is more realistic in view of applications to complex problems where the exact coarse grid solution cannot be attained.

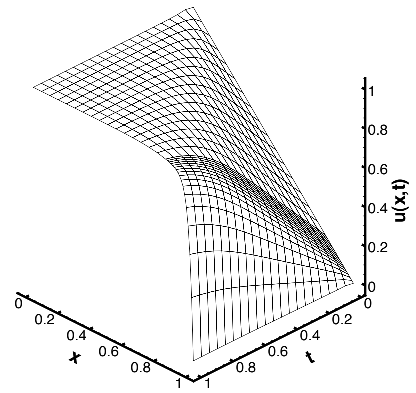
The problem can be solved in two ways: time accurate with $\Delta t = 0.05$ which corresponds to $C_{\Delta t} \approx \mathcal{O}(1)$ or directly steady-state with $\Delta t = 5$ which corresponds to $C_{\Delta t} \approx \mathcal{O}(100)$. In Figure 6, the space-time solution and the convergence in pseudo-time for a few physical time steps are shown. With eight orders of convergence in fifty cycles, an effective damping factor of 0.7 is achieved. In Figure 7, the steady-state solution is shown. With a single time step the convergence in pseudo-time is ten orders in one hundred and fifty cycles which corresponds to a damping factor of 0.85. Despite the presence of boundary conditions and the inaccurate solution of the coarse grid problem, these convergence rates are in agreement with the rate obtained from the analysis. The latter being 0.74 when diffusion is dominating (Table 2), which is the case in the boundary layer.

These results show that the EXI and EXV methods can indeed be combined to form a cheap local smoother for a full multigrid setting as expected from the analysis.

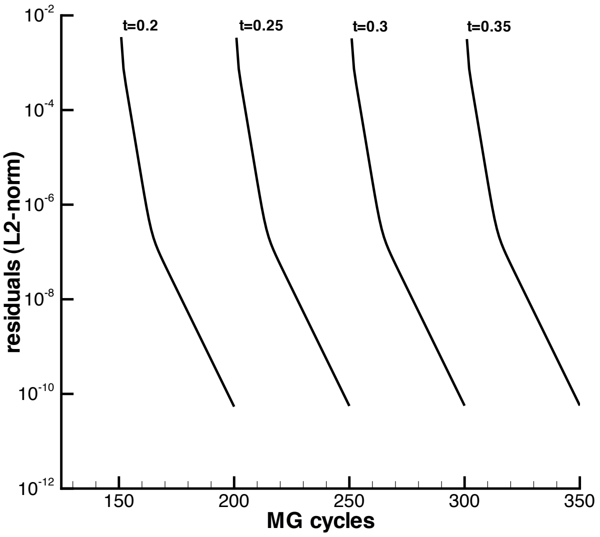
5 Conclusion

In this paper, we have studied the convergence of Runge-Kutta multigrid iteration for solving the system of algebraic equations resulting from the space-time DG discretization of the scalar advection-diffusion equation. In this local mode analysis we avoid the cell-staggering problem of transferring cell data from coarse to fine cells, by associating the data in a cell with a nodal point.

The analysis shows that explicit Runge-Kutta methods can be applied as smoothers in a multigrid setting. Depending on the case under consideration, two-level convergence factors between 0.48 and 0.74 are obtained.

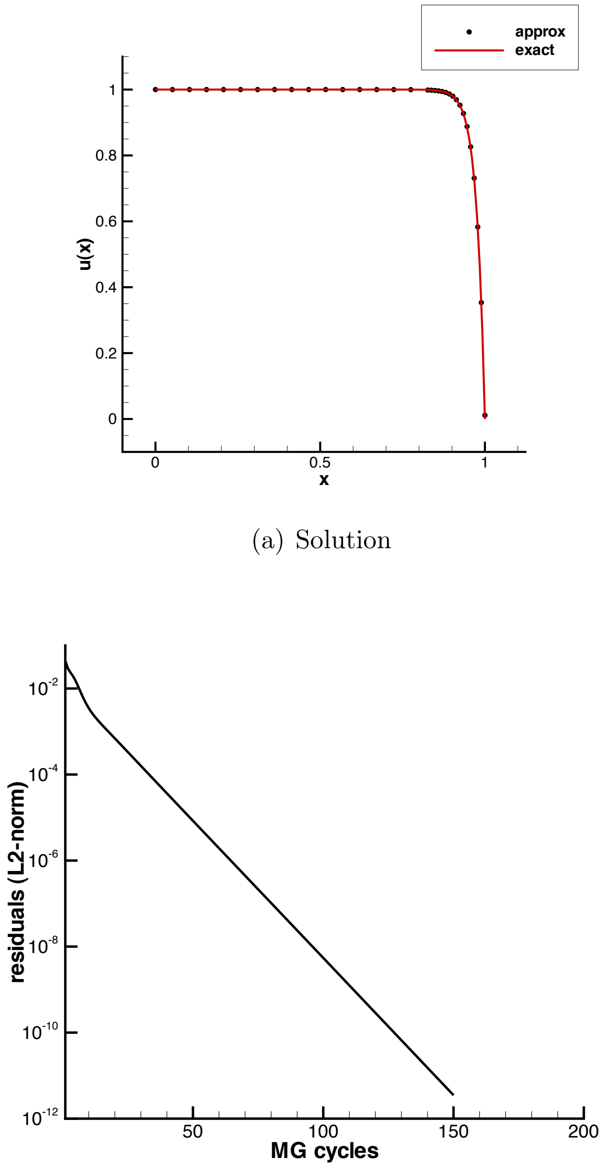


(a) Solution



(b) Convergence

Figure 6: The space-time solution of the advection-diffusion equation ($a = 1$, $d = 0.025$) on a Shishkin mesh with 32 elements and the convergence in pseudo-time of the MG algorithm for a few physical time steps $\Delta t = 0.05$.



(a) Solution

(b) Convergence

Figure 7: The steady-state solution of the advection-diffusion equation ($a = 1$, $d = 0.025$) on a Shishkin mesh with 32 elements and the convergence in pseudo-time of the MG algorithm for a single physical time step $\Delta t = 5$.

With a numerical illustration we show the multigrid convergence behaviour in practice and we find multigrid convergence factors that are in agreement with the convergence factors obtained from the analysis.

The advantage of using Runge-Kutta smoothing is that it results in a fully explicit approach, which can be extended to complex multidimensional problems where implicit smoothing may be too costly.

Acknowledgements

The research of Marc van Raalte has been supported by the Computational Life Science programme of the Netherlands Organisation of Scientific Research (NWO) (C-pump project).

The research of Christiaan M. Klaij has been conducted in the STW project TWI.5541, entitled *Advanced simulation techniques for vortex dominated flows in aerodynamics.* The financial support from STW and the National Aerospace Laboratory NLR is gratefully acknowledged.

We thank Ben Sommeijer, Jaap van der Vegt, Harmen van der Ven and the members of the C-pump project for their good suggestions and discussions.

References

[1] D. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39:1749–1779, 2002.

[2] F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous Galerkin methods for elliptic problems. *Numer. Methods Partial Differential Eq.*, 16:365–278, 2000.

[3] W.L. Briggs, Van Emden Henson, and S.F. McCormick. *A Multigrid tutorial.* SIAM, 2000.

[4] J. Gopalakrishan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numerische Mathematic*, 95:527–550, 2003.

[5] P.W. Hemker, W. Hoffmann, and M.H. van Raalte. Two-level Fourier Analysis of a multigrid Approach for Discontinuous Galerkin Discretisation. *SIAM Journal on Scientific Computing*, 25:1018–1041, 2004.

[6] P.W. Hemker and M.H. van Raalte. Fourier two-level analysis for higher dimensional discontinuous Galerkin discretisation. *Computing and Visualization in Science*, 7:159–172, 2004.

[7] K. Johannsen. Multigrid Methods for NIPG. Technical Report ICES 05-32, University of Texas, 2005.

[8] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Pseudo-time stepping methods for space-time discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *J. Comput. Phys. (in press)*, 2006.

[9] C.M. Klaij, J.J.W. van der Vegt, and H. van der Ven. Space-time discontinuous Galerkin method for the compressible Navier-Stokes equations. *J. Comput. Phys. (in press)*, 2006.

[10] W.L. Kleb, W.A. Wood, and B. van Leer. Efficient Multi-Stage Time Marching for Viscous Flows via Local Preconditioning. *AIAA J.*, 99-3267:181–194, 1999.

[11] N.D. Melson, M.D. Sanetrik, and H.L. Atkins. Time-accurate Navier-Stokes calculations with multigrid acceleration. In *Proc. 6th Copper Mountain Confer. on Multigrid Methods*, 1993.

[12] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR 73-479, LANL, 1973.

[13] M. Stynes and E. O'Riordan. A Uniformly Convergent Galerkin Method on a Shishkin Mesh for Convection-Diffusion Problem. *J. Math. Anal. Appl.*, 214:36–54, 1997.

[14] J.J. Sudirham, J.J.W. van der Vegt, and R.M.J. van Damme. Space-time discontinuous Galerkin method for advection-diffusion problems. Application to wet-chemical etching processes. *Appl. Numer. Mathematics (in press)*, 2006.

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

[16] 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*, 182:546–585, 2002.

[17] M. H. van Raalte. *Multigrid Analysis and Embedded Boundary Conditions for Discontinuous Galerkin Discretization.* PhD thesis, Korteweg-de Vries institute, University of Amsterdam, 2004.

[18] P. Wesseling. A robust and efficient multigrid method. In W. Hackbush and U. Trottenberg, editors, *Multigrid Methods*, pages 614–630. Springer-Verlag, New York, 1982.