Multigrid Reduced in Time for Isogeometric Analysis

R. Tielen*, M. Möller* and C. Vuik*

* Delft Institute of Applied Mathematics (DIAM)
Delft University of Technology
Delft, the Netherlands
e-mail: r.p.w.m.tielen@tudelft.nl, m.moller@tudelft.nl, c.vuik@tudelft.nl

Key words: Multigrid Reduced in Time, Isogeometric Analysis, Multigrid

Abstract: *Isogeometric Analysis (IgA) can be seen as the natural extension of the Finite Element Method (FEM) to high-order B-spline basis functions. Combined with a time integration scheme within the method of lines, IgA has become a viable alternative to FEM for time-dependent problems. However, as processors' clock speeds are no longer increasing but the number of cores are going up, traditional (i.e., sequential) time integration schemes become more and more the bottleneck within these large-scale computations. The Multigrid Reduced in Time (MGRIT) method is a parallel-in-time integration method that enables exploitation of parallelism not only in space but also in the temporal direction. In this paper, we apply MGRIT to discretizations arising from IgA for the first time in the literature. In particular, we investigate the (parallel) performance of MGRIT in this context for a variety of geometries, MGRIT hierarchies and time integration schemes. Numerical results show that the MGRIT method converges independent of the mesh width, spline degree of the B-spline basis functions and time step size Δt and is highly parallelizable when applied in the context of IgA.*

1 INTRODUCTION

Isogeometric Analysis (IgA) [1] can be seen as the natural extension of the Finite Element Method (FEM) to high-order B-spline basis functions. By using the same building blocks (i.e., B-splines and Non-Uniform Rational B-Splines) as in Computer Aided Design (CAD), IgA tries to bridge the gap between CAD and FEM, resulting in a highly accurate represention of (curved) geometries. Furthermore, the use of high-order B-spline basis functions has shown to be advantageous in many applications [3, 4, 5] and the accuracy per degree of freedom (DOF) compared to FEM is significantly higher [6].

For time-dependent partial differential equations (PDEs), Isogeometric Analysis is often combined with a time integration scheme within the method of lines. However, as with all traditional time integration schemes, the latter part is sequential by design and hence, a bottleneck in numerical simulations. When the spatial resolution is increased to improve accuracy, the time step size has to be reduced accordingly to ensure stability of the overall method. At the same time, processors' clock speeds are no longer increasing, but the core count goes up, which calls for the parallelization of the calculation process to benefit from modern computer hardware. As traditional time integration schemes are sequential by nature, new parallel-in-time methods are needed to resolve this problem.

The Multigrid Reduced in Time (MGRIT) method [2] is a parallel-in-time algorithm based on multigrid reduction (MGR) techniques [7]. In contrast to space-time methods, in which time is considered as an extra spatial dimension, sequential time stepping is still necessary within MGRIT. Space-time methods have been combined in the literature with IgA [8]. Although very successful, a drawback of such methods is the fact that they are more intrusive on existing codes, while MGRIT just requires a routine to integrate the fully discrete problem between two time instances. Over the years, MGRIT has been studied in detail and applied to a variety of problems in the literature [9, 10].

https://doi.org/10.4995/YIC2021.2021.12219

To the best of our knowledge, this is the first publication that reports on combining Isogeometric Analysis and MGRIT and therefore our focus lies on the performance of MGRIT when different multigrid hierarchies, geometries and time integration schemes are considered within an IgA setting.

This paper is structured as follows: Section 2 presents our two-dimensional model problem and its spatial and temporal discretization. The MGRIT algorithm is then described in Section 3. Numerical results, including CPU timings, obtained for different geometries and time integration schemes are presented for different configurations of the MGRIT method in Section 4. Finally, conclusions are drawn in Section 5.

2 MODEL PROBLEM AND DISCRETIZATION

As a model problem, we consider the transient diffusion equation:

$$\partial_t u(\mathbf{x},t) - \kappa \Delta u(\mathbf{x},t) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega, t \in [0,T]. \tag{1}$$

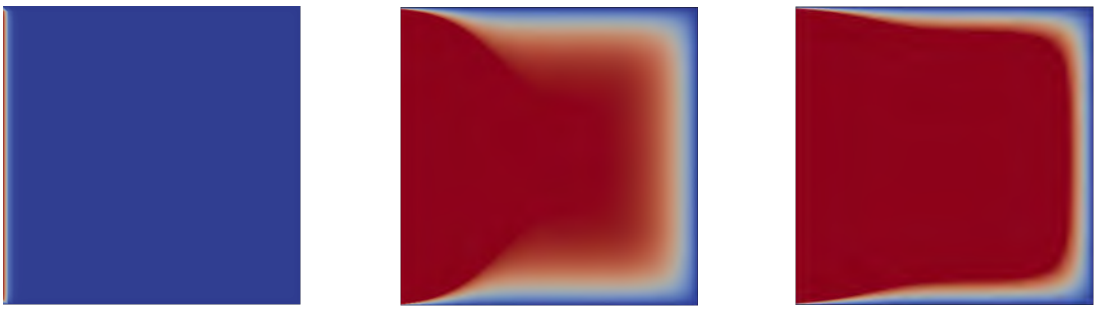
Here, κ denotes a constant diffusion coefficient, Ω the unit square (i.e., $[0,1]^2$) and $f \in L^2(\Omega)$ a source term. The above equation is complemented by initial conditions and both Dirichlet and Neumann boundary conditions:

$$u(\mathbf{x},0) = u^0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2}$$

$$u(\mathbf{x},t) = 0, \quad \mathbf{x} \in \partial\Omega \setminus \partial\Omega_W, t \in [0,T], \tag{3}$$

$$\frac{\partial u(\mathbf{x},t)}{\partial n} = 1, \quad \mathbf{x} \in \partial\Omega_W, t \in [0,T], \tag{4}$$

where Ω_W denotes the left boundary of Ω. Figure 1 denotes the solution of Equation (1) subject to these initial and boundary conditions at various time instances.



(a) $t = 0$ (b) $t = 0.01$ (c) $t = 0.02$

Figure 1: Solution to Equation (1) on the unit square at different times t.

First, we discretize Equation (1) (in time) by dividing the time interval $[0,T]$ in N_t subintervals of size Δt and applying the θ-scheme to the temporal derivative, which leads to the following equation to be solved at every time step:

$$u(\mathbf{x})^{k+1} - \kappa \Delta t \theta \Delta u(\mathbf{x})^{k+1} = u(\mathbf{x})^k + \kappa \Delta t (1-\theta) \Delta u(\mathbf{x})^k + \Delta t f(\mathbf{x}), \quad \mathbf{x} \in \Omega, k = 0, \ldots, N_t. \tag{5}$$

To obtain the variational formulation, let $\mathcal{V} = H_0^1(\Omega)$ be the space of functions in the Sobolev space $H^1(\Omega)$ that vanish on the boundary $\partial\Omega$. Equation (5) is multiplied with a test function $v \in \mathcal{V}$ and the result is then integrated over the domain Ω:

$$\int_\Omega \left(u^{k+1} v - \kappa \Delta t \theta \Delta u^{k+1} v \right) \mathrm{d}\Omega = \int_\Omega \left(u^k v + \kappa \Delta t (1-\theta) \Delta u^k v + \Delta t f v \right) \mathrm{d}\Omega. \tag{6}$$

Applying integration by parts on the second term on both sides of the equation results in

$$\int_\Omega \left(u^{k+1}v + \kappa\Delta t\theta\nabla u^{k+1}\cdot\nabla v\right)\ \mathrm{d}\Omega = \int_\Omega \left(u^{k+1}v - \kappa\Delta t(1-\theta)\nabla u^{k}\cdot\nabla v + \Delta t f v\right)\ \mathrm{d}\Omega, \tag{7}$$

for $\mathbf{x}\in\Omega, k=0,\dots,N_t$, where the boundary integral integral vanishes since $v=0$ on $\partial\Omega$. To parameterize the physical domain Ω, a geometry function $\mathbf{F}$ is then defined, describing an invertible mapping to connect the parameter domain $\Omega_0=(0,1)^2$ with the physical domain Ω:

$$\mathbf{F}:\Omega_0\to\Omega,\quad \mathbf{F}(\boldsymbol{\xi})=\mathbf{x}. \tag{8}$$

Provided that the physical domain Ω is topologically equivalent to the unit square, the geometry can be described by a single geometry function $\mathbf{F}$. In case of more complex geometries, a family of functions $\mathbf{F}^{(m)}$ $(m=1,\dots,K)$ is defined and we refer to Ω as a multipatch geometry consisting of K patches. For a more detailed description of the spatial discretization in Isogeometric Analysis and multipatch constructions, the authors refer to chapter 2 of [1].

Then, we express u at every time step by a linear combination of multivariate B-spline basis functions. Multivariate B-spline basis functions are defined as the tensor product of univariate B-spline basis functions $\phi_{i,p}$ $(i=1,\dots,N)$, which are uniquely defined on the parameter domain $(0,1)$ by an underlying knot vector $\Xi=\{\xi_1,\xi_2,\dots,\xi_{N+p},\xi_{N+p+1}\}$. Here, N denotes the number of univariate B-spline basis functions and p the spline degree. Based on this knot vector, the basis functions are defined recursively by the Cox-de Boor formula [11], starting from the constant ones

$$\phi_{i,0}(\xi)=\begin{cases}1 & \text{if } \xi_i\le\xi<\xi_{i+1},\\ 0 & \text{otherwise.}\end{cases} \tag{9}$$

Higher-order B-spline basis functions of order $p>0$ are then defined recursively

$$\phi_{i,p}(\xi)=\frac{\xi-\xi_i}{\xi_{i+p}-\xi_i}\phi_{i,p-1}(\xi)+\frac{\xi_{i+p+1}-\xi}{\xi_{i+p+1}-\xi_{i+1}}\phi_{i+1,p-1}(\xi). \tag{10}$$

The resulting B-spline basis functions $\phi_{i,p}$ are non-zero on the interval $[\xi_i,\xi_{i+p+1})$ and possess the partition of unity property. Furthermore, the basis functions are C^{p-m_i}-continuous, where m_i denotes the multiplicity of knot ξ_i. Throughout this paper, we consider a uniform knot vector with knot span size h, where the first and last knot are repeated $p+1$ times. As a consequence, the resulting B-spline basis functions are C^{p-1} continuous and interpolatory at both end points. Figure 2 illustrates both linear and quadratic B-spline basis functions based on such a knot vector.

Denoting the total number of multivariate B-spline basis functions $\Phi_{i,p}$ by N_{dof}, the solution u is thus approximated at every time step as follows:

$$u(\mathbf{x})\approx u_{h,p}(\mathbf{x})=\sum_{i=1}^{N_{\text{dof}}} u_i\Phi_{i,p}(\mathbf{x}),\quad u_{h,p}\in\mathcal{V}_{h,p}. \tag{11}$$

Here, the spline space $\mathcal{V}_{h,p}$ is defined, using the inverse of the geometry mapping $\mathbf{F}^{-1}$ as pull-back operator, as follows:

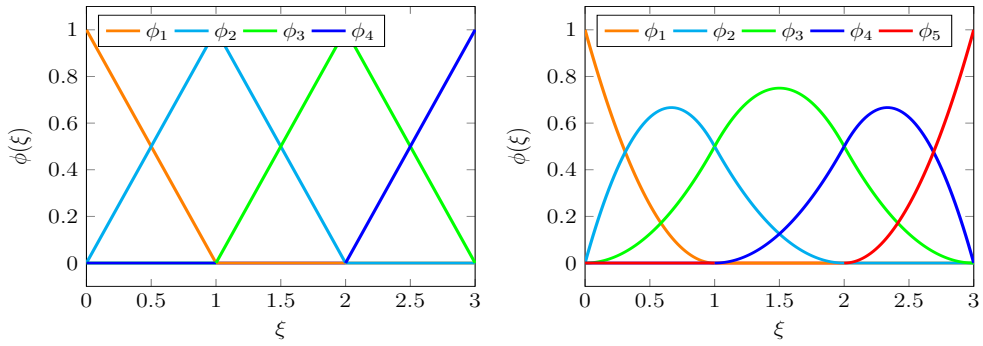
$$\mathcal{V}_{h,p}:=\operatorname{span}\left\{\Phi_{i,p}\circ\mathbf{F}^{-1}\middle|\ \Phi_{i,p}=0 \text{ on } \partial\Omega_0\right\}_{i=1,\dots,N_{\text{dof}}}. \tag{12}$$

By setting $v=\Phi_{j,p}$, Equation (7) can be written as follows:

$$(\mathbf{M}+\kappa\Delta t\theta\mathbf{K})\,\mathbf{u}^{k+1}=(\mathbf{M}-\kappa\Delta t(1-\theta)\mathbf{K})\,\mathbf{u}^{k}+\Delta t\mathbf{f},\ k=0,\dots,N_t, \tag{13}$$

where $\mathbf{M}$ and $\mathbf{K}$ denote the mass and stiffness matrix, respectively:

$$\mathbf{M}_{i,j}=\int_\Omega \Phi_{i,p}\Phi_{j,p}\ \mathrm{d}\Omega,\qquad \mathbf{K}_{i,j}=\int_\Omega \nabla\Phi_{i,p}\cdot\nabla\Phi_{j,p}\ \mathrm{d}\Omega,\quad i,j=1,\dots,N_{\text{dof}}. \tag{14}$$



(a) Linear B-spline basis functions (b) Quadratic B-spline basis functions

Figure 2: Linear and quadratic B-spline basis functions based on the knot vectors (a) $\Xi_1 = \{0, 0, 1, 2, 3, 3\}$ and (b) $\Xi_2 = \{0, 0, 0, 1, 2, 3, 3, 3\}$, respectively.

3 MULTIGRID REDUCED IN TIME

Instead of solving Equation (13) step-by-step directly, we apply the Multigrid Reduced in Time (MGRIT) method. For the ease of notation, we set $\theta = 1$ throughout the remainder of this section. Let $\Psi = (\mathbf{M} + \kappa\Delta t\mathbf{K})^{-1}$ denote the inverse of the left-hand side operator. Equation (13) can then be written as follows:

$$\mathbf{u}^{k+1} = \Psi\mathbf{M}\mathbf{u}^k + \mathbf{g}^{k+1},\ k = 0, \dots, N_t, \tag{15}$$

where $\mathbf{g}^{k+1} = \Psi\Delta t\mathbf{f}$. Setting $\mathbf{g}^0$ equal to the initial condition $u^0(\mathbf{x})$ projected on the spline space $\mathcal{V}_{h,p}$, the time integration method can be written as a linear system of equations:

$$\mathbf{A}\mathbf{u} = \begin{bmatrix} I & & & \\ -\Psi\mathbf{M} & I & & \\ & \ddots & \ddots & \\ & & -\Psi\mathbf{M} & I \end{bmatrix} \begin{bmatrix} \mathbf{u}^0 \\ \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^{N_t} \end{bmatrix} = \begin{bmatrix} \mathbf{g}^0 \\ \mathbf{g}^1 \\ \vdots \\ \mathbf{g}^{N_t} \end{bmatrix} = \mathbf{g}. \tag{16}$$

The two-level MGRIT method combines the use of a cheap coarse-level time integration method with an accurate more expensive fine-level one which can be performed in parallel. That is, Equation (16) can be solved iteratively by introducing a coarse temporal mesh with time step size $\Delta t_C = m\Delta t_F$. Here, Δt_F coincides with the Δt from the previous sections and m denotes the coarsening factor. It can be observed that the solution of Equation (16) at the coarse-level times $T_0, T_1, \dots, T_{N_t/m}$ satisfies:

$$\mathbf{A}_\Delta\mathbf{u}_\Delta = \begin{bmatrix} I & & & \\ -(\Psi\mathbf{M})^m & I & & \\ & \ddots & \ddots & \\ & & -(\Psi\mathbf{M})^m & I \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Delta^0 \\ \mathbf{u}_\Delta^1 \\ \vdots \\ \mathbf{u}_\Delta^{N_t/m} \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\Delta^0 \\ \mathbf{g}_\Delta^1 \\ \vdots \\ \mathbf{g}_\Delta^{N_t/m} \end{bmatrix} = \mathbf{g}_\Delta. \tag{17}$$

Here, $\mathbf{u}_\Delta^j = \mathbf{u}^{jm}$ and the vector $\mathbf{g}_\Delta$ is given by the original vector $\mathbf{g}$ multiplied by a restriction operator:

$$\mathbf{g}_\Delta = \begin{bmatrix} I & (\Psi\mathbf{M})^{m-1} & \dots & \Psi\mathbf{M} & I & & & & \\ & & & & & \ddots & & & \\ & & & & & & (\Psi\mathbf{M})^{m-1} & \dots & \Psi\mathbf{M} & I \end{bmatrix} \begin{bmatrix} \mathbf{g}^0 \\ \mathbf{g}^1 \\ \vdots \\ \mathbf{g}^{N_t} \end{bmatrix}. \tag{18}$$

A two-level MGRIT method solves the coarse system given by Equation (17) iteratively and computes the fine-level values in parallel within each interval (t_{jm}, t_{jm+m-1}). The coarse system is solved using the following residual correction scheme:

$$\mathbf{u}_{\Delta}^{(n+1)} = \mathbf{u}_{\Delta}^{(n)} + \mathbf{B}_{\Delta}^{-1}\left(\mathbf{g}_{\Delta} - \mathbf{A}_{\Delta}\mathbf{u}_{\Delta}^{(n)}\right), \tag{19}$$

where $\mathbf{B}_{\Delta}$ is the coarse-level equivalent of the matrix $\mathbf{A}$ based on Δt_C instead of Δt_F. More precisely, solving for $\mathbf{B}_{\Delta}$ gives the solution on the coarse mesh by coarse time stepping (using Δt_C), while solving for $\mathbf{A}_{\Delta}$ results in the solution on the coarse mesh by fine time stepping (using Δt_F). Here, the fine-level values are computed in parallel, denoted by the action of operator $\mathbf{A}_{\Delta}$. This is in contrast to the action of $\mathbf{B}_{\Delta}$ which typically is performed on a single processor.

The two-level MGRIT algorithm can be seen as a multigrid reduction (MGR) method that combines a coarse time stepping method with (parallel) fine time stepping within each coarse time interval. Here, the time stepping from a coarse point C to all neighbouring fine points is also referred to as F-relaxation [2]. On the other hand, time stepping to a C-point from the previous F-point is referred to as C-relaxation. It should be noted that both types of relaxation are highly parallel and can be combined leading to so-called CF- or FCF-relaxation.

3.1 Multilevel MGRIT method

Next, we consider the true multilevel MGRIT method. First, we define a hierarchy of L temporal meshes, where the time step size for the discretization at level l ($l = 0, 1, \ldots, L$) is given by $\Delta t_F m^l$. The total number of levels L is related to the coarsening factor m and the total number of fine steps Δt_F by $L = log_m(N_t)$. Let $\mathbf{A}^{(l)}\mathbf{u}^{(l)} = \mathbf{g}^{(l)}$ denote the linear system of equations based on the considered time step size at level l. The MGRIT method can then be written as follows:

Algorithm 1 MGRIT

if $l =$ L **then**
  Solve $\mathbf{A}^{(L)}\mathbf{u}^{(L)} = \mathbf{g}^{(L)}$
else
  Apply FCF-relaxation on $\mathbf{A}^{(l)}\mathbf{u}^{(l)} = \mathbf{g}^{(l)}$
  Restrict the residual $\mathbf{g}^{(l)} - \mathbf{A}^{(l)}\mathbf{u}^{(l)}$ using injection
  Call **MGRIT** setting $l \rightarrow l + 1$
  Update $\mathbf{u}^{(l)} \rightarrow \mathbf{u}^{(l)} + P\mathbf{u}^{(l+1)}$
end if

Here, the prolongation operator P is based on ordering the F-points and C-points, starting with the F-points. The matrix $\mathbf{A}$ can then be written as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{ff} & \mathbf{A}_{fc} \\ \mathbf{A}_{cf} & \mathbf{A}_{cc} \end{bmatrix}. \tag{20}$$

and the operator P is then defined as the "ideal interpolation" [2]:

$$P = \begin{bmatrix} -\mathbf{A}_{ff}\mathbf{A}_{fc} \\ \mathbf{I}_c \end{bmatrix}. \tag{21}$$

The recursive algorithm described above leads to a so-called V-cycle. However, as with standard multigrid methods, alternative cycle types (i.e., W-cycles, F-cycles) can be defined. At all levels of the multigrid hierarchy, the operators are obtained by rediscretizing Equation (1) using a different time step size.

4 NUMERICAL RESULTS

To assess the effectiveness of the MGRIT method when applied in combination with Isogeometric Analysis, we solve Equation (1) on the time domain $T = [0, 0.1]$, where the initial condition is chosen equal to zero and a right-hand side equal to one. This initial condition is adopted as well as initial guess at all times $t > 0$. The MGRIT method is said to have reached convergence if the relative residual at the end of an iteration is smaller or equal to 10^{-10}, unless stated otherwise.

Throughout this section, the MGRIT hierarchy, the domain of interest Ω and the time integration scheme are varied. The MGRIT hierarchies that will be adopted are two-level methods, a V-cycle and an F-cycle. As a domain, we consider the unit square (i.e., $\Omega = [0, 1]^2$), a quarter annulus defined in the first quadrant with inner radius of 1 and an outer radius of 2 and a multipatch geometry, see Figure 3. As a time integration scheme, we consider a value of θ of 0, 0.5 and 1 for the θ-scheme throughout this section, which corresponds to forward Euler, Crank-Nicolson and backward Euler, respectively.

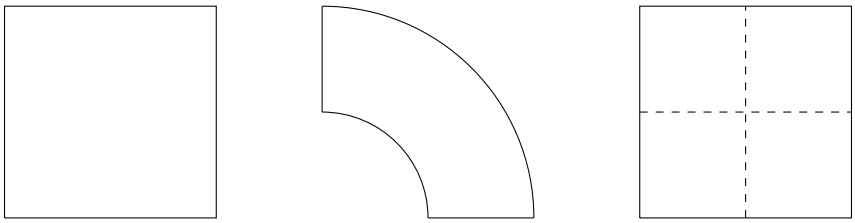


Figure 3: Spatial domains Ω considered throughout this section.

4.1 MGRIT hierarchies

First, we consider the MGRIT method using different hierarchies for the implicit case (i.e., backward Euler). At each time step, the linear system (Equation (13)) is solved by the Conjugate Gradient method. Table 1 shows the number of MGRIT iterations for different values of h and p when a two-level method, V-cycles or F-cycles are considered. Here, F-relaxation is applied at all levels of the MGRIT hierarchy. The number of time steps N_t for all configurations equals 100. For all three hierarchies, the number of MGRIT iterations needed to reach convergence is independent of h and p. The results obtained with a two-level method or F-cycles are identical and lead to a lower number of iterations compared to the use of V-cycles for all configurations.

	$p=2$			$p=3$			$p=4$			$p=5$		
	TL	V	F	TL	V	F	TL	V	F	TL	V	F
$h=2^{-4}$	7	9	7	7	9	7	7	9	7	7	9	7
$h=2^{-5}$	7	9	7	7	9	7	7	9	7	7	9	8
$h=2^{-6}$	8	9	8	8	9	8	8	9	8	8	9	8
$h=2^{-7}$	8	9	8	8	9	8	8	9	8	8	9	8

Table 1: Number of MGRIT iterations for solving the model problem when adopting a two-level (TL) method, V-cycles (V) or F-cycles (F).

Instead of varying the mesh width, the number of time steps can be increased as well. This is particularly interesting as MGRIT is a parallel-in-time method, where speed-ups will primarily come from parallelization in the temporal component. Table 2 shows the number of MGRIT iterations adopting different hierarchies for different numbers of time steps, different values of p

and $h = 2^{-6}$. For all configurations, the use of a two-level method or F-cycles leads to a lower number of iterations compared to the use of V-cycles. In particular, the number of iterations are independent of the number of time steps for all MGRIT hierarchies and comparable to the ones obtained when considering different values of the mesh width.

	$p=2$			$p=3$			$p=4$			$p=5$		
	TL	V	F	TL	V	F	TL	V	F	TL	V	F
$N_t = 250$	7	10	7	7	10	7	7	10	7	7	10	7
$N_t = 500$	7	10	7	7	10	7	7	10	7	7	10	7
$N_t = 1000$	7	11	7	7	11	7	7	11	7	7	11	7
$N_t = 2000$	7	11	7	7	11	7	7	11	7	7	11	7

Table 2: Number of MGRIT iterations for solving the model problem when adopting a two-level (TL) method, V-cycles (V) or F-cycles (F).

4.2 Varying geometries

Next, we apply MGRIT on a curved and multipatch geometry, respectively. Table 3 shows the number of V-cycles needed with MGRIT, using backward Euler for the time integration, for both geometries. Results can be compared to the ones presented in Table 2, showing identical iteration numbers for all geometries.

	Quarter Annulus				Multipatch			
	$p=2$	$p=3$	$p=4$	$p=5$	$p=2$	$p=3$	$p=4$	$p=5$
$N_t = 250$	10	10	10	10	10	10	10	10
$N_t = 500$	10	10	10	10	10	10	10	10
$N_t = 1000$	11	11	11	11	11	11	11	11
$N_t = 2000$	11	11	11	11	11	11	11	11

Table 3: Number of MGRIT iterations for solving Equation (1) on a quarter annulus and multipatch geometry when adopting V-cycles for varying time step sizes.

Table 4 shows the results when the number of time steps is kept constant ($N_t = 100$) for the quarter annulus and multipatch geometry when adopting V-cycles. Results can be compared to Table 1 and are (again) identical for all three geometries.

	Quarter Annulus				Multipatch			
	$p=2$	$p=3$	$p=4$	$p=5$	$p=2$	$p=3$	$p=4$	$p=5$
$h = 2^{-4}$	9	9	9	9	9	9	9	9
$h = 2^{-5}$	9	9	9	9	9	9	9	9
$h = 2^{-6}$	9	9	9	9	9	9	9	9
$h = 2^{-7}$	9	9	9	9	9	9	9	9

Table 4: Number of MGRIT iterations for solving Equation (1) on a quarter annulus and multipatch geometry when adopting V-cycles for varying mesh widths.

4.3 Time integration schemes

Next to the implicit backward Euler scheme, we have considered alternative time integration schemes as well. In this subsection, we consider the forward Euler and (second-order accurate)

Crank-Nicolson method. The use of explicit time integration schemes in the context of parallel-in-time integration is on the one hand highly relevant, as the required number of time steps needed to ensure stability is relatively high. On the other hand, coarsening with respect to the time step size might still exhibit stability issues at coarser levels. Therefore, explicit-implicit methods are often considered, where explicit time integration is applied on the fine level problem, while implicit methods are adopted at the coarser levels. The question remains to which extent the resulting MGRIT algorithm remains robust in the mesh width and/or spline degree.

Table 5 shows the number of MGRIT iterations for different numbers of time steps when adopting V-cycles and a mesh width of $h = 2^{-4}$. Here, forward Euler/Crank-Nicolson is applied at the fine level, while backward Euler is applied at the coarse levels. For some of the considered configurations, the resulting MGRIT method does not converge for forward Euler (indicated by '$*$'). It should be noted, however, that for these configurations, forward Euler applied as a sequential time integration scheme does not converge either, which is a direct consequence of the CFL condition. When the Crank-Nicolson method is applied the resulting MGRIT method converges in a relatively low number of iterations.

	forward Euler				Crank-Nicolson			
	$p=2$	$p=3$	$p=4$	$p=5$	$p=2$	$p=3$	$p=4$	$p=5$
$N_t = 250$	$*$	$*$	$*$	$*$	11	11	14	24
$N_t = 500$	13	$*$	$*$	$*$	11	11	11	12
$N_t = 1000$	13	13	$*$	$*$	11	11	11	11
$N_t = 2000$	13	13	13	$*$	11	11	11	11

Table 5: Number of MGRIT iterations for solving Equation (1) on the unit square using forward Euler and Crank-Nicolson when adopting V-cycles.

Table 6 shows the number of MGRIT iterations for a varying mesh width and 1000 time steps for both time integration methods. For many configurations, MGRIT using forward Euler does not convergence, while the Crank-Nicolson method converges for all configurations. A small dependency on h and p is, however, visible.

	forward Euler				Crank-Nicolson			
	$p=2$	$p=3$	$p=4$	$p=5$	$p=2$	$p=3$	$p=4$	$p=5$
$h = 2^{-3}$	13	13	13	14	11	11	11	12
$h = 2^{-4}$	13	13	$*$	$*$	11	11	11	11
$h = 2^{-5}$	$*$	$*$	$*$	$*$	11	11	13	23
$h = 2^{-6}$	$*$	$*$	$*$	$*$	13	28	52	88

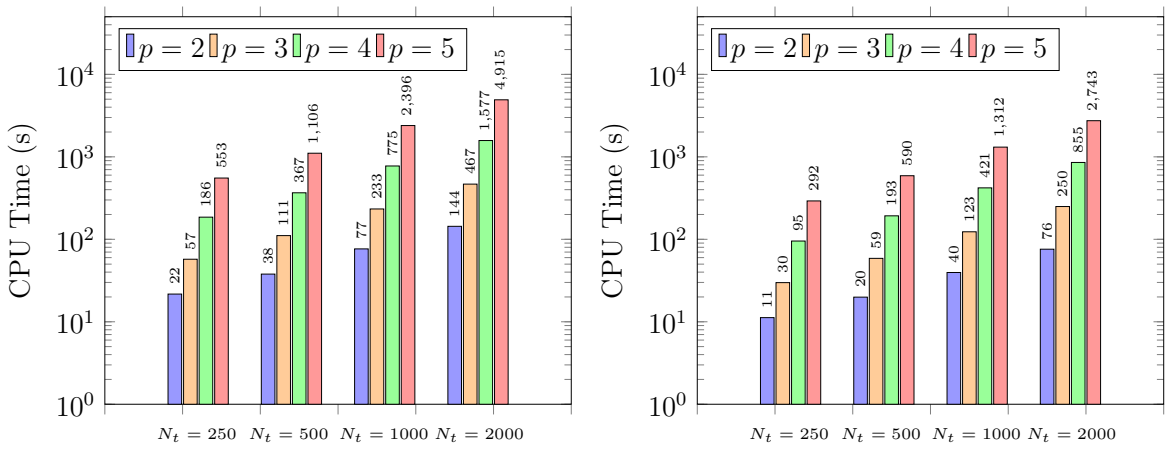
Table 6: Number of MGRIT iterations for solving Equation (1) on the unit square using forward Euler and Crank-Nicolson when adopting V-cycles.

4.4 CPU timings

Next to investigating the iteration numbers needed with MGRIT to reach convergence, CPU timings have been obtained as well. Here, we adopt V-cycles, a mesh width of $h = 2^{-6}$ and the unit square as our domain of interest. Note that the corresponding iteration numbers can be found in Table 2. The computations are performed on three nodes, which consist each of an Intel(R) i7-10700 (@ 2.90GHz) processor.

As shown in Figure 4a, doubling the number of time steps roughly doubles the time needed to reach convergence for all values of p. Furthermore, the CPU times significantly increase for higher values of p which is related to the spatial solves at every time step. It is known from the literature that standard iterative solvers have a detoriating performance for increasing values of p, leading to an increased number of CG iterations and, hence, higher computational costs.

In Figure 4b, results obtained adopting six cores can be found. In general, the same behavior can be observed with respect to the number of time steps and the spline degree. It should be noted, however, that doubling the number of cores significantly reduces the CPU time needed to reach convergence. More precisely, a reduction of $45 - 50\%$ can be observed when doubling the number of cores, implying the MGRIT algorithm is highly parallelizable.



(a) three cores

(b) six cores

Figure 4: CPU times for MGRIT using V-cycles and backward Euler on the unit square for a fixed problem size ($h = 2^{-6}$) adopting a different number of cores. The cores are evenly distributed over the nodes.

5 CONCLUSIONS

In this paper, we successfully combined Isogeometric Analysis with the Multigrid Reduced in Time (MGRIT) method to solve the time-dependent diffusion equation. Here, both (curved) single patch and multipatch geometries have been considered. Furthermore, different time integration methods and MGRIT hierarchies have been adopted. Numerical results show for all considered benchmarks that the MGRIT method converges independent of the considered mesh width h, spline degree p or time step size Δt. Furthermore, the use of an implicit time integration method has shown to be more robust compared to explicit time integration methods when applied within MGRIT. In general, a two-level hierarchy as well as the use of F-cycles leads to a slightly lower number of MGRIT cycles, but they are associated to higher costs per iteration. CPU timings show that the time needed to reach convergence does not only dependent on the number of time steps, but also on the spline degree of the B-spline basis functions when a standard iterative method is considered for the spatial solves. Future work will therefore focus on the use of state-of-the-art solvers for Isogeometric Analysis within MGRIT to mitigate this dependency. As increasing the number of CPUs significantly decreases the computational times, future research will focuss as well on the parallel performance of the MGRIT method and its comparison to traditional sequential time integration methods for large-scale simulations.

REFERENCES

[1] Hughes, T.J.R. and Cottrell, J.A. and Bazilevs, Y. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement *Computer Methods in Applied Mechanics and Engineering*, Vol. **194**, pp. 4135–4195, (2005).

[2] Falgout, R.D. and Friedhoff, S. and Kolev, Tz.V. and MacLachlan, S.P. and Schröder. J.B. Parallel Time Integration with Multigrid *SIAM Journal on Scientific Computing*, Vol. **36**, pp. C635–C661, (2014).

[3] Cottrell, J.A. and Reali, A. and Bazilevs, Y. and Hughes, T.J.R. Isogeometric Analysis of structural vibrations *Computer Methods in Applied Mechanics and Engineering*, Vol. **195**(41-43), pp. 5257–5296, (2006).

[4] Bazilevs, Y. and Calo, V.M. and Zhang, Y. and Hughes, T.J.R. Isogeometric Fluid-structure Interaction Analysis with Applications to Arterial Blood Flow *Computational Mechanics*, Vol. **38**, pp. 310–322, (2006).

[5] Wall, W.A. and Frenzel, M.A. and Cyron, C. Isogeometric structural shape optimization *Computer Methods in Applied Mechanics and Engineering*, Vol. **197**, pp. 2976–2988, (2008).

[6] Hughes, T.J.R. and Reali, A. and Sangalli, G. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS *Computer Methods in Applied Mechanics and Engineering*, Vol. **197**, pp. 4104–4124, (2008).

[7] Ries, M. and Trottenberg, U. and Winter, G. A note on MGR methods *Linear Algebra and its Applications*, Vol. **49**, pp. 1–26, (1983).

[8] Langer, U. and Moore, S.E. and Neumüller, M. Space-time isogeometric analysis of parabolic evolution problems *Computer Methods in Applied Mechanics and Engineering*, Vol. **306**, pp. 342–363, (2016).

[9] Günther, S. and Gauger, N.R. and Schröder, J.B. A non-intrusive parallel-in-time adjoint solver with the XBraid library *Computing and Visualization in Science*, Vol. **19**, pp. 85–95, (2018).

[10] Lecouvez, M. and Falgout, R.D. and Woodward, C.S. and Top, P. A parallel multigrid reduction in time method for power systems *C2016 IEEE Power and Energy Society General Meeting (PESGM)*, pp. 1–5, (2016).

[11] De Boor, C. *A practical guide to splines*. Springer, 2006.