Journal of Scientific Computing (2022) 91:28
https://doi.org/10.1007/s10915-022-01802-1

Implicitly Extrapolated Geometric Multigrid on Disk-Like Domains for the Gyrokinetic Poisson Equation from Fusion Plasma Applications

Martin J. Kühn[1] · **Carola Kruse**[2] · **Ulrich Rüde**[3]

Received: 25 May 2021 / Revised: 8 October 2021 / Accepted: 4 February 2022

Abstract
The gyrokinetic Poisson equation arises as a subproblem of Tokamak fusion reactor simulations. It is often posed on disk-like cross sections of the Tokamak that are represented in generalized polar coordinates. On the resulting curvilinear anisotropic meshes, we discretize the differential equation by finite differences or low order finite elements. Using an implicit extrapolation technique similar to multigrid τ-extrapolation, the approximation order can be increased. This technique can be naturally integrated in a matrix-free geometric multigrid algorithm. Special smoothers are developed to deal with the mesh anisotropy arising from the curvilinear coordinate system and mesh grading.

Keywords Fusion plasma · Disk-like · Cross section · Multigrid · Extrapolation

Mathematics Subject Classification 65N55 · 65N06 · 65N30 · 65B99

1 Introduction

In the context of Tokamak fusion plasma, a Poisson equation has to be solved on disk-like domains which correspond to the poloidal cross sections of the Tokamak geometry; see, e.g., [5, 10, 37]. In its most simplified form, this cross section takes a circular form but deformed (cf. Fig. 1, center) or more realistic D-shaped geometries were found to be advantageous; see, e.g., [5, 10]. For more details on the problem setting and the physical details, we refer the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No 824158.

✉ Martin J. Kühn
Martin.Kuehn@DLR.de

[1] Department for High-Performance Computing, Institute for Software Technology, German Aerospace Center (DLR), Cologne, Germany

[2] Parallel Algorithms Team, CERFACS (Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique), 42 Avenue Gaspard Coriolis, 31057 Toulouse Cedex 01, France

[3] Friedrich-Alexander Universität Erlangen-Nürnberg, Cauerstr. 11, 91058 Erlangen, Germany

reader to [5, 10, 13, 32, 37–39]. Here, we propose a tailored solver for

$$\begin{aligned} -\nabla \cdot (\alpha \nabla u) &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

where $\Omega \subset \mathbb{R}^2$ is a disk-like domain, $f : \Omega \to \mathbb{R}$, and $\alpha : \Omega \to \mathbb{R}$ is a varying coefficient, also refered to as *density profile*.

The purpose of this article is to develop a problem-specific solver for this scenario. Our particular problem setting is taken from [5, 13, 32, 38, 39]. The solution of this system is a part of the iterative solution process in large gyrokinetic codes such as Gysela [13] where the 2D problem must be solved repeatedly over many times steps on hundreds or thousands of such 2D cross sections. Note that the solution of the Gyrokinetic equation on each cross section is only a moderately sized problem, but each simulation run with the plasma physics code may require the solution of this subproblem millions of times. Already small improvements in the methods can lead to substantial reductions in computation time of the field solver. This motivates and justifies the effort for a dedicated development and for optimizing the algorithms.

The article will study several mathematical difficulties and propose special techniques to handle each of them. In particular,

- the domain geometry and the variable coefficient formally lead to a linear system with a sparse matrix. However, the conventional assembly and the repeated access to this matrix will limit performance. In fact, on many modern architectures, memory access can be more critical for performance than the floating point operations. Therefore, as an alternative to working with sparse matrix formats, we will here develop techniques that are suitable for a *matrix-free implementation*.
- The smoothness of the domain and the regularity of the coefficients justify the use of *higher order* discretizations, promising to reach better accuracy with fewer unknowns. Conventionally, however, higher order discretizations would lead to sparse matrices with more complex structure, which are also more densely populated. Solving for these discretizations would thus incur an increased cost. Here we will focus on a nonstandard alternative, where higher order is achieved using computationally cheap low order discretizations only, but using an *implicit extrapolation* step [18] to reach higher order.
- The solver algorithm itself must be scalable or, in other words, asymptotically optimal. Recall that linear (or close to linear) algorithmic complexity is a necessary condition for reaching scalability. Beyond this, we are here not only interested in the order of complexity, but also to keep the *absolute* cost minimal. Our focus is on *geometric multigrid* methods which belong to the most efficient solvers for elliptic problems, also in terms of absolute cost. In the development below, we will carefully optimize them for the given mesh structures. This requires in particular dealing with the anisotropy of the meshes and thus introducing special *zebra line smoothers*.
- Finally, all algorithms must be suitable for parallelization. Although developing an optimized parallel implementation is beyond the scope of the current article, we will design the algorithms such that parallelization will be possible in future work. This means that all algorithmic components, e.g., smoothers or grid transfer operators, are constructed such as to avoid any sequential bottlenecks or global dependencies, except the limitations caused by the multigrid hierarchy itself. In the envisioned application, many cross sections will be computed independently, leading to a significant degree of coarse grain parallelism. Beyond this the solution on each of the 2D cross sections must be suitable to exploit node level parallelism and instruction level parallelism. Therefore all algorithms

developed here are ready to exploit *intra-node parallelism*. Additional care has also been taken so that a *vectorization* of the loop kernels will be possible and also executing the solver on *accelerators such as a GPU* becomes possible.

In the article, we will consider two illustrative domain shapes. The first one is a simple circle or circular annulus and the second one is a deformed circle as introduced in [5]; see Fig. 1. These domains can be described in curvilinear coordinates. In its simplest form, the geometry can be described by polar coordinates, but for a more realistic geometry, more general transformations must be used.

One drawback of the curvilinear coordinates is the introduction of an artificial singularity in the origin of the mapping. Further challenges for our solver come from the anisotropy in the transformed meshes and a varying coefficient α describing a physical density. Additionally, the meshes may be refined anisotropically to account for particular physical effects.

Multigrid methods can achieve optimal complexity for many problems and are among the most efficient solvers for elliptic model problems such as (1.1); see, e.g., [6, 35]. However, multigrid methods for curvilinear (e.g., polar) meshes are less commonly studied topics; cf. [1, 3, 23, 33, 35] for some results. When additional difficulties arise, such as generalized polar coordinates, varying coefficients, and locally refined, anisotropic meshes, then the multigrid components must be suitably modified and adapted to maintain excellent convergence rates at low cost per iteration. Designing the algorithms for parallel execution on modern computer architectures and achieving a low memory footprint put additional constraints on the design of the multigrid components for coarsening, prolongation, and smoothing of the iterates.

We present a geometric multigrid algorithm using special line smoothers tailored to support parallel scalability. Additionally, we propose an implicit extrapolation scheme based on [15, 16, 18] with the goal to improve the order of *differential* convergence. Note that this refers to convergence of the algorithm with respect to the solution of the PDE, as different from *algebraic* convergence when considering the multigrid method as a linear system solver.

Conventional Richardson-style extrapolation for PDE relies on global asymptotic error expansions for the discrete solution and thus requires strict smoothness assumptions [22] to guarantee the existence of global asymptotic error expansions. An interesting variant of Richardson extrapolation that can save cost for higher dimensional problems is called *splitting* or *multi-parameter extrapolation*, see e.g. [20, 26, 30].

The combination of such extrapolation methods with iterative multilevel solvers, such as the multigrid method, is in many ways natural [29, 36] and has recently seen renewed interest [9, 19]. In particular, it is attractive to combine Richardson-style extrapolation with cascadic multigrid [4, 31] methods, as studied e.g. in [25, 34]. Since the cascadic multigrid method is not optimal in the L_2-norm, special extrapolation formulas must be developed [7, 8].

τ-extrapolation and other implicit extrapolation variants rely on extrapolation applied to local quantities, such as the residual or the local energy, and thus they can also be used when only local smoothness is guaranteed. In this article, we will use implicit variants of extrapolation as proposed in [15, 27, 28]. These methods are related to τ-extrapolation that has been proposed in combination with multigrid solvers [6, 14]. Different from τ-extrapolation, our method is based on the existence of an underlying energy functional and is thus limited in its current form to self-adjoint PDE.

The remainder of the article is organized as follows. In Sect. 2, we present the detailed problem setting and the geometry motivated from fusion plasma applications. In Sect. 3, we briefly introduce our five and nine point finite difference stencils as well as the finite elements combined with nonstandard numerical integration. We also briefly discuss the handling of the mesh singularity at the origin. In the main section, Sect. 4, we introduce the new geo-

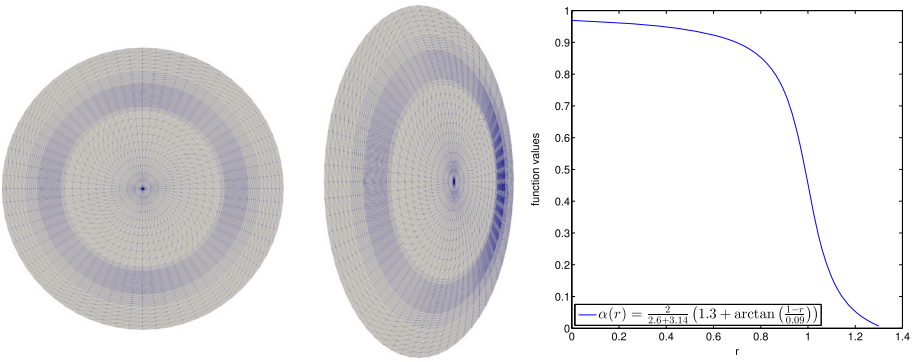

Fig. 1 Circular (left) and deformed circular (center) geometry that can be described by curvilinear coordinates $(r, \theta) \in [r_1, 1.3] \times [0, 2\pi]$ with the mapping F_{P} (2.1) (left) and F_{GP} (2.2), $\kappa = 0.3$, $\delta = 0.2$, (center), respectively. Rapidly decaying density profile (2.3) (right). Around the decay of the coefficient, the meshes are locally refined in r; here, with $h_{\max}/h_{\min} = 8$

metric multigrid algorithm with optimized line smoothers and using implicit extrapolation. In Sect. 5, we present numerical results.

2 Curvilinear Coordinates and Model Problem Representations

In this paper, we will consider *physical* domains Ω_{r_1} that can be described by a mapping from a *logical* domain $(r_1, 1.3) \times [0, 2\pi)$ onto Ω_{r_1} for $r_1 \in [0, 1.3)$. Except for the singularity arising for $r_1 = 0$, the mapping is invertible. We will later present different strategies to handle the artificial singularity. Note that $r_{\max} = 1.3$ is purely problem-specific and it will be used throughout this paper only for simplicity of the presentation; other values are possible.

First, we will consider the circular geometry, which can be described by the polar coordinate transformation $F_{\mathrm{P}}(r, \theta) = (x, y)$ with

$$x = r\cos(\theta), \quad y = r\sin(\theta), \quad (r, \theta) \in [r_1, 1.3] \times [0, 2\pi]; \tag{2.1}$$

see Fig. 1 (left). A generalized transformation $F_{\mathrm{GP}}(r, \theta) = (x, y)$ is given by

$$x = (1-\kappa)r\cos(\theta) - \delta r^2, \quad y = (1+\kappa)r\sin(\theta), \quad (r, \theta) \in [r_1, 1.3] \times [0, 2\pi]. \tag{2.2}$$

The particular model setting is based on [5, 32, 39] to describe more realistic Tokamak cross-sections. According to [5, 39], we use $\kappa = 0.3$ and $\delta = 0.2$. The resulting domain is illustrated in Fig. 1 (center). Note that (2.2) reduces for $\kappa = \delta = 0$ to (2.1). Nevertheless, we will also give explicit formulas for (2.1) since we consider the polar coordinate transformation as a case of particular interest.

In our simulations, we either use $\alpha \equiv 1$ to consider the Poisson equation or use a typical density profile given by

$$\alpha(r, \theta) = \alpha(r) = \frac{2}{2.6 + 3.14}\left(1.3 + \arctan\left(\frac{1-r}{0.09}\right)\right); \tag{2.3}$$

see Fig. 1 (right). The density profile (2.3) is motivated by [12, 32, 38] and models the rapid decay from the core to the edge region of the separatrix in the Tokamak.

Concerning the mesh, we use local refinements to pass from the core to the edge region of the separatrix; see, e.g., [24] or Fig. 1 for a representative refinement by a ratio of 8 in direction of r and a minimal mesh size of 49×64.

For the polar coordinate transformation and the coefficient (2.3) the partial differential equation from (1.1) reads

$$-\frac{2}{(2.6+3.14)\left(0.09-\frac{(1-r)^2}{0.09}\right)}\frac{\partial u}{\partial r}+\alpha(r)\left(\frac{\partial^2 u}{\partial r^2}+\frac{1}{r}\frac{\partial u}{\partial r}+\frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}\right)=f, \tag{2.4}$$

where the term in the last parenthesis corresponds to the well-known Laplacian operator expressed in polar coordinates.

Remark 2.1 Note that we neither explicitly distinguish between the functions $u(r,\theta) = \widetilde{u}(x,y)$ nor the operators $\widetilde{\nabla} = \nabla_{x,y}$ and $\nabla = \nabla_{r,\theta}$ in (1.1) and (2.4), defined for the corresponding variables. We will do this to a certain extent by using $\widetilde{\cdot}$ for operators and functions expressed in Cartesian coordinates in the following. However, to not overload the notation we waive this notational overhead whenever this disctinction becomes clear from the context.

In the following, we will consider the energy functional to be minimized corresponding to (1.1). For a scalar coefficient, it writes

$$\begin{aligned}\widetilde{E}(\widetilde{u}) &= \int_{\Omega^*_{r_1}} \left(\frac{1}{2}\widetilde{\alpha}|\widetilde{\nabla}\widetilde{u}|^2 - \widetilde{f}\widetilde{u}\right)\mathrm{d}(x,y)\\ &= \int_{r_1}^{1.3}\int_0^{2\pi}\left(\frac{1}{2}\alpha(DF_*^{-T}\nabla u,\ DF_*^{-T}\nabla u) - fu\right)|\det DF_*|\mathrm{d}(r,\theta) = E(u),\end{aligned} \tag{2.5}$$

where $* \in \{\mathrm{P},\ \mathrm{GP}\}$, DF_* is the Jacobian matrix, and $\Omega^*_{r_1}$ is the physical domain for either (2.1) or (2.2). In the remaining part of the paper, we mostly use the index $\cdot_*$ to refer to both transformations likewise. For the inverse transformations, see [5]. Due to space limitations, we only provide

$$\det DF_{\mathrm{P}} = r, \quad DF_{\mathrm{P}}^{-1} = \frac{1}{\det DF_{\mathrm{P}}}\begin{pmatrix} r\cos(\theta) & r\sin(\theta)\\ -\sin(\theta) & \cos(\theta)\end{pmatrix}, \tag{2.6}$$

and

$$\begin{aligned}\det DF_{\mathrm{GP}} &= r(1+\kappa)(1-\kappa-2\delta r\cos(\theta)),\\ DF_{\mathrm{GP}}^{-1} &= \frac{1}{\det DF_{\mathrm{GP}}}\begin{pmatrix}(1+\kappa)r\cos(\theta) & (1-\kappa)r\sin(\theta)\\ -(1+\kappa)\sin(\theta) & (1-\kappa)\cos(\theta)-2\delta r\end{pmatrix}.\end{aligned} \tag{2.7}$$

In order to simplify the notation for (2.5), we define

$$\begin{pmatrix} a_*^{rr} & \frac{1}{2}a_*^{r\theta}\\ \frac{1}{2}a_*^{\theta r} & a_*^{\theta\theta}\end{pmatrix} := \frac{1}{2}\alpha(r)DF_*^{-1}(r,\theta)DF_*^{-T}(r,\theta)|\det DF_*(x,y)|. \tag{2.8}$$

Note that (2.8) is symmetric and thus $a_*^{r\theta} = \frac{1}{2}a_*^{r\theta} + \frac{1}{2}a_*^{\theta r}$. Additionally, note that $a_{\mathrm{P}}^{r\theta} = 0$, i.e., for the polar coordinate transformation the offdiagonal entries are zero as long as the diffusion term α remains scalar.

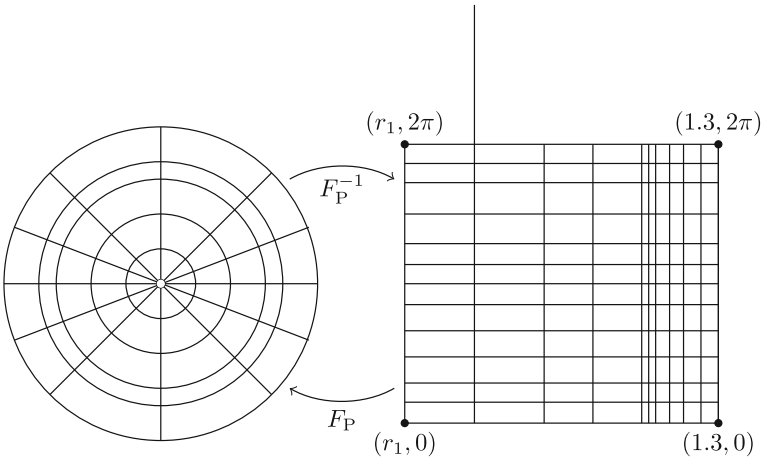

Fig. 2 Physical (left) and logical (right) domain for $r_1 > 0$ and polar coordinate transformation (2.1) with two-level hierarchical, anisotropic tensor-product mesh

3 Discretization Methods

In this paper, we will use linear finite elements and compact finite differences to construct a geometric multigrid algorithm on anisotropic grids represented by curvilinear coordinates. We use particular finite difference stencils from [18] which maintain the symmetry of the energy functional also for anisotropic grids. For finite elements, we introduce nonstandard integration rules that are advantageous when implicit extrapolation is used within the multigrid algorithm; cf. [16, 18].

We first consider an anisotropic hierarchical grid with two levels. This is suitable to apply the extrapolation method developed in [18]. The two-level hierarchical grid is given in tensor-product form on the logical domain $(r_1, 1.3) \times [0, 2\pi)$ (see Fig. 2) with

$$r_1 \geq 0,\ r_{n_r} := 1.3, \quad r_{2i} := r_{2i-1} + h_i,\ r_{2i+1} := r_{2i} + h_i,\ h_i > 0,\ 1 \leq i < \lfloor \frac{n_r}{2} \rfloor,$$

$$\theta_1 := 0,\ \theta_{n_\theta} := 2\pi, \quad \theta_{2j} := \theta_{2j-1} + k_j,\ \theta_{2j+1} := \theta_{2j} + k_j,\ k_j > 0,\ 1 \leq j < \lfloor \frac{n_\theta}{2} \rfloor,$$

where n_r and n_θ denote the (odd) number of nodes in r- and θ-direction, respectively. We denote $h := \max_i h_i$ and $k := \max_j k_j$.

3.1 Finite Element Discretization

We briefly recapitulate the nonstandard integration rule from [16, 18, 21]. In [18], it was already shown that the nonstandard quadrature rules may be better suited than standard quadrature for linear elements when approaching the singularity as $r_1 \to 0$.

Remark 3.1 (Nonstandard Finite Element integration) We use nodal $\mathcal{P}_1$ basis functions with a nonstandard numerical integration rule; see [16, 18, 21]. Let us note that this nonstandard integration rule does *not* use standard edge-midpoint quadrature. In this approach, **each additive term** of a reformulated integrand will be **evaluated at one term-specific evaluation node only**.

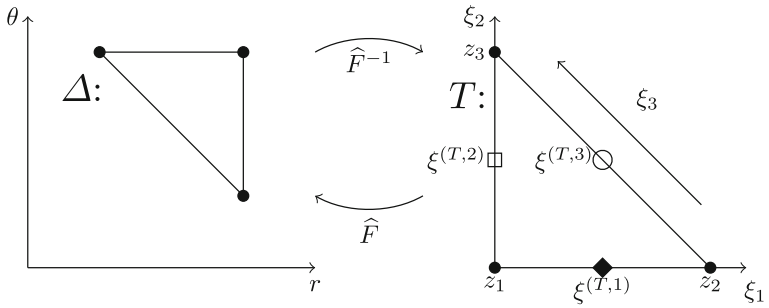

Fig. 3 Mesh element Δ (left) with transformation onto a reference triangle T (right). Definition of the directions $\xi_1 = e_1, \xi_2 = e_2$, and $\xi_3 = e_2 - e_1$ as well as the definition of the evaluation nodes $\xi^{(T,1)}, \xi^{(T,2)}$, and $\xi^{(T,3)}$ (right)

As in the case of standard integration, we map any triangle Δ of the triangulation of the logical domain onto the reference triangle $T = \{(\xi_1, \xi_2) \in \mathbb{R}^2 \ : \ 0 \le \xi_1, \xi_2 \le 1, \ \xi_1 + \xi_2 \le 1\}$. Then, we introduce the directional derivative

$$\frac{\partial \varphi}{\partial \xi_3} = \frac{\partial \varphi}{\partial \xi_2} - \frac{\partial \varphi}{\partial \xi_1}. \tag{3.1}$$

For any two finite element basis functions φ_α and φ_β with $\Delta \in \mathrm{supp}(\varphi_\alpha) \cap \mathrm{supp}(\varphi_\beta)$, we have for the bilinear form on the logical domain

$$\begin{aligned}\int_\Delta \left(\frac{\alpha}{2}\left(DF^{-T}\nabla_{r,\theta}\varphi_\alpha \, DF^{-T}\nabla_{r,\theta}\varphi_\beta\right)\right) |\det DF| \mathrm{d}(r,\theta) \\ = \int_T \left(\overline{b}^{\xi_1\xi_1}\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_1}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_1} + \overline{b}^{\xi_1\xi_2}\left(\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_1}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_2} + \frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_2}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_1}\right) + \overline{b}^{\xi_2\xi_2}\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_2}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_2}\right) \mathrm{d}(\xi_1,\xi_2),\end{aligned} \tag{3.2}$$

where $\widehat{\varphi_\alpha}$ and $\widehat{\varphi_\beta}$, $\widehat{\alpha}, \widehat{\beta} \in \{1, 2, 3\}$, are the corresponding functions on the reference element and where

$$\frac{\alpha}{2} D\widehat{F}^{-1} DF^{-1} DF^{-T} D\widehat{F}^{-T} |\det DF| |\det D\widehat{F}| =: \begin{pmatrix} \overline{b}^{\xi_1\xi_1} & \overline{b}^{\xi_1\xi_2} \\ \overline{b}^{\xi_1\xi_2} & \overline{b}^{\xi_2\xi_2} \end{pmatrix}; \tag{3.3}$$

with the mapping $\widehat{F}^{-1}(\Delta) = T$; cf. Fig. 3.

For the nonstandard quadrature rule, we first transform (3.2) by using (3.1) to

$$\int_T \left(b^{\xi_1\xi_1}\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_1}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_1} + b^{\xi_2\xi_2}\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_2}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_2} + b^{\xi_3\xi_3}\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_3}\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_3}\right) \mathrm{d}(\xi_1,\xi_2); \tag{3.4}$$

where

$$b^{\xi_1\xi_1} := \overline{b}^{\xi_1\xi_1} + \overline{b}^{\xi_1\xi_2}, \ b^{\xi_2\xi_2} := \overline{b}^{\xi_2\xi_2} + \overline{b}^{\xi_1\xi_2}, \ \text{and} \ b^{\xi_3\xi_3} := -\overline{b}^{\xi_1\xi_2}. \tag{3.5}$$

The numerical approximation of the integral (3.2) is then given by

$$|T| \sum_{n=1}^{3} b^{\xi_n\xi_n}\left(\xi^{(T,n)}\right)\frac{\partial\widehat{\varphi_\alpha}}{\partial\xi_n}\left(\xi^{(T,n)}\right)\frac{\partial\widehat{\varphi_\beta}}{\partial\xi_n}\left(\xi^{(T,n)}\right), \tag{3.6}$$

The linear form is approximated by using

$$\int_T g(\xi_1, \xi_2)\mathrm{d}(\xi_1, \xi_2) = \frac{|T|}{3} \sum_{i=1}^{3} g\left(z_i\right), \tag{3.7}$$

where z_i, $i \in \{1, 2, 3\}$, are the corner nodes; cf. Fig. 3.

3.2 Finite Difference Discretizations

For completeness, we additionally present the finite difference stencils as they will be used here. We refer to [18] for further detail. For any rectangular grid element $\square := (r_i, r_{i+1}) \times (\theta_j, \theta_{j+1})$ of the logical domain, we consider the discretized local energy function

$$\int_{\square} \left(a_*^{rr} u_r^2 + a_*^{r\theta} u_r u_\theta + a_*^{\theta\theta} u_\theta^2 - fu|\det DF_*|\right)\mathrm{d}(r, \theta) \tag{3.8}$$

corresponding to (1.1) and where a_*^{rr}, $a_*^{r\theta}$, and $a_*^{\theta\theta}$ are implicitly given by F_* as defined in (2.1) or (2.2), respectively.

Note that $a_*^{r\theta} = 0$ if $F_* = F_\mathrm{P}$ is the standard polar coordinate transformation. Note also that this does only generally hold for scalar diffusion α. For this case, we obtain the five point stencil

$$\begin{aligned}
\mathbf{u_{s+1,t}} : (*_5)_{s+1,t} &:= -\frac{k_t + k_{t-1}}{h_s} \frac{a_{s,t}^{rr} + a_{s+1,t}^{rr}}{2} \\
\mathbf{u_{s-1,t}} : (*_5)_{s-1,t} &:= -\frac{k_t + k_{t-1}}{h_{s-1}} \frac{a_{s-1,t}^{rr} + a_{s,t}^{rr}}{2} \\
\mathbf{u_{s,t+1}} : (*_5)_{s,t+1} &:= -\frac{h_s + h_{s-1}}{k_t} \frac{a_{s,t}^{\theta\theta} + a_{s,t+1}^{\theta\theta}}{2} \\
\mathbf{u_{s,t-1}} : (*_5)_{s,t-1} &:= -\frac{h_s + h_{s-1}}{k_{t-1}} \frac{a_{s,t-1}^{\theta\theta} + a_{s,t}^{\theta\theta}}{2} \\
\mathbf{u_{s,t}} : (*_5)_{s,t} &:= -\left[(*_5)_{s+1,t} + (*_5)_{s-1,t} + (*_5)_{s,t+1} + (*_5)_{s,t-1}\right],
\end{aligned} \tag{3.9}$$

with right hand side

$$\frac{(h_s + h_{s-1})(k_t + k_{t-1})}{4} f_{s,t}|\det DF_{s,t}| \tag{3.10}$$

and quadratic error convergence. Note that this stencil differs slightly in $\mathbf{u_{s+1,t}}$ and $\mathbf{u_{s-1,t}}$ when compared to [18]. This results from the use of the trapezoidal rule instead of the midpoint rule. It was chosen to have the five point stencil (3.9) as the reduced version of the nine point stencil (3.11).

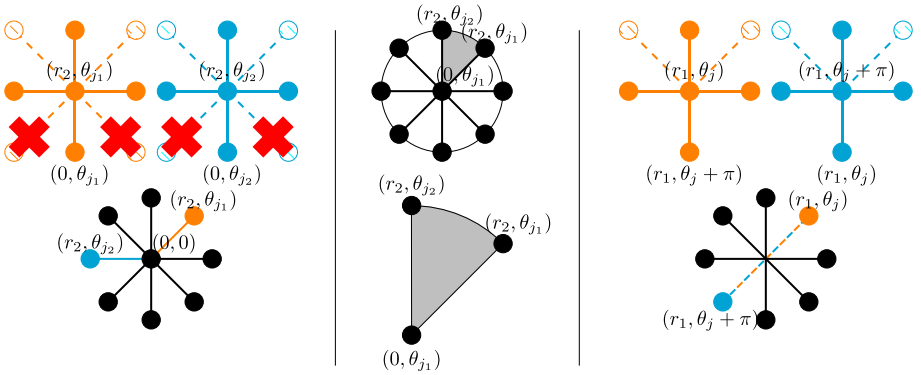

Fig. 4 Finite difference stencils around $r_1 = 0$ (left), finite element discretization around $r_1 = 0$ (center), finite difference discretization across the origin for finite differences and finite elements and $r_1 > 0$ (right)

In case of a transformation where $a_*^{r\theta} \neq 0$, we have to use a seven or nine point stencil, to obtain a quadratic discretization error. The nine point stencil used here is given by

$$
\begin{aligned}
\mathbf{u_{s+1,t}} : (*9)_{s+1,t} &:= (*5)_{s+1,t} \qquad \mathbf{u_{s-1,t}} : (*9)_{s-1,t} := (*5)_{s-1,t} \\
\mathbf{u_{s,t+1}} : (*9)_{s,t+1} &:= (*5)_{s,t+1} \qquad \mathbf{u_{s,t-1}} : (*9)_{s,t-1} := (*5)_{s,t-1} \\
\mathbf{u_{s+1,t+1}} : (*9)_{s+1,t+1} &:= -\frac{a^{r\theta}_{s+1,t} + a^{r\theta}_{s,t+1}}{4} \\
\mathbf{u_{s+1,t-1}} : (*9)_{s+1,t-1} &:= \frac{a^{r\theta}_{s,t-1} + a^{r\theta}_{s+1,t}}{4} \\
\mathbf{u_{s-1,t+1}} : (*9)_{s-1,t+1} &:= \frac{a^{r\theta}_{s-1,t} + a^{r\theta}_{s,t+1}}{4} \\
\mathbf{u_{s-1,t-1}} : (*9)_{s-1,t-1} &:= -\frac{a^{r\theta}_{s-1,t} + a^{r\theta}_{s,t-1}}{4} \\
\mathbf{u_{s,t}} : (*9)_{s,t} &:= -\left[(*9)_{s+1,t} + (*9)_{s-1,t} + (*9)_{s,t+1} + (*9)_{s,t-1}\right]
\end{aligned}
\tag{3.11}
$$

with right hand side (3.10). We refer to [18] for its derivation.

3.3 Handling of the Artificial Singularity

In the following, we propose some ways to handle the artificial singularity for $r \to 0$. All our proposals are based on the idea to retain a symmetric operator.

3.3.1 The Origin as Discretization Node

A natural approach consists in integrating the node $(r_1, \theta) = (0, 0)$ into the mesh. This, however, needs an adaptation of the discretization rules since the logical nodes $(0, \theta_j)$, $j = 1, \ldots, n_\theta$ all coincide geometrically.

For our finite difference stencils, we modify the discretization around the origin as following. Let us consider an arbitrary node (r_2, θ_j), $1 < j < n_\theta - 1$. We remove all interactions of the stencil with (r_1, θ_{j-1}) and (r_1, θ_{j+1}); see Fig. 4 (top left). We then take the interaction with (r_1, θ_j) and set it also as connection from (r_1, θ_j) to (r_2, θ_j) to obtain a symmetric

matrix. The diagonal entry for $(r_1, 0)$ is then given by the negative sum of the values on all angles.

For our finite element discretization, we integrate the basis functions over the triangles with nodes (r_1, θ_j), (r_2, θ_j), (r_2, θ_{j+1}); see Fig. 4 (center). Note that it is important to pass (r_1, θ_j) to the assembly of the transformation onto the reference angle, although it physically corresponds to $(r_1, 0)$ for all $1 \leq j \leq n_\theta$. If $(r_1, 0)$ is passed for all angles, the orthogonality of the mesh (i.e., the tridiagonal structure of T) is lost locally and the connections of (r_1, θ_{j_1}) to (r_2, θ_{j_1}) and (r_1, θ_{j_2}) to (r_2, θ_{j_2}) can differ for $j_1 \neq j_2$.

3.3.2 Artificial Boundary Conditions

A simple and often used workaround to overcome the problem of the artificial singularity is to choose $0 < r_1 \ll 1$ and to enforce Dirichlet or Neumann boundary conditions for the artificial boundary $r_1 \times [0, 2\pi]$. A direct drawback is that these conditions are hard or even impossible to determine in practical cases. This workaround is however used in the Gysela implementation as presented in [13].

3.3.3 Discretization Across the Origin

Another approach that we propose is the discretization across the origin. Instead of explicitly using $(0, 0)$ as discretization node or imposing boundary conditions at $r_1 > 0$, we first assemble the stiffness matrix for $r_1 > 0$ without any condition on (r_1, θ_j), $1 \leq j \leq n_\theta$.

To *discretize across the origin*, we only assume $n_\theta - 1$ to be even. For finite differences and finite elements likewise, we then take the finite difference stencil entry $(*5)_{-1,j}$ or $(*9)_{-1,j}$ with $r_{-1} = r_1$ and $h_{-1} = 2r_1$ since the geometrical distance is $2r_1$ between the nodes (r_1, θ_j) and $(r_1, \theta_j + \pi)$ to define a stencil entry from (r_1, θ_j) to $(r_1, \theta_j + \pi)$ (and vice versa).

Note that this may lead to an unsymmetric operator if nonsymmetric domains and seven point stencils are considered. In this case, one could copy the values from the first half circle to the second half circle to retain a symmetric operator.

4 Geometric Multigrid for Curvilinear Coordinates

Multigrid methods are among the most efficient solvers for elliptic model problems such as (1.1); see, e.g., [6, 35]. Multigrid methods for meshes in polar coordinates were considered in, e.g., [1, 3, 23, 33, 35] but are, however, less studied. In the following sections, we will develop special multigrid components for the model problem in curvilinear coordinates such as the generalized polar coordinates proposed in (2.2).

In order to define the notation, we first define a hierarchy of $L + 1$ grids with $\Omega_{l-1} \subset \Omega_l$, $1 \leq l \leq L$, and $|\Omega_L| = n_r * n_\theta$. To identify matrices and vectors on grid Ω_l, we use the subindex l, $0 \leq l \leq L$. The iterates of step m are characterized by a superindex m, $m \geq 0$. The restriction operator from grid l to grid $l - 1$ is denoted I_l^{l-1} and I_{l-1}^l represents the interpolation from grid $l - 1$ to grid l. The presmoothing operation with ν_1 steps is denoted $\mathbf{S}^{\nu_1}$, the postsmoothing operation with ν_2 steps is denoted $\mathbf{S}^{\nu_2}$. The multigrid cycle $u_L^{m+1} = \mathbf{MGC}(L, \gamma, u_L^m, A_L, f_L, \nu_1, \nu_2)$ is then given recursively for $0 \leq l \leq L$.

The multigrid cycle

$u_l^{m+1} = \mathbf{MGC}(l, \gamma, u_l^m, K_l, f_l, \nu_1, \nu_2)$

- Presmoothing: $u_l^{m+1/3} = \mathbf{S}^{\nu_1}(u_l^m, K_l, f_l)$
- Coarse grid correction
  - Compute the residual: $r_l^{m+2/3} = f_l - K_l u_l^{m+1/3}$
  - Restrict the residual: $r_{l-1}^{m+2/3} = I_l^{l-1} r_l^{m+2/3}$
  - Solve $A_{l-1}\widehat{e}_{l-1}^{m+2/3} = r_{l-1}^{m+2/3}$ by
    - (if $l = 0$:) the use of a direct solver.
    - (if $l \geq 1$:) γ-times recursively calling
      $\widehat{e}_{l-1}^{m+2/3} = \mathbf{MGC}(l-1, \gamma, \diamond, K_{l-1}, r_{l-1}^{m+2/3}, \nu_1, \nu_2)$
  - Interpolate the correction: $\widehat{e}_l^{m+2/3} = I_{l-1}^l \widehat{e}_{l-1}^{m+2/3}$
  - Compute the corrected approximation: $u_l^{m+2/3} = u_l^{m+1/3} + \widehat{e}_l^{m+2/3}$
- Postsmoothing: $u_l^{m+1} = \mathbf{S}^{\nu_2}(u_l^{m+2/3}, K_l, f_l)$

In the recursive call, $\diamond$ stands for zero as a first approximation and in further calls (W-cycle) for an approximation taken from the previous cycle.

4.1 Optimized Zebra Line Smoothers

For highly anisotropic problems, point relaxation and standard coarsening (i.e., coarsening by a factor of 2 in each dimension) do not yield satisfactory results. Pointwise smoothing then only has poor smoothing properties with respect to *weakly-coupled* degrees of freedom (dofs); cf. [35,Sec. 5.1]. In the context of multigrid, we speak of *strong coupling* between one dof to another if the offdiagonal entry of the considered matrix is "relatively" large; compared to the other offdiagonal entries of the same dof. If the entry is "relatively" small, we speak of *weak coupling*.

If the anisotropy is aligned with the grid, standard coarsening can be kept and only the smoothing operation has to be adapted to obtain good multigrid performance. Line relaxations are block relaxations where all the connections between degrees of freedom of one line are taken into account to update this line in one single step. Using line relaxation, errors become smooth if strongly connected degrees of freedom are updated together. For a more detailed introduction to line smoothers, see, e.g., [35,Sec. 5.1].

For compact finite difference stencils and linear nodal basis functions, zebra line smoothers correspond to Gauß-Seidel line relaxation methods where all even and all odd lines (rows or columns), respectively, are processed simultaneously. For operators where the anisotropy changes across the domain, alternating zebra relaxation has been proposed; see [33]. The polar coordinate transformed Laplace operator yields strong connections on circle lines on the interior part of the domain and strong connections on radial lines on the outer part; cf. (2.4). Consequently, alternating zebra relaxation was proposed for the unit disk [1]. We will now briefly introduce zebra relaxation and then explain our particular choice of smoothers for all parts of the (deformed) domain from Fig. 1 described by curvilinear coordinates.

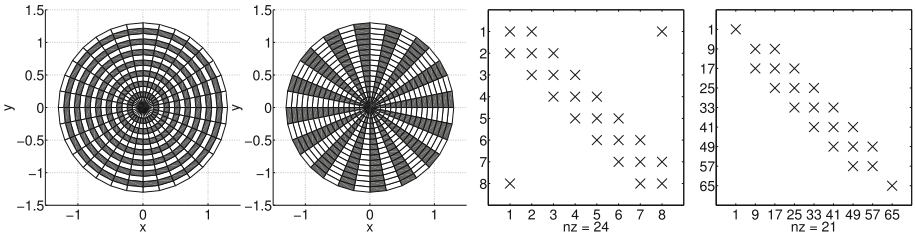

Fig. 5 Circle (left) and radial (second to left) zebra coloring for the equidistant discretized annulus with $r_1 = 1e-6$. Nonzero pattern of $K_{l,BB}$ restricted to an interior circle (second to right) and of $K_{l,BB}$ restricted to one radial direction (right) assuming a circle-by-circle numeration of the nodes with $n_{l,r} = n_{l,\theta} = 9$. The periodic boundary conditions at $(r_{l,i}, \theta_{n_{l,\theta}}) = (r_{l,i}, 2\pi)$ introduce the interaction in the upper right and lower left corners of the the circle relaxation operator (second to right). The first and last line of radial relaxation operator (right) only have one entry since Dirichlet boundary conditions were set there, entries $\cdot_{1,9}$ and $\cdot_{57,65}$ were put on the right hand side; only the corresponding nonzero rows and columns are printed

Let $n_l = n_{l,r} \times n_{l,\theta}$ be the number of nodes on grid $l \in \{0, L\}$. Furthermore, let B_l and W_l be disjoint index sets such that $B_l \cup W_l = \{1, 2, \ldots, n_l\}$ and by reordering

$$u_l^m = \begin{pmatrix} u_{l,B}^m \\ u_{l,W}^m \end{pmatrix}, \quad f_l = \begin{pmatrix} f_{l,B} \\ f_{l,W} \end{pmatrix}, \quad \text{and } K_l = \begin{pmatrix} K_{l,BB} & K_{l,BW} \\ K_{l,WB} & K_{l,WW} \end{pmatrix} \tag{4.1}$$

for any grid $l \in \{0, L\}$. Note that we drop the second index l in B and W to avoid a proliferation of indices.

In the following, we will focus on zebra colorings such that $K_{l,BB}$ and $K_{l,WW}$ can be partitioned into a block diagonal system with blocks of size $\mathcal{O}(\sqrt{n_l})$. Note that this property does not hold for the radial directions if a full (deformed) disk is considered; if $r_1 = 0$, then all these directions are coupled by the origin.

For curvilinear coordinates, the two natural line smoothing operations are denoted circle and radial zebra relaxation. For circle zebra relaxation, all nodes $(r_{l,i}, \theta_{l,j})$, $j \in \{1, \ldots, n_{l,\theta}\}$ get the same color while $(r_{l,i-1}, \theta_{l,j})$ and $(r_{l,i+1}, \theta_{l,j})$, $j \in \{1, \ldots, n_{l,\theta}\}$, get another color. For radial zebra relaxation $(r_{l,i}, \theta_{l,j})$, $i \in \{1, \ldots, n_{l,r}\}$ are colored together; see Fig. 5 (left and second to left).

Let us color each line (row or column) alternatingly black and white. Then, the diagonal blocks of size $\mathcal{O}(\sqrt{n_l})$ in $K_{l,BB}$ and $K_{l,WW}$ only have three entries per row for all finite difference stencils and finite element basis functions introduced in Sect. 3. For a coloring in accordance with the ordering of the nodes, the local block can be tridiagonal. However, also the banded systems with three entries per row can be solved in $\mathcal{O}(\sqrt{n})$ operations by a direct solver; see Fig. 5 (second to right and right) for the nonzero structure.

The presmoothing operation $\mathbf{S}^{\nu_1}(u_l^m, K_l, f_l)$ can be expressed as follows.

The circle or radial presmoothing operation

$$
\begin{aligned}
&u_{l,B}^{m,0} = u_{l,B}^{m}, u_{l,W}^{m,0} = u_{l,W}^{m} \\
&\textbf{for } i = 1, \ldots, \nu_1 \textbf{ solve} \\
&\qquad K_{l,BB} u_{l,B}^{m,i} = f_{l,B} - K_{l,BW} u_{l,W}^{m,i-1} \\
&\qquad K_{l,WW} u_{l,W}^{m,i} = f_{l,W} - K_{l,WB} u_{l,B}^{m,i} \\
&\textbf{endfor} \\
&u_{l,B}^{m+\frac{1}{3}} = u_{l,B}^{m,\nu_1}, u_{l,W}^{m+\frac{1}{3}} = u_{l,W}^{m,\nu_1}
\end{aligned}
\tag{4.2}
$$

The postsmoothing operation $\mathbf{S}^{\nu_2}(u_l^{m+\frac{2}{3}}, K_l, f_l)$ is obtained equivalently. In order to smooth the coarse degrees of freedom first, we will color them always in black.

Remark 4.1 Note that the zebra-line Gauss-Seidel preconditioner is not triangular but block-triangular. That means that all nonzero entries shown in Fig. 5 (also those in the upper triangular part) remain on the left hand side of the system. The shown entries all belong to the same line (row or column). For larger finite difference stencils or hierarchical finite element bases, $K_{l,BB}$ and $K_{l,WW}$ from (4.2) may have more than three nonzeros per row. Then, either more colors have to be used or a part of the upper triangular matrix has to be brought to the right hand side.

Let us consider the annulus $\overline{\Omega}_{h_i} := [r_i, r_i + h_i] \times [0, 2\pi]$ as an individual domain; with a constant discretization parameter $k_j = k$ in the second dimension, i.e, $n_\theta k = 2\pi$. From [1], we know that the smoothing factors of circle and radial relaxation, $\mu_{\mathrm{CZ},h_i,k_j}$ and $\mu_{\mathrm{RZ},h_i,k_j}$, on $\overline{\Omega}_{h_i}$ are given by

$$
\begin{aligned}
\mu_{\mathrm{CZ},h_i,k_j} &= \max_{r_i \le r \le r_i + h_i} \left\{ \left(\frac{q_{i,j}^2 r^2}{1 + q_{i,j}^2 r^2} \right)^2, C_C \right\} \\
\mu_{\mathrm{RZ},h_i,k_j} &= \max_{r_i \le r \le r_i + h_i} \left\{ \left(\frac{1}{1 + q_{i,j}^2 r^2} \right)^2, C_R \right\}
\end{aligned}
\tag{4.3}
$$

with $q_{i,j} = \frac{k_j}{h_i}$ as well as $C_C \in \{0.23, 0.34\}$, depending on $r_i \ge 0$, and $C_R = 0.23$, independently of r_i. From Fig. 6 (left), we see that both relaxations behave very differently on different annuli of size h_i of the global domain. We see that radial relaxation is prohibitive around the origin but shows good smoothing behavior for $r \to 1.3$. Circle relaxation shows good smoothing behavior around the origin but does not provide essential smoothing where the mesh was refined and for $r \to 1.3$. In order to obtain a reasonable smoothing procedure on the entire domain, we thus have to combine circle relaxation with radial relaxation. In [1], alternating zebra relaxation, consisting of one step with each smoothing operator, was proposed.

To reduce the workload and to optimize the smoothing operation, we propose the following smoothing procedure. Since circle relaxation leads to good smoothing around the origin, we color the nodes around the origin in circle lines. For each following circle with radius $r_i > r_1$, we then check in accordance to (4.3), if

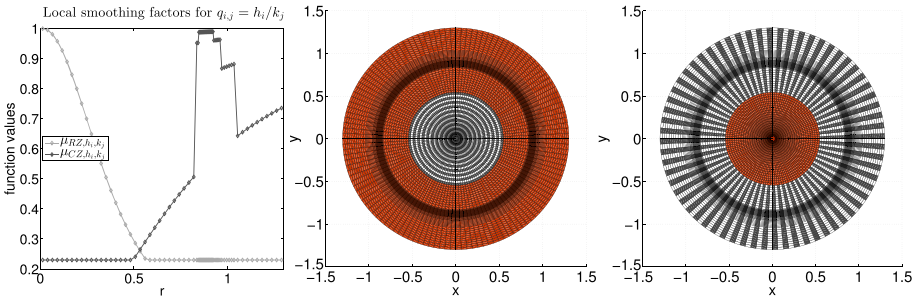

Fig. 6 Approximated local smoothing factors $\mu_{\mathrm{CZ},h_i,k_j}$ and $\mu_{\mathrm{RZ},h_i,k_j}$ for a finer discretization of the mesh depicted in Fig. 1 (left) with $r_1 = 1e-6$. Approximation by evaluating the argument of the maximum functions in (4.3) at r_i; we use that k_j is constant on each circle line represented by $r_i, i = 1, \ldots, n_r$. Domain decomposition and optimized circle and radial smoothers (center and right). The red parts of the domain are not smoothed by the corresponding smoothing operation

$$q_{i,j}^2 r^2 > 1 \quad \Leftrightarrow \quad \frac{k_j}{h_i} r_i > 1 \tag{4.4}$$

and change to radial relaxation if this is the case. Note that we use that k_j is constant on each circle line represented by r_i, $i = 1, \ldots, n_r$. We then obtain a decomposition of the domain into two domains, where different relaxation methods are used; see Fig. 6.

Although the decomposition rule (4.4) was developed for a domain described by polar coordinates, we also use this as a rule of thumb for the deformed geometries described by transformation (2.2). See Sect. 5.2 for a numerical evaluation.

The optimized presmoothing operation $\mathbf{S}^{\nu_1}(u_l^m, K_l, f_l)$ is then given with six colors: black (for circle and radial, denoted B_C and B_R), white (for circle and radial, denoted W_C and W_R), and orange (denoted O_C and O_R), which itself is not smoothed; see Fig. 6. The values of the previous half-step of relaxation are implicitly used as Dirichlet boundary conditions on the orange-colored part of the decomposition.

The optimized circle-radial presmoothing operation

$$u_{l,B_*}^{m,0} = u_{l,B_*}^m, u_{l,W_*}^{m,0} = u_{l,W_*}^m, u_{l,O_*}^{m,0} = \begin{pmatrix} u_{l,B_{*^\perp}}^m \\ u_{l,W_{*^\perp}}^m \end{pmatrix} \text{ for } * \in \{C, R\}$$

Note: $*^\perp = R$ if $* = C$ and vice versa.

for $i = 1, \ldots, \nu_1$

for $* \in \{C, R\}$ **solve**

$$\begin{aligned} K_{l,B_*B_*} u_{l,B_*}^{m,i} &= f_{l,B_*} - K_{l,B_*W_*} u_{l,W_*}^{m,i-1} - K_{l,B_*O_*} u_{l,O_*}^{m,i-1} \\ K_{l,W_*W_*} u_{l,W_*}^{m,i} &= f_{l,W_*} - K_{l,W_*B_*} u_{l,B_*}^{m,i} - K_{l,W_*O_*} u_{l,O_*}^{m,i-1} \end{aligned} \tag{4.5}$$

update: $u_{l,O_{*^\perp}}^{m,i-1} = \begin{pmatrix} u_{l,B_*}^{m,i} \\ u_{l,W_*}^{m,i} \end{pmatrix}$

endfor

endfor

$$u_{l,B_*}^{m+\frac{1}{3}} = u_{l,B_*}^{m,\nu_1}, u_{l,W_*}^{m+\frac{1}{3}} = u_{l,W_*}^{m,\nu_1} \text{ for } * \in \{C, R\}$$

The values $u_{l,O_*}^{m,i}$ on the orange colored part of the domain contain the interface boundary conditions for each half-step of smoother. Note that only those values next to the interior interface, which represent the interface boundary conditions, have to be updated in each step of the iterative process. The larger the stencil, the more lines have to be updated in practice.

Note that (4.5) is not parallelized across the two different smoothers (circle and radial) but that the radial smoothers use information from the circle smoothers. If no information is exchanged, an increase in iterations is to be expected. However, if the color of the outermost circle-smoother line is smoothed first, then for compact FE or FD stencils such as provided in this paper, the two sequential colors of the radial smoothers can be executed in parallel with the second color of the circle smoother. Since the circle color lines are of larger size than the radial color lines, both parallel steps are expected to finish at similar times.

4.2 Coarsening and Intergrid Transfer Operators

The coarsening and intergrid transfer operators use the classical choices. We always employ standard coarsening and we use bilinear interpolation, which is also well-defined for anisotropic meshes, if the additional extrapolation algorithm is not used. In case of implicit extrapolation, we use bilinear interpolation for $l = 1, \ldots, L-1$ only and transfer between the two finest grids is adapted. As presented in Sect. 4.3, extrapolation will only affect the transfer between the two finest grid levels. In case $(0, 0)$ is an actual discretization node and is chosen as first coarse node, we have to adapt the restriction and prolongation there. Our restriction operator is always defined as the adjoint

$$I_l^{l-1} = \left(I_{l-1}^l\right)^T, \quad l = 1, \ldots, L. \tag{4.6}$$

Remark 4.2 [Scaling between prolongation and restriction] Note that there is no scaling constant in definition (4.6) since for the finite element discretizations as well as for our tailored finite difference schemes, the right hand side is locally scaled with $\mathcal{O}(h_i k_j)$, $1 \leq i \leq n_r$ and $1 \leq j \leq n_\theta$; cf. [18] for details on the derivation of the finite difference stencils. As a potential source of implementation error, this has to be taken into account.

4.3 Implicit Extrapolation

In this section, we introduce the implicit extrapolation step within our multigrid algorithm, based on the extrapolation strategy of [15, 16, 18]. The extrapolation step is only conducted between the two finest levels of multigrid hierarchy, affecting the operators on and interpolation between Ω_L and Ω_{L-1}.

Let us assume that the coarse degrees of freedom are ordered before the fine degrees of freedom. By using the indices $\cdot_c$ for coarse and $\cdot_f$ for fine nodes, we have

$$K_L = \begin{pmatrix} K_{L,cc} & K_{L,cf} \\ K_{L,fc} & K_{L,ff} \end{pmatrix}, \quad f_l = \begin{pmatrix} f_{L,c} \\ f_{L,f} \end{pmatrix}, \quad u_l^m = \begin{pmatrix} u_{L,c}^m \\ u_{L,f}^m \end{pmatrix},$$

and equivalently for any other entity defined on Ω_L.

In accordance to [15,p. 173], we present the new smoothing procedure that excludes coarse grid nodes from the (pre- or post-)smoothing procedure

$$\begin{aligned} u_{L,f}^{m+1/3} &= \mathbf{S}^{\nu_1}(u_{L,f}^m, K_{L,ff}, f_{L,f} - K_{L,fc} u_{L,c}^m) \\ \text{and} \quad u_{L,f}^{m+1} &= \mathbf{S}^{\nu_2}(u_{L,f}^m, K_{L,ff}, f_{L,f} - K_{L,fc} u_{L,c}^{m+2/3}) \end{aligned} \tag{4.7}$$

The new smoother on the finest level is the previously defined smoother only acting on the fine nodes.

Remark 4.3 Only the fine grid nodes are smoothed on the first level and the nodes belonging to the coarse grid are excluded from the smoothing operation. This differs from the introduction of τ-extrapolation in [2, 6, 14]. The weaker smoother may lead to a reduced algebraic convergence of the multigrid iteration, but it has the advantage that the fixed point of the multigrid iteration is uniquely defined. For more details, we refer to [15,p. 173] and the references therein.

Before presenting the extrapolated multigrid cycle, we must also introduce the modified intergrid transfer operators I_{L-1}^{L} and $I_{L}^{L-1} := (I_{L-1}^{L})^T$. In order to do so, denote by $\mathcal{T}_{L-1}$ the triangulation on Ω_{L-1}. We then define

$$I_{L-1}^{L} := \begin{pmatrix} I_c \\ T_{fc} \end{pmatrix}, \tag{4.8}$$

where I_c is the identity matrix on the coarse degrees of freedom and

$$\left(T_{fc}\right)_{s-n_{L-1},t} := \begin{cases} \frac{1}{2}, & \text{if there exists an edge } e \text{ in } \mathcal{T}_{L-1} \text{ s.t. } x_s \in e \text{ and } x_t \in \partial e, \\ 0, & \text{otherwise.} \end{cases}$$

Note that edges are open sets, i.e., $\overset{\circ}{e} = e$.

The implicitly extrapolated multigrid cycle $u_L^{m+1} = \mathbf{IEMGC}(L, \gamma, u_L^m, K_L, f_L, \nu_1, \nu_2)$ is then given as in [15,Algorithm 1].

The implicitly extrapolated multigrid cycle

$u_L^{m+1} = \mathbf{IEMGC}(L, \gamma, u_L^m, K_L, f_L, \nu_1, \nu_2)$

- Presmoothing: $u_{L,f}^{m+1/3} = \mathbf{S}^{\nu_1}(u_{L,f}^m, K_{L,ff}, f_{L,f} - K_{L,fc}u_{L,c}^m)$
- Define iterate: $u_L^{m+1/3} = \begin{pmatrix} u_{L,c}^m \\ u_{L,f}^{m+1/3} \end{pmatrix}$
- Coarse grid correction
  - Compute and restrict the residual: $r_{L-1}^{m+2/3} = \frac{4}{3} I_L^{L-1}(f_L - K_L u_L^{m+1/3}) - \frac{1}{3}(f_{L-1} - K_{L-1}u_{L,c}^{m+1/3})$
  - Call a standard multigrid cycle on $L-1$ levels: $\widehat{e}_{L-1}^{m+2/3} = \mathbf{MGC}(L-1, \gamma, 0, K_{L-1}, r_{L-1}^{m+2/3}, \nu_1, \nu_2)$
  - Interpolate and correct approximation: $u_l^{m+2/3} = u_l^{m+1/3} + I_{L-1}^{L}\widehat{e}_{L-1}^{m+2/3}$
- Postsmoothing: $u_{L,f}^{m+1} = \mathbf{S}^{\nu_2}(u_{L,f}^m, K_{L,ff}, f_{L,f} - K_{L,fc}u_{L,c}^{m+2/3})$
- Define iterate: $u_L^{m+1} = \begin{pmatrix} u_{L,c}^{m+2/3} \\ u_{L,f}^{m+1} \end{pmatrix}$

Remark 4.4 In [15,pp. 169f], it was shown that the implicitly extrapolated multigrid algorithm for linear elements can be interpreted as a multigrid algorithm solving the original PDE when discretized by quadratic nodal basis functions.

In [15], only constant coefficients were considered. Note that in our applications, due to the transformation of the physical domain, even $\alpha \equiv 1$ leads to nonconstant coefficients; cf. Sect. 2. Nonconstant coefficients were considered with hierarchical bases in [16]. In contrast to [16], we use the intergrid transfer operator given in [15]. This results from the discretization by nodal basis functions.

The proof of Remark 4.4 is based on the relation between linear nodal, linear quadratic, and h- and p-hierarchical basis functions. The transfer operator I_{L-1}^{L} is part of the transformation between a nodal and a hierarchical basis; see also [18,Sec. 4.4.1]. The necessary relations [15,(55) and (56)] are formally proven for nonconstant coefficients in [18,Lemma 4.2, Theorem 4.3]. In particular, we can write

$$\begin{aligned}&\frac{4}{3} I_L^{L-1}\left(f_L - K_L u_L^{m+1/3}\right) - \frac{1}{3}\left(f_{L-1} - K_{L-1} u_{L,c}^{m+1/3}\right)\\ &= I_L^{L-1}\Big[\underbrace{\begin{pmatrix}\frac{4}{3} f_{L,c} - \frac{1}{3} f_{L-1}\\ \frac{4}{3} f_{L,f}\end{pmatrix}}_{=:f_L^{ex}} - \underbrace{\begin{pmatrix}\frac{4}{3} K_{L,cc} - \frac{1}{3} K_{L-1} & \frac{4}{3} K_{L,cf}\\ \frac{4}{3} K_{L,fc} & \frac{4}{3} K_{L,ff}\end{pmatrix}}_{=:K_L^{ex}} u_L^{m+1/3}\Big],\end{aligned} \tag{4.9}$$

where the term in brackets corresponds to the residual computation of the quadratic approach.

Remark 4.5 We note that the direct discretization of a PDE with higher order finite elements will typically lead to a denser matrix structure and consequently to a higher flop cost per matrix-vector multiplication or smoother application. Here, we construct an equivalent higher order discretization using by way of a clever recombination of low order components as they arise canonically in a multigrid solver. In this way, we avoid the explicit set up of any more expensive higher order discrete operator. In other words, the implicit extrapolation multigrid method leads to a qualitatively equivalent high order discretization at reduced cost. This reduces memory cost avoiding the setup of more densely populated matrices, and they avoid the corresponding memory traffic and higher flop cost incurred in each iteration of an iterative solution process. In fact, except the computation of the extrapolated residual in the restriction phase, the cost of the extrapolated multigrid algorithm is the same as for standard low order discretization.

Of course, this alone does not account for other solver cost such as induced by the possibly slower (algebraic) convergence of the extrapolated multigrid algorithm (meaning that more iterations are needed) and the need to solve the discrete system with higher (algebraic) accuracy in order to exploit the lower discretization error. Because of these two effects the cost of computing a proper solution with the extrapolated multigrid algorithm is still expected to be more expensive than solving for a low order discretization. For an in-depth analysis of the so-called *textbook efficiency* of parallel multigrid algorithms, see also [11, 17].

5 Numerical Results

In this section, we study (1.1) with α as given in (2.4) to model the density of the fusion plasma according to [32, 38]. As test case for our new method, we use the manufactured solution

$$u(x, y) = (1.3^2 - r^2(x, y)) \cos(2\pi x) \sin(2\pi y), \tag{5.1}$$

where $r(x, y)$ is defined by (2.1) or (2.2). The right hand side f and the Dirichlet boundary conditions on $(r, \theta) \in 1.3 \times [0, 2\pi]$ are given accordingly. This example is taken from [39].

We thank Edoardo Zoni for providing his Python script for symbolic differentiation and, in the interest of saving space, we refrain from representing the right hand side explicitly.

We use an anisotropic discretization in $r \in [r_1, 1.3]$ with $r_{i+1} = r_i + h_i$, $i = 1, \ldots, n_r$ to account for the density profile drop in the separatrix' edge area; cf. [12, 38]. We restrict the anisotropy to $h = \max_i h_i = 8 \min_i h_i$.

For our multigrid algorithm, we conduct one step of pre- and one step of postsmoothing, i.e., $\nu = \nu_1 + \nu_2 = 2$. In prospect of a parallel implementation, we only use V-cycles. We use a strong convergence criterion by demanding a relative residual reduction by a factor of 10^8. The maximum number of iterations is set to 150. In all tables, we provide the finest mesh size as $n_r \times n_\theta$. We also provide the iteration count of the multigrid algorithm needed to convergence as *its* as well as

$$\widehat{\rho} = \sqrt[\text{its}]{\frac{\|r_L^{\text{its}}\|_2}{\|r_L^0\|_2}}, \tag{5.2}$$

the mean residual reduction factor. Note that the measured $\widehat{\rho}$ is generally slightly smaller than the theoretical reduction factor and becomes more precise when more iterations are executed. For all simulations, we present the error of the iterative solution compared to the exact solution evaluated at the nodes in the (weighted) $\|\cdot\|_{\ell_2}$-norm, defined by

$$\|v\|_{\ell_2} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} v_i^2},$$

and the $\|\cdot\|_\infty$-norm. We also provide the error reduction order as *ord.* for both norms. The order is here calculated as the error reduction from one row to its predecessor, i.e.,

$$\text{ord} = \log\left(\frac{\|err_{k-1}\|}{\|err_k\|}\right) / \log\left(\sqrt{\frac{gridsize_k}{gridsize_{k-1}}}\right).$$

The orders were computed with errors norms rounded to the fifth nontrivial digit.

Remark 5.1 (Residual and algebraic error convergence) As mentioned in Remark 4.4, the implicitly extrapolated multigrid algorithm can be considered as a multigrid algorithm based on a second order discretization. Consequently, we require for the residual

$$\|r_L^m\| := \|f_L^{ex} - K_L^{ex} u_L^m\| \le 10^{-8} \|f_L^{ex} - K_L^{ex} u_L^0\| =: 10^{-8} \|r_L^0\|; \tag{5.3}$$

for which the norm is directly available from the multigrid context; cf. (4.9).

We test several different configurations and provide comparisons in the following sections.

- In Sect. 5.1, we show that neither circle nor radial relaxation alone are sufficient to obtain fast convergence of our multigrid algorithm. Our choice of optimized circle-radial relaxation always leads to fast convergence.
- In Sect. 5.2, we show that (4.5) results in an optimal domain decomposition to execute the optimized smoothing operations,
- In Sect. 5.3, we compare the multigrid algorithm based on finite elements with nonstandard integration and finite difference discretizations for different approaches to handle the artificial singularity from Sect. 3.3.
- In Sect. 5.4, we proceed similarly to Sect. 5.3 by using the multigrid algorithm with implicit extrapolation as described in Sect. 4.3.

Table 1 Comparison of zebra line smoothers

	Circle smoothing				Radial smoothing				Optimized smoothing			
$n_r \times n_\theta$	its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	$\|err\|_\infty$	its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	$\|err\|_\infty$	its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	$\|err\|_\infty$
Circular geometry (2.1)–FD 5p stencil (3.9)												
49×64	150	0.93	5.1e-02	9.6e-02	150	0.92	5.1e-02	9.6e-02	13	0.23	5.1e-02	9.6e-02
97×128	150	0.94	1.3e-02	2.4e-02	150	0.96	1.3e-02	2.4e-02	13	0.23	1.3e-02	2.4e-02
193×256	150	0.94	3.2e-03	5.9e-03	150	0.96	3.2e-03	6.0e-03	13	0.22	3.2e-03	6.0e-03
385×512	150	0.95	8.0e-04	1.5e-03	150	0.97	8.6e-04	4.9e-03	13	0.22	8.0e-04	1.5e-03
Deformed geometry (2.2)–FD 9p stencil (3.11)												
49×64	150	0.98	7.6e-02	1.5e-01	150	0.95	7.1e-02	1.5e-01	46	0.67	7.1e-02	1.5e-01
97×128	150	0.98	3.4e-02	1.4e-01	150	0.97	1.8e-02	4.1e-02	45	0.66	1.8e-02	4.1e-02
193×256	150	0.98	3.0e-02	1.4e-01	150	0.97	4.7e-03	1.5e-02	44	0.66	4.5e-03	1.1e-02
385×512	150	0.98	2.9e-02	1.4e-01	150	0.97	1.6e-03	1.5e-02	44	0.65	1.1e-03	2.6e-03

Multigrid *without extrapolation* based on *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 = 1e-8$ and *Dirichlet boundary conditions* on the *innermost circle*. Iteration counts its (*max. its*=150), mean residual reduction factor $\widehat{\rho}$, and errors of iterative solution to exact solution evaluated at the nodes in $\|\cdot\|_{\ell_2}$ and $\|\cdot\|_\infty$ norms

Remark 5.2 (Expectations on convergence orders) For the non-extrapolated versions of our solvers, we expect quadratic error convergence in the ℓ_2-norm, as, e.g., predicted by the derivation of the finite difference stencils. For the implicitly extrapolated version, we expect quartic convergence for a non-refined grid and to come close to 3.7 if a fast descending diffusion coefficient and local grid refinement is used. For more details, see [18].

5.1 Multigrid with Circle, Radial, and Optimized Circle-Radial Relaxation

In this section, we study different smoothing procedures: circle smoothing, radial smoothing, and our optimized circle-radial smoothing described in Sect. 4.1 and denoted as *optimized smoothing* in Table 1. For $r_1 = 0$, radial relaxation is prohibitive since the origin couples all directions. We thus choose $r_1 = 1e-8$ and enforce Dirichlet boundary conditions on the innermost circle to avoid an additional influence of the artificial singularity.

From Table 1, we see that neither circle nor radial smoothing alone are sufficient to obtain satisfactory residual reduction factors. Note that pointwise smoothers yielded even worse results. The optimized smoother, although smoothing each node also only once, yields good results (i.e., quadratic error reduction).

5.2 Multigrid with Optimized Circle-Radial Relaxation

In this section, we numerically show the optimality of our circle-radial domain decomposition by testing it against other decompositions. As a basic rule, we use (4.5). We compare this optimized smoother with other decompositions where we color $\pm n$ (additional or less) circles, $n \in \mathbb{N}$, circle by circle and the remaining part in a radial manner.

Table 2 shows that the optimal residual reduction factor, as well as the minimum number of iterations, is obtained with rule (4.5).

Table 2 Smoother optimization

		Circular geometry—FD 5p		Deformed geometry—FD 9p	
$n_r \times n_\theta$	Decomp	Its	$\widehat{\rho}$	Its	$\widehat{\rho}$
145×256	(4.5)	8	0.09	19	0.36
	(4.5) − 4	9	0.11	19	0.36
	(4.5) − 8	11	0.16	22	0.43
	(4.5)+4	15	0.27	30	0.53
	(4.5)+8	26	0.48	48	0.68

Multigrid *without extrapolation* based on *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 = 1e-8$ and *Dirichlet boundary conditions* on the *innermost circle*. Different decompositions of the domain and influence of the optimized circle-radial smoothing operators. For decomp, $(4.5 \pm n, n \in \mathbb{N})$, means that $\pm n$ circles are colored circle-wise instead of radial-wise as proposed by (4.5). Further notation as in Table 1

Table 3 Comparison of discretizations

$n_r \times n_\theta$	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.
Circular geometry												
	FE P1 (nonstandard integ.)						FD 5p					
49×64	25	0.47	5.9e-02	–	1.0e-01	–	25	0.47	5.2e-02	–	9.6e-02	–
97×128	23	0.44	1.6e-02	1.93	4.5e-02	1.23	23	0.44	1.3e-02	2.02	2.4e-02	2.02
193×256	23	0.43	4.1e-03	1.93	2.4e-02	0.91	23	0.43	3.2e-03	2.01	6.0e-03	2.01
385×512	22	0.43	1.1e-03	1.89	1.2e-02	0.98	22	0.43	8.0e-04	2.00	1.5e-03	2.00
Deformed geometry												
	FE P1 (nonstandard integ.)						FD 9p					
49×64	88	0.81	7.3e-02	–	1.6e-01	–	88	0.81	7.2e-02	–	1.5e-01	–
97×128	79	0.79	1.9e-02	1.94	4.8e-02	1.77	80	0.79	1.8e-02	2.00	4.1e-02	1.87
193×256	76	0.78	4.9e-03	1.98	1.3e-02	1.93	76	0.78	4.6e-03	2.00	1.1e-02	1.96
385×512	74	0.78	1.2e-03	1.98	5.7e-03	1.15	74	0.78	1.1e-03	2.00	2.6e-03	1.99

Multigrid *without extrapolation* based on *finite element* and *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 = 0$. Error reduction given by ord.; further notation as in Table 1

5.3 Multigrid Based on Different Discretizations

In this section, we consider different ways to handle the artificial singularity as proposed in Sect. 3.3. We consider the case where $r_1 = 0$, i.e., where $(0, 0)$ is a node on the grid as well as $r_1 \in \{10^{-2}, 10^{-5}, 10^{-8}\}$. For $r_1 > 0$, we consider the case of Dirichlet boundary conditions as well as our strategy of *discretizing across the origin*; cf. Sect. 3.3.3. We observe that we obtain identical results for the configurations with Dirichlet boundary conditions on the innermost circle and by discretizing across the origin, respectively, if $r_1 \to 0$; cf. Table 5.

We consider the multigrid algorithm based on finite element discretization with nonstandard integration techniques and on the finite difference five and nine point stencil, respectively, where the latter is only used if the deformed geometry is considered.

From Tables 3, 4, and 5, we first see that the multigrid algorithm needs about twice as many iterations if $(0, 0)$ is used as explicit mesh node. For the circular geometry and $r_1 > 0$, we only need 13 iterations to reduce the residual by a factor of 10^8. The number of iterations

Table 4 Comparison of discretizations

$n_r \times n_\theta$	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.
Circular geometry												
	FE P1 (nonstandard integ.)						FD 5p					
Dirichlet boundary conditions on innermost circle												
49 × 64	13	0.24	5.1e-02	–	9.6e-02	–	13	0.24	5.1e-02	–	9.6e-02	–
97 × 128	13	0.23	1.3e-02	2.01	2.4e-02	2.03	13	0.23	1.3e-02	2.01	2.4e-02	2.03
193 × 256	12	0.22	3.2e-03	2.01	6.0e-03	2.01	12	0.22	3.2e-03	2.00	6.0e-03	2.01
385 × 512	12	0.20	8.0e-04	2.00	1.5e-03	2.00	12	0.20	8.1e-04	2.00	1.5e-03	2.00
Discretization across the origin												
49 × 64	13	0.24	5.1e-02	–	9.6e-02	–	13	0.24	5.1e-02	–	9.5e-02	–
97 × 128	13	0.23	1.3e-02	2.01	2.4e-02	2.03	13	0.23	1.2e-02	2.04	2.6e-02	1.90
193 × 256	13	0.22	3.2e-03	2.01	5.9e-03	2.01	13	0.22	3.3e-03	1.93	3.0e-02	-0.24
385 × 512	13	0.22	8.0e-04	2.00	1.5e-03	2.00	13	0.22	1.7e-03	0.99	3.3e-02	-0.13
Deformed geometry												
	FE P1 (nonstandard integ.)						FD 9p					
Dirichlet boundary conditions on innermost circle												
49 × 64	47	0.67	7.1e-02	–	1.6e-01	–	47	0.67	7.0e-02	–	1.5e-01	–
97 × 128	45	0.66	1.8e-02	2.00	4.6e-02	1.80	45	0.66	1.8e-02	2.00	4.0e-02	1.87
193 × 256	43	0.65	4.5e-03	2.00	1.2e-02	1.94	43	0.65	4.4e-03	2.00	1.0e-02	1.96
385 × 512	41	0.63	1.1e-03	2.00	3.1e-03	1.99	41	0.63	1.1e-03	2.00	2.6e-03	1.99
Discretization across the origin												
49 × 64	47	0.67	7.0e-02	–	1.6e-01	–	47	0.67	6.9e-02	–	1.5e-01	–
97 × 128	46	0.67	1.8e-02	2.01	4.6e-02	1.81	46	0.67	1.7e-02	2.02	3.9e-02	1.90
193 × 256	45	0.66	4.4e-03	2.02	1.8e-02	1.33	45	0.66	4.5e-03	1.93	4.1e-02	−0.07
385 × 512	45	0.66	1.4e-03	1.64	2.0e-02	−0.14	45	0.66	2.4e-03	0.94	4.6e-02	−0.17

Multigrid *without extrapolation* based on *finite element* and *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 = 1e-2$ and *Dirichlet boundary conditions* on the *innermost circle* or *discretization across the origin*, respectively. Further notation as in Table 3

Table 5 Comparison of discretizations

$n_r \times n_\theta$	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.
Circular geometry												
	FE P1 (nonstandard integ.)						FD 5p					
Dirichlet boundary conditions on innermost circle												
49 × 64	13	0.23	5.1e-02	–	9.6e-02	–	13	0.23	5.1e-02	–	9.6e-02	–
97 × 128	13	0.23	1.3e-02	2.01	2.4e-02	2.03	13	0.23	1.3e-02	2.01	2.4e-02	2.03
193 × 256	13	0.22	3.2e-03	2.01	6.0e-03	2.01	13	0.22	3.2e-03	2.00	6.0e-03	2.01
385 × 512	13	0.22	8.0e-04	2.00	1.5e-03	2.00	13	0.22	8.0e-04	2.00	1.5e-03	2.00
Discretization across the origin												
49 × 64	13	0.23	5.1e-02	–	9.6e-02	–	13	0.23	5.1e-02	–	9.6e-02	–
97 × 128	13	0.23	1.3e-02	2.01	2.4e-02	2.03	13	0.23	1.3e-02	2.01	2.4e-02	2.02
193 × 256	13	0.22	3.2e-03	2.01	6.0e-03	2.01	13	0.22	3.2e-03	2.00	6.0e-03	2.01
385 × 512	13	0.22	8.0e-04	2.00	1.5e-03	2.00	13	0.22	8.0e-04	2.00	1.5e-03	2.00
Deformed geometry												
	FE P1 (nonstandard integ.)						FD 9p					
Dirichlet boundary conditions on innermost circle												
49 × 64	46	0.67	7.2e-02	–	1.6e-01	–	46	0.67	7.1e-02	–	1.5e-01	–
97 × 128	45	0.66	1.8e-02	2.00	4.7e-02	1.80	45	0.66	1.8e-02	2.00	4.1e-02	1.87
193 × 256	44	0.66	4.6e-03	2.00	1.2e-02	1.94	44	0.66	4.5e-03	2.00	1.1e-02	1.96
385 × 512	44	0.65	1.2e-03	2.00	3.1e-03	1.99	44	0.65	1.1e-03	2.00	2.6e-03	1.99
Discretization across the origin												
49 × 64	46	0.67	7.2e-02	–	1.6e-01	–	46	0.67	7.1e-02	–	1.5e-01	–
97 × 128	45	0.66	1.8e-02	2.00	4.7e-02	1.80	45	0.66	1.8e-02	2.00	4.1e-02	1.87
193 × 256	44	0.66	4.6e-03	2.00	1.2e-02	1.94	44	0.66	4.5e-03	2.00	1.1e-02	1.96
385 × 512	44	0.65	1.2e-03	2.00	3.1e-03	1.99	44	0.65	1.1e-03	2.00	2.6e-03	1.99

Multigrid *without extrapolation* based on *finite element* and *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 \in [1e-5, 1e-8]$ and *Dirichlet boundary conditions* on the *innermost circle* or *discretization across the origin*, respectively (Values grouped for $r_1 = 1e-5$ and $r_1 = 1e-8$ since the results are identical). Further notation as in Table 3

Table 6 Comparison of extrapolated discretizations

$n_r \times n_\theta$	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.
Circular geometry												
	FE P1 (nonstandard integ.)						FD 5p					
49×64	54	0.71	1.2e-02	–	6.5e-02	–	54	0.71	3.6e-03	–	1.6e-02	–
97×128	42	0.64	2.9e-03	2.08	3.6e-02	0.87	42	0.64	2.4e-04	3.95	1.5e-03	3.47
193×256	42	0.64	9.5e-04	1.62	1.8e-02	0.99	42	0.64	1.8e-05	3.77	1.8e-04	3.10
385×512	42	0.64	3.3e-04	1.52	9.1e-03	1.00	42	0.64	1.4e-06	3.66	2.2e-05	3.01
Deformed geometry												
	FE P1 (nonstandard integ.)						FD 9p					
49×64	150	0.90	1.4e-02	–	8.4e-02	–	150	0.90	7.6e-03	–	2.6e-02	–
97×128	140	0.88	2.1e-03	2.80	2.2e-02	1.98	140	0.88	5.6e-04	3.80	2.9e-03	3.15
193×256	136	0.87	4.6e-04	2.16	8.4e-03	1.37	135	0.87	4.2e-05	3.74	3.6e-04	3.02
385×512	134	0.87	1.5e-04	1.63	4.1e-03	1.03	133	0.87	3.2e-06	3.72	4.5e-05	3.00

Multigrid *with extrapolation* based on *finite element* and *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 = 0$. Further notation as in Table 3

and residual reduction factors are higher in the case of the deformed geometry. The number of iterations is still only between 41 and 47. The convergence of the iterative scheme is (almost) independent of the choice how to handle $r_1 > 0$. The number of iterations of our multigrid algorithm is independent of the discretization parameter.

The error convergence over the different levels of discretizations is unsatisfactory for the finite element discretization if $r_1 = 0$. For $r_1 = 1e-2$, our strategy to discretize across the origin also leads to unsatisfactory results; $r_1 = 1e-2$ might still be too large for this heuristic. For $r_1 \in [1e-5, 1e-8]$, we obtain identical results with this heuristic and Dirichlet boundary conditions on the innermost circle. We have optimal quadratic error convergence in l_2- as well as inf-norm.

5.4 Extrapolated Multigrid Based on Different Discretizations

We now consider the multigrid algorithm as in the previous section by only adding our extrapolation step between the two finest grids. Note that the analysis of [15] predicts an accuracy equivalent to P2-elements, and thus to reach $\mathcal{O}(h^3)$ in L_2 up from $\mathcal{O}(h^2)$. The meshes enjoy an approximate symmetry, which can lead to a cancellation of odd error terms. In fact, we will observe below that the approximation order reaches close to $\mathcal{O}(h^4)$ in some cases.

From Tables 6 and 7, we again see that the multigrid algorithm needs about twice as many iterations if $(0, 0)$ is used as explicit mesh node.

For the circular geometry and $r_1 > 0$, we need less than 40 iterations to reduce the residual by a factor of 10^8. The number of iterations and residual reduction factors are higher in the case of the deformed geometry but still only between 73 and 85. Again, the number of iterations is independent of the discretization parameter.

The slower convergence (compared to the previous section) is due to the influence of a strengthened Cauchy inequality. For more details, we refer to Remark 4.5 and [15].

Table 7 Comparison of extrapolated discretizations

$n_r \times n_\theta$	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.	Its	$\widehat{\rho}$	$\|err\|_{\ell_2}$	Ord.	$\|err\|_\infty$	Ord.
Circular geometry												
	FE P1 (nonstandard integ.)						FD 5p					
Dirichlet boundary conditions on innermost circle												
49 × 64	36	0.60	3.6e-03	–	1.6e-02	–	36	0.60	3.6e-03	–	1.6e-02	–
97 × 128	38	0.61	2.4e-04	3.95	1.5e-03	3.48	38	0.61	2.4e-04	3.94	1.5e-03	3.47
193 × 256	39	0.62	1.8e-05	3.77	1.8e-04	3.09	39	0.62	1.8e-05	3.76	1.8e-04	3.10
385 × 512	39	0.62	1.4e-06	3.66	2.2e-05	3.01	39	0.62	1.4e-06	3.66	2.2e-05	3.01
Discretization across the origin												
49 × 64	37	0.60	3.6e-03	–	1.6e-02	–	37	0.60	3.6e-03	–	1.6e-02	–
97 × 128	38	0.61	2.4e-04	3.95	1.5e-03	3.48	38	0.61	2.4e-04	3.94	1.5e-03	3.47
193 × 256	39	0.62	1.8e-05	3.77	1.8e-04	3.09	39	0.62	1.8e-05	3.76	1.8e-04	3.10
385 × 512	39	0.62	1.4e-06	3.66	2.2e-05	3.01	39	0.62	1.4e-06	3.65	2.2e-05	3.01
Deformed geometry												
	FE P1 (nonstandard integ.)						FD 9p					
Dirichlet boundary conditions on innermost circle												
49 × 64	75	0.78	7.9e-03	–	3.0e-02	–	73	0.77	7.6e-03	–	2.6e-02	–
97 × 128	80	0.79	6.1e-04	3.72	4.3e-03	2.83	77	0.79	5.6e-04	3.79	2.9e-03	3.15
193 × 256	83	0.80	4.8e-05	3.69	4.5e-04	3.26	78	0.79	4.2e-05	3.73	3.6e-04	3.02
385 × 512	84	0.80	3.7e-06	3.72	4.5e-05	3.32	78	0.79	3.2e-06	3.72	4.5e-05	3.00
Discretization across the origin												
49 × 64	75	0.78	7.9e-03	–	3.0e-02	–	73	0.77	7.6e-03	–	2.6e-02	–
97 × 128	80	0.79	6.1e-04	3.72	4.3e-03	2.84	77	0.79	5.6e-04	3.79	2.9e-03	3.15
193 × 256	83	0.80	4.8e-05	3.69	4.5e-04	3.25	78	0.79	4.2e-05	3.73	3.6e-04	3.02
385 × 512	84	0.80	3.7e-06	3.72	4.5e-05	3.35	78	0.79	3.2e-06	3.72	4.5e-05*1	3.00

Multigrid *with extrapolation* based on *finite element* and *finite difference* discretizations on *circular* and *deformed* geometry with $r_1 \in [1e-5, 1e-8]$ and *Dirichlet boundary conditions* on the *innermost circle* or *discretization across the origin*, respectively (Values grouped for $r_1 = 1e-5$ and $r_1 = 1e-8$ since the results are identical). Further notation as in Table 3. *1 : Values differ for $r_1 \in [1e-5, 1e-8]$ by $1e-6$

The error convergence over the different levels of discretizations is unsatisfactory for the finite element discretization if $r_1 = 0$.

For $r_1 > 0$, the convergence of the iterative scheme is independent of the choice on how to handle the innermost circle, if r_1 is sufficiently small. For $r_1 \in [1e-5, 1e-8]$, we obtain almost identic results with the heuristic of discretizing across the origin and Dirichlet boundary conditions on the innermost circle. We have an error convergence order between 3.5 and 4.0 in l_2- and a convergence order of about 3.0 in inf-norm.

6 Conclusion

We have presented a novel scalable geometric multigrid solver for a Poisson equation arising in gyrokinetic fusion plasma models. We have developed a new optimized radial-circle smoothing procedure to take into account the anisotropies of the underlying partial differential equation and of the mesh, particularly in the edge area of the separatrix in the Tokamak.

Furthermore, we have constructed an implicit extrapolation scheme that leads to third order error convergence in the inf-norm and shows an error convergence order between 3.5 and 4.0 in the l_2-norm. For simpler meshes and geometries as considered here, e.g., without deformation, artificial singularity, and anisotropic mesh-refinement, we expect up to convergence order four when the odd order terms in the error expansions vanish due to symmetry.

If using implicit extrapolation, the iteration counts are slightly larger but they still remain modest and independent of the mesh size, so that the solver is asymptopically optimal. This is necessary for algorithmic scalability. Further improvements may be possible based on more efficient smoothing procedures. The numerical results for our multigrid algorithm based on finite elements with nonstandard integration and the finite difference nine point stencil are almost identical. Our extrapolated finite difference stencil gives thus rise to a matrix-free implementation with low memory footprint and high precision. With the fast implicitly extrapolated multigrid method, we have constructed an algorithm to compute cost-effective high precision approximations of the gyrokinetic Poisson equation.

Author Contributions All authors developed the numerical algorithm, the implementation was done by M.J.K and C.K., all authors wrote and revised the script.

Funding Open Access funding enabled and organized by Projekt DEAL. The first two authors gratefully acknowledge the funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 824158.

Availability of Data and Material No particular data was used.

Declarations

Conflict of interest/Competing interests None.

Code Availability Code can be shared upon request.

References

1. Barros, S.R.M.: The Poisson equation on the unit disk: a multigrid solver using polar coordinates. Appl. Math. Comput. **25**(2), 123–135 (1988). https://doi.org/10.1016/0096-3003(88)90110-5
2. Bernert, K.: τ-extrapolation–theoretical foundation, numerical experiment, and application to Navier-Stokes equations. SIAM J. Sci. Comput. **18**(2), 460–478 (1997)
3. Börm, S., Hiptmair, R.: Analysis of tensor product multigrid. Numer. Algorithms **26**(3), 219–234 (2001). https://doi.org/10.1023/A:1016686408271
4. Bornemann, F.A., Deuflhard, P.: The cascadic multigrid method for elliptic problems. Numer. Math. **75**(2), 135–152 (1996)
5. Bouzat, N., Bressan, C., Grandgirard, V., Latu, G., Mehrenberger, M.: Targeting realistic geometry in Tokamak code Gysela. ESAIM Proc. **63**, 179–207 (2018). https://doi.org/10.1051/proc/201863179
6. Brandt, A., Livne, O.E.: Multigrid techniques: 1984 guide with applications to fluid dynamics, vol. 67. Society for Industrial and Applied Mathematics, Philadelphia (2011). https://doi.org/10.1137/1.9781611970753
7. Chen, C., Shi, Z.C., Hu, H.: On extrapolation cascadic multigrid method. J. Comput. Math. pp. 684–697 (2011)
8. Chuanmiao, C., Hongling, H., Ziqing, X., Chenliang, L.: L2-error of extrapolation cascadic multigrid (excmg). Acta Math. Sci. **29**(3), 539–551 (2009)
9. Dai, R., Lin, P., Zhang, J.: An efficient sixth-order solution for anisotropic Poisson equation with completed Richardson extrapolation and multiscale multigrid method. Comput. Math. Appl. **73**(8), 1865–1877 (2017). https://doi.org/10.1016/j.camwa.2017.02.020
10. Fundamenski, W.: Power exhaust in fusion plasmas. Cambridge University Press, Cambridge (2010). https://doi.org/10.1017/CBO9780511770609
11. Gmeiner, B., Rüde, U., Stengel, H., Waluga, C., Wohlmuth, B.: Towards textbook efficiency for parallel multigrid. Numer. Math. Theory Methods Appl. **8**(1), 22–46 (2015)
12. Görler, T., Tronko, N., Hornsby, W.A., Bottino, A., Kleiber, R., Norscini, C., Grandgirard, V., Jenko, F., Sonnendrücker, E.: Intercode comparison of gyrokinetic global electromagnetic modes. Phys. Plasmas **23**(7), 072503 (2016). https://doi.org/10.1063/1.4954915
13. Grandgirard, V., Abiteboul, J., Bigot, J., Cartier-Michaud, T., Crouseilles, N., Dif-Pradalier, G., Ehrlacher, C., Esteve, D., Garbet, X., Ghendrih, P., Latu, G., Mehrenberger, M., Norscini, C., Passeron, C., Rozar, F., Sarazin, Y., Sonnendrücker, E., Strugarek, A., Zarzoso, D.: A 5d gyrokinetic full-f global semi-lagrangian code for flux-driven ion turbulence simulations. Comput. Phys. Commun. **207**, 35–68 (2016). https://doi.org/10.1016/j.cpc.2016.05.007
14. Hackbusch, W.: Multigrid methods and applications. Springer, New York (1985). https://doi.org/10.1007/978-3-662-02427-0
15. Jung, M., Rüde, U.: Implicit extrapolation methods for multilevel finite element computations. SIAM J. Sci. Comput. **17**(1), 156–179 (1996). https://doi.org/10.1137/0917012
16. Jung, M., Rüde, U.: Implicit extrapolation methods for variable coefficient problems. SIAM J. Sci. Comput. **19**(4), 1109–1124 (1998). https://doi.org/10.1137/S1064827595293557
17. Kohl, N., Rüde, U.: Textbook efficiency: Massively parallel matrix-free multigrid for the Stokes system. SIAM Journal on Scientific Computing (2021). Accepted for publication. Preprint: arXiv:2010.13513
18. Kühn, M.J., Kruse, C., Rüde, U.: Energy-minimizing, symmetric discretizations for anisotropic meshes and energy functional extrapolation. SIAM J. Sci. Comput. **43**(4), A2448–A2473 (2021). https://doi.org/10.1137/21M1397520
19. Li, M., Zheng, Z., Pan, K.: An efficient extrapolation full multigrid method for elliptic problems in two and three dimensions. Int. J. Comput. Math. **98**(6), 1183–1198 (2021). https://doi.org/10.1080/00207160.2020.1812584
20. Liem, C.B., Shih, T.: Splitting extrapolation method, the: a new technique in numerical solution of multidimensional Prob, vol. 7. World Scientific (1995)
21. Lyness, J., Rüde, U.: Cubature of integrands containing derivatives. Numer. Math. **78**(3), 439–461 (1998). https://doi.org/10.1007/s002110050320
22. Marchuk, G., Shaidurov, V.: Difference methods and their extrapolations. Springer, New York (1983). https://doi.org/10.1007/978-1-4613-8224-9

23. Masthurah, N., Ni'mah, I., Muttaqien, F.H., Sadikin, R.: On comparison of multigrid cycles for Poisson solver in polar plane coordinates. In: Proceedings of the 2015 9th International Conference on Telecommunication Systems Services and Applications (TSSA), pp. 1–5 (2015). https://doi.org/10.1109/TSSA.2015.7440445
24. Pamela, S., Huijsmans, G., Thornton, A., Kirk, A., Smith, S., Hoelzl, M., Eich, T., Contributors, J., Team, M., Team, J., et al.: A wall-aligned grid generator for non-linear simulations of mhd instabilities in tokamak plasmas. Comput. Phys. Commun. **243**, 41–50 (2019). https://doi.org/10.1016/j.cpc.2019.05.007
25. Pan, K., He, D., Hu, H.: An extrapolation cascadic multigrid method combined with a fourth-order compact scheme for 3D Poisson equation. J. Sci. Comput. **70**(3), 1180–1203 (2017). https://doi.org/10.1007/s10915-016-0275-9
26. Qun, L., Tao, L.: The splitting extrapolation method for multidimensional problems. J. Comput. Math. pp. 45–51 (1983)
27. Rüde, U.: Extrapolation and related techniques for solving elliptic equations. Technical Report TUM-I9135, Institut für Informatik, TU München (1991). http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.9471
28. Rüde, U.: The hierarchical basis extrapolation method. SIAM J. Sci. Stat. Comput. **13**(1), 307–318 (1992). https://doi.org/10.1137/0913016
29. Rüde, U.: Multilevel, extrapolation, and sparse grid methods. In: Multigrid Methods IV, pp. 281–294. Springer (1994)
30. Rüde, U., Zhou, A.: Multi-parameter extrapolation methods for boundary integral equations. Adv. Comput. Math. **9**(1), 173–190 (1998)
31. Shaidurov, V.V.: Some estimates of the rate of convergence for the cascadic conjugate-gradient method. Comput. Math. Appl. **31**(4–5), 161–171 (1996)
32. Sonnendrücker, E.: (2019). Private communication
33. Stüben, K., Trottenberg, U.: Multigrid methods: Fundamental algorithms, model problem analysis and applications. In: Multigrid methods, pp. 1–176. Springer (1982). https://doi.org/10.1007/BFb0069928
34. Tikhovskaya, S.: Solving a singularly perturbed elliptic problem by a cascadic multigrid algorithm with Richardson extrapolation. In: International Conference on Finite Difference Methods, pp. 533–541. Springer (2018). https://doi.org/10.1007/978-3-030-11539-5_62
35. Trottenberg, U., Oosterlee, C.W., Schüller, A.: Multigrid. Academic Press, London San Diego (2001)
36. Wang, Y., Zhang, J.: Sixth order compact scheme combined with multigrid method and extrapolation technique for 2d poisson equation. J. Comput. Phys. **228**(1), 137–146 (2009)
37. Wesson, J., Campbell, D.J.: Tokamaks, vol. 149. Oxford University Press, Oxford (2011)
38. Zoni, E.: Theoretical and numerical studies of gyrokinetic models for shaped tokamak plasmas. Ph.D. thesis, Technische Universität München, München (2019)
39. Zoni, E., Güçlü, Y.: Solving hyperbolic-elliptic problems on singular mapped disk-like domains with the method of characteristics and spline finite elements. J. Comput. Phys. **398**, 108889 (2019). https://doi.org/10.1016/j.jcp.2019.108889