Should Multilevel Methods for Discontinuous Galerkin Discretizations Use Discontinuous Interpolation Operators?

Martin J. Gander and José Pablo Lucero Lorca

1 Discontinuous Interpolation for a Model Problem

Interpolation operators are very important for the construction of a multigrid method. Since multigrid's inception by Fedorenko [7], interpolation was identified as key, deserving an entire appendix in Brandt's seminal work [5]: *'[...] even a small and smooth residual function may produce large high-frequency residuals, and significant amount of computational work will be required to smooth them out.'*

For discontinuous Galerkin (DG) discretizations [2], the problem of choosing an interpolation becomes particularly interesting. A good interpolation operator will not produce undesirable high frequency components in the residual. In an inherited (Galerkin) coarse operator, the choice of restriction and prolongation operators defines the coarse space itself, and then convergence of multigrid algorithms with classical restriction and interpolation operators for DG discretizations of elliptic problems cannot be independent of the number of levels [1]. In 1D, the reason for this was identified in [9, §4.3]): the DG penalization is doubled at each coarsening, causing the coarse problem to become successively stiffer.

A simple classical interpolation operator is linear interpolation: in 1D one takes the average from two adjacent points in the coarser grid and sets the two DG degrees of freedom at the midpoint belonging to the fine mesh to this same value, therefore imposing continuity at that point and discontinuity at coarse grid points. But why should continuity be imposed on the DG interpolated solution on the fine mesh? Can solver performance be improved with a discontinuous interpolation operator?

Convergence of two-level methods for DG discretizations has been analyzed for continuous interpolation operators using classical analysis, see [8, 3] and references therein, and also Fourier analysis [10, 11, 12, 9]. We use Fourier analysis here

Martin J. Gander
University of Geneva, e-mail: martin.gander@unige.ch

José Pablo Lucero Lorca
University of Colorado at Boulder, e-mail: pablo.lucero@colorado.edu

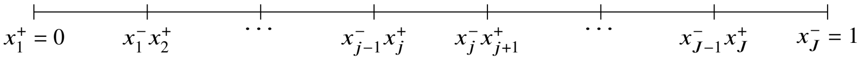

Fig. 1: Mesh for our DG discretization of the Poisson equation.

to investigate the influence of a discontinuous interpolation operator on the two level solver performance. We consider a symmetric interior penalty discontinuous Galerkin (SIPG) finite element discretization of the Poisson equation as in [2],

$$a_h(u,v) := \int_{\mathbb{T}} \nabla u \cdot \nabla v dx + \int_{\mathbb{F}} \left([[u]] \left\{ \frac{\partial v}{\partial n} \right\} + \left\{ \frac{\partial u}{\partial n} \right\} [[v]] \right) ds + \int_{\mathbb{F}} \delta \, [[u]] \, [[v]] \, ds, \quad (1)$$

on a 1D mesh as shown in Fig. 1. The resulting linear system is (for details see [9])

$$A\boldsymbol{u} = \frac{1}{h^2} \begin{pmatrix} \ddots & \ddots & -\frac{1}{2} & & & \\ \ddots & \delta_0 & 1-\delta_0 & -\frac{1}{2} & & \\ -\frac{1}{2} & 1-\delta_0 & \delta_0 & & -\frac{1}{2} & \\ & -\frac{1}{2} & & \delta_0 & 1-\delta_0 & -\frac{1}{2} \\ & & -\frac{1}{2} & 1-\delta_0 & \delta_0 & \ddots \\ & & & -\frac{1}{2} & \ddots & \ddots \end{pmatrix} \begin{pmatrix} \vdots \\ u_{j-1}^+ \\ u_j^- \\ u_j^+ \\ u_{j+1}^- \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ f_{j-1}^+ \\ f_j^- \\ f_j^+ \\ f_{j+1}^- \\ \vdots \end{pmatrix} =: \boldsymbol{f}, \quad (2)$$

where the top and bottom blocks will be determined by the boundary conditions, h is the mesh size, $\delta_0 \in \mathbb{R}$ is the DG penalization parameter, $\boldsymbol{f} = (\ldots, f_{j-1}^+, f_j^-, f_j^+, f_{j+1}^-, \ldots) \in \mathbb{R}^{2J}$ is the source vector, analogous to the solution $\boldsymbol{u}$.

The two-level preconditioner M^{-1} we study consists of a cell-wise nonoverlapping Schwarz (a *cell* block-Jacobi) smoother D_c^{-1}, since the discretization leads to a block matrix (see [8, 6])[1], and a new discontinuous interpolation operator P with discontinuity parameter c, i.e.

$$D_c^{-1}\boldsymbol{u} := h^2 \begin{pmatrix} \ddots & & & \\ & \delta_0 & & \\ & & \delta_0 & \\ & & & \ddots \end{pmatrix}^{-1} \begin{pmatrix} \vdots \\ u_j^+ \\ u_j^- \\ \vdots \end{pmatrix}, \quad P := \begin{pmatrix} 1 & & & \\ c & 1-c & & \\ 1-c & c & & \\ & 1 & & \\ & & \ddots & \\ & & \ddots & \ddots \\ & & \ddots & \ddots \\ & & & \ddots \end{pmatrix}, \quad (3)$$

where $c = \frac{1}{2}$ gives a continuous interpolation on the nodes not present in the coarse mesh, and discontinuous elsewhere. The restriction operator is $R := \frac{1}{2}P^{\intercal}$, and we use $A_0 := RAP$. The action of our two-level preconditioner M^{-1}, with one presmoothing step and a relaxation parameter α, acting on a residual $\boldsymbol{g}$, is given by

[1] In 1D this is simply a Jacobi smoother, which is not the case in higher dimensions.

1. compute $\boldsymbol{x} := \alpha D_c^{-1}\boldsymbol{g}$,
2. compute $\boldsymbol{y} := \boldsymbol{x} + PA_0^{-1}R(\boldsymbol{g} - A\boldsymbol{x})$,
3. obtain $M^{-1}\boldsymbol{g} = \boldsymbol{y}$.

2 Study of optimal parameters by Local Fourier Analysis

In [9] we described in detail, for classical interpolation, how Local Fourier Analysis (LFA) can be used to block diagonalize all the matrices involved in the definition of M^{-1} by using unitary transformations. The same approach still works with our new discontinuous interpolation operator, and we thus use the same definitions and notation for the block-diagonalization matrices Q, Q_l, Q_r, Q_0, Q_{l0} and Q_{r0} from [9], working directly with matrices instead of stencils in order to make the important LFA more accessible to our linear algebra community. We extract a submatrix $\widetilde{A}$ containing the degrees of freedom of two adjacent cells from the SIPG operator defined in (2),

$$\widetilde{A} = \frac{1}{h^2}\begin{pmatrix} -\frac{1}{2} & 1-\delta_0 & \delta_0 & 0 & -\frac{1}{2} & & & \\ & -\frac{1}{2} & 0 & \delta_0 & 1-\delta_0 & -\frac{1}{2} & & \\ & & -\frac{1}{2} & 1-\delta_0 & \delta_0 & 0 & -\frac{1}{2} & \\ & & & -\frac{1}{2} & 0 & \delta_0 & 1-\delta_0 & -\frac{1}{2} \end{pmatrix},$$

which we can block-diagonalize, $\widehat{A} = Q_l\widetilde{A}Q_r$, to obtain

$$\widehat{A} = \frac{1}{h^2}\begin{pmatrix} \delta_0 + \cos(2\pi(k-J/2)h) & 1-\delta_0 & & \\ 1-\delta_0 & \delta_0 + \cos(2\pi(k-J/2)h) & & \\ & & \delta_0 - \cos(2\pi kh) & 1-\delta_0 \\ & & 1-\delta_0 & \delta_0 - \cos(2\pi kh) \end{pmatrix}.$$

The same mechanism can be applied to the smoother, $\widehat{D}_c = Q_l\widetilde{D}_cQ_r = \frac{\delta_0}{h^2}I$, where I is the 4×4 identity matrix, and also to the restriction and prolongation operators, $\widehat{R} = \frac{1}{2}Q_{l0}\widetilde{R}Q_r$ with

$$\widehat{R} = \frac{1}{\sqrt{2}}\begin{pmatrix} 1+(c-1)e^{\frac{2i\pi k}{J}} & -ce^{\frac{2i\pi k}{J}} & (-1)^j\left(1-(c-1)e^{\frac{2i\pi k}{J}}\right) & (-1)^jce^{\frac{2i\pi k}{J}} \\ (-1)^jce^{-\frac{2i\pi k}{J}} & (-1)^j\left(1+(c-1)e^{-\frac{2i\pi k}{J}}\right) & ce^{-\frac{2i\pi k}{J}} & 1-(c-1)e^{-\frac{2i\pi k}{J}} \end{pmatrix},$$

and $P = 2R^\intercal$, $\widehat{P} = Q_l\widetilde{P}Q_{r0} = 2\widehat{R}^*$. Finally, for the coarse operator, we obtain $Q_0^*A_0Q_0 = Q_0^*RAPQ_0 = Q_0^*RQQ^*AQQ^*PQ_0$, and thus $\widehat{A}_0 = \widehat{R}\widehat{A}\widehat{P}$ with

$$\widehat{A}_0 = \frac{1}{H^2}\begin{pmatrix} \frac{1}{2}\left(c\left(4(c-1)\delta_0 - 2c + 3\right) + (c-1)\cos\left(\frac{4\pi k}{J}\right) + 2\delta_0 - 1\right) & \frac{1}{2}(-1)^j\left(-(2c-1)\left(c\left(2\delta_0 - 1\right) - \delta_0 + 1\right)e^{\frac{4i\pi k}{J}} - c - \delta_0 + 1\right) \\ \frac{1}{2}(-1)^j\left(-(2c-1)\left(c\left(2\delta_0 - 1\right) - \delta_0 + 1\right)e^{-\frac{4i\pi k}{J}} - c - \delta_0 + 1\right) & \frac{1}{2}\left(c\left(4(c-1)\delta_0 - 2c + 3\right) + (c-1)\cos\left(\frac{4\pi k}{J}\right) + 2\delta_0 - 1\right) \end{pmatrix},$$

where $H = 2h$. We notice that the coarse operator is different for j even and j odd; however, the matrices obtained for both cases are similar, with similarity matrix $(-1)^j I$ where I is the identity matrix, and therefore have the same spectrum. In what follows we assume j to be even, without loss of generality. This means that we will be studying a node that is present in both the coarse and fine meshes.

The error reduction capabilities of our two level preconditioner M^{-1} are given by the spectrum of the stationary iteration operator

$$E = (I - PA_0^{-1}RA)(I - \alpha D_c^{-1}A),$$

and as in [9], the 4-by-4 block Fourier-transformed operator from LFA,

$$\widehat{E}(k) = (I - \widehat{P}(k)\widehat{A}_0^{-1}(k)\widehat{R}(k)\widehat{A}(k))(I - \alpha\widehat{D}_c^{-1}(k)\widehat{A}(k)),$$

has the same spectrum. Thus, we focus on studying the spectral radius $\rho(\widehat{E}(k))$ in order to find the optimal choices for the relaxation parameter α, the penalty parameter δ_0 and the discontinuity parameter c. The non zero eigenvalues of $\widehat{E}(k)$ are of the form $\lambda_\pm := c_1 \pm \sqrt{\frac{c_2}{c_3}}$, with

$$c_1 = \left\{\begin{array}{l} \Big(\Big\{ - \alpha\left(3c^2\delta_0\left(4\delta_0 - 3\right) + c\left(-12\delta_0^2 + 9\delta_0 + 1\right) + 4\delta_0^2 - 2\delta_0 - 1\right) \\ + \delta_0\left(c^2\left(8\delta_0^2 - 4\delta_0 - 1\right) + c\left(-8\delta_0^2 + 4\delta_0 + 2\right) + 2\delta_0^2 - 1\right) \\ +(1-c)\left(\alpha + \alpha c\left(\delta_0 - 2\right) + (c-1)\delta_0\right)\cos\left(\dfrac{4\pi k}{J}\right)\Big\} \Big/ \\ \left(2\delta_0^2 - 1 + \delta_0 c^2\left(8\delta_0^2 - 4\delta_0 - 1\right) + \delta_0 c\left(-8\delta_0^2 + 4\delta_0 + 2\right) - \delta_0(c-1)^2\cos\left(\dfrac{4\pi k}{J}\right)\right)\Big), \end{array}\right.$$

$$c_2 = \left\{\begin{array}{l} 2\alpha^2\Big(16(c-1)^2c^2\delta_0^4 - 2(c-1)^2\left(4c^2 + c + 2\right)\delta_0 - 8(c-1)c(3(c-1)c - 1)\delta_0^3 \\ + \left(c(17c + 8)(c-1)^2 + 2\right)\delta_0^2 + 2(c-1)^2((c-1)c + 1)\Big) \\ + 4\alpha^2\left(4(c-1)c\delta_0^2 - 3(c-1)c\delta_0 + c + \delta_0 - 1\right)\left(c\left(3(c-1)\delta_0 - 2c + 3\right) + \delta_0 - 1\right)\cos\left(\dfrac{4\pi k}{J}\right) \\ + 2\alpha^2(c-1)^2c\left(c\left(\left(\delta_0 - 4\right)\delta_0 + 2\right) + 2\left(\delta_0 - 1\right)\right)\cos^2\left(\dfrac{4\pi k}{J}\right), \end{array}\right.$$

$$c_3 = \left\{\begin{array}{l} \delta_0^2\left(4c(c-1)\delta_0 - 2(1-2c)^2\delta_0^2 + (c-1)^2\right)^2 \\ + 2\delta_0^2\left(-2\left(2c^2 - 3c + 1\right)^2\delta_0^2 + 4c(c-1)^3\delta_0 + (c-1)^4\right)\cos\left(\dfrac{4\pi k}{J}\right) \\ + (c-1)^4\delta_0^2\cos^2\left(\dfrac{4\pi k}{J}\right). \end{array}\right.$$

A first approach to optimize would be to minimize the spectral radius for all frequency parameters k, but if we can find a combination of the parameters (α, δ_0, c) such that the eigenvalues of the error operator do not depend on the frequency parameter

k, then the spectrum of the iteration operator, and therefore the preconditioned system becomes perfectly clustered , i.e. only a few eigenvalues repeat many times, regardless of the size of the problem. The solver then becomes mesh independent, and the preconditioner very attractive for a Krylov method that will converge in a finite number of steps.

For these equations not to depend on k, they must be independent of $\cos\left(\frac{4\pi k}{J}\right)$, and to achieve this, we impose three conditions on the coefficients accompanying the cosine, and we deduce a combination of the parameters (α, δ_0, c) which we verify *a posteriori* fall into the allowed range of values for each parameter. Our conditions are:

1. Set the coefficient accompanying the cosine in the numerator of c_1 to zero.
2. Since the denominator of c_1 also contains the cosine, set the rest of the numerator of c_1 to zero in order to get rid of c_1 entirely. Note that this requirement immediately implies an equioscillating spectrum (i.e. the maximum and the minimum have equal absolute value), which often is characterizing the solution minimizing the spectral radius, see e.g. [9].
3. c_2 and c_3 are second order polynomials in the cosine variable, if we want the quotient to be non zero and independent of the cosine, we need for the polynomials to simplify and for that, they must differ only by a multiplying factor independent of the cosine. We then equate the quotient of the quadratic terms with the quotient of the linear terms and verify *a posteriori* that c_2/c_3 becomes indeed independent of the cosine.

These three conditions lead to the nonlinear system of equations

$$\begin{cases} \alpha + \alpha c\,(\delta_0 - 2) + (c-1)\delta_0 = 0, \\[2ex] \alpha\left(3c^2\delta_0\,(4\delta_0 - 3) + c\left(-12\delta_0^2 + 9\delta_0 + 1\right) + 4\delta_0^2 - 2\delta_0 - 1\right) = \\ \delta_0\left(c^2\left(8\delta_0^2 - 4\delta_0 - 1\right) + c\left(-8\delta_0^2 + 4\delta_0 + 2\right) + 2\delta_0^2 - 1\right), \\[2ex] \dfrac{2\alpha^2(c-1)^2 c\,(c\,((\delta_0 - 4)\,\delta_0 + 2) + 2\,(\delta_0 - 1))}{(c-1)^4\delta_0^2} = \\ \dfrac{4\alpha^2\left(4(c-1)c\delta_0^2 - 3(c-1)c\delta_0 + c + \delta_0 - 1\right)\left(c\,(3(c-1)\delta_0 - 2c + 3) + \delta_0 - 1\right)}{2\delta_0^2\left(-2\left(2c^2 - 3c + 1\right)^2\delta_0^2 + 4c(c-1)^3\delta_0 + (c-1)^4\right)}. \end{cases}$$

This system of equations can be solved either numerically or symbolically. After a significant effort, the following values solve our nonlinear system:

$$\begin{aligned} c =& \text{Root of } 3 - 8\tilde{c} + 8\tilde{c}^2 - 8\tilde{c}^3 + 4\tilde{c}^4 \text{ such that } \tilde{c} \in \mathbb{R} \text{ and } 0 < \tilde{c} < 1, \\ \delta_0 =& \text{Root of } -1 - 4\tilde{\delta_0} + 24\tilde{\delta_0}^2 - 32\tilde{\delta_0}^3 + 12\tilde{\delta_0}^4 \text{ such that } \tilde{\delta_0} \in \mathbb{R} \text{ and } 1 < \tilde{\delta_0}, \text{ and} \\ \alpha =& \text{Root of } -1 - 40\tilde{\alpha} + 214\tilde{\alpha}^2 - 352\tilde{\alpha}^3 + 183\tilde{\alpha}^4 \text{ such that } \tilde{\alpha} \in \mathbb{R} \text{ and } 0 < \tilde{\alpha} < 1. \end{aligned}$$

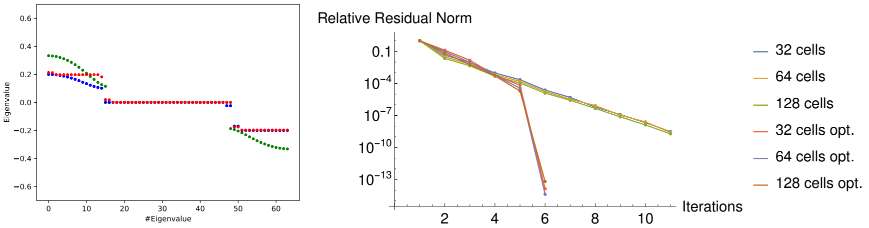

Fig. 2: Solving $-\Delta u = 1$ in 1D with Dirichlet boundary conditions. Left: eigenvalues of the error operator E, for a 32-cell mesh. Top curve at 0: optimizing α for $\delta_0 = 2$ (classical choice). Bottom curve at 0: optimizing α and δ_0. Middle curve at 0: optimizing α, δ_0 and c. Right: GMRES iterations for classical interpolation $c = 0.5$, with $\delta_0 = 2$ and $\alpha = 8/9$, and for the optimized clustering choice, leading to finite step convergence.

The corresponding numerical values are approximately

$$c \approx 0.564604, \quad \delta_0 \approx 1.516980, \quad \alpha \approx 0.908154,$$

and we see that indeed the interpolation should be discontinuous! We have found a combination of parameters that perfectly clusters the eigenvalues of the iteration operator of our two level method, and therefore also the spectrum of the preconditioned operator. Such clustering is not very often possible in preconditioners, a few exceptions are the HSS preconditioner in [4], and some block preconditioners, see e.g. [13]. Furthermore, the spectrum is equioscillating, which often characterizes the solution minimizing the spectral radius of the iteration operator.

3 Numerical Results

We show in Fig. 2 on the left the eigenvalues of the iteration operator for a 32-cell mesh in 1D with Dirichlet boundary conditions, for continuous interpolation and $\delta_0 = 2$ optimizing only α, optimizing both α and δ_0, and the optimized clustering choice. We clearly see the clustering of the eigenvalues, including some extra clusters due to the Dirichlet boundary conditions. We also note that the spectrum is nearly equioscillating due to condition (1) and (2), which delivers visibly an optimal choice in the sense of minimizing the spectral radius of the error operator. With periodic boundary conditions, the spectral radius for the optimal choice of α, δ_0 and c is 0.19732, while only optimizing α and δ_0 it is 0.2. The eigenvalues due to the Dirichlet boundary conditions are slightly larger than 0.2, but tests with periodic boundary conditions confirm that then these larger eigenvalues are not present. Refining the mesh conserves the shape of the spectrum shown in Fig. 2 on the left, but with more eigenvalues in each cluster, except for the clusters related to the Dirichlet boundary conditions. Note also that since the error operator is equioscillating around zero, the

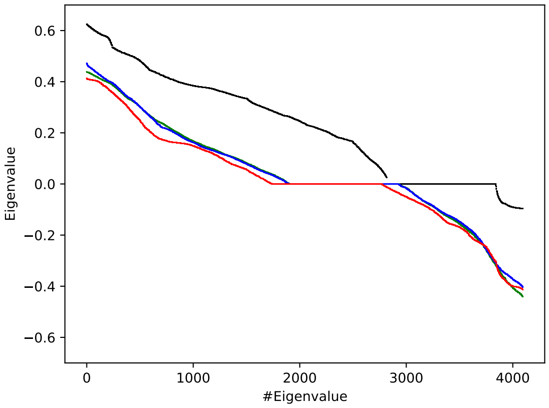

Fig. 3: Spectrum of the iteration operator for a 32-by-32 square 2D mesh. First curve at 0 from the top: optimizing α for $\delta_0 = 2$ (classical choice) in 1D. Third curve at 0 from the top: optimizing α and δ_0 in 1D. Second curve at 0 from the top: optimizing α, δ_0 and c in 1D. Fourth curve at 0 from the top: numerically optimizing α, δ_0 and c in 2D.

spectrum of the preconditioned system is equioscillating around one, and since the spectral radius is less than one, the preconditioned system has a positive spectrum and is thus invertible.

In Fig. 2 on the right we show the GMRES iterations needed to reduce the residuals by 10^{-8} for different parameter choices and the clustering choice, for different mesh refinements. We observe that the GMRES solver becomes exact after six iterations for the clustering choice.

We next perform tests in two dimensions using an interpolation operator with a stencil that is simply a tensor product of the 1D stencil $\begin{pmatrix} 1 & 0 \\ c & 1-c \\ 1-c & c \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ c & 1-c \\ 1-c & c \\ 0 & 1 \end{pmatrix}$, where $\otimes$ stands for the Kronecker product. This is very common in DG methods where even the cell block-Jacobi matrix can be expressed as a Kronecker sum for fast inversion. We show in Fig. 3 the spectrum for different optimizations in two dimensions. We observe that the clustering is not present, however as shown in detail in [9] for classical interpolation, the optimal choice from the 1D analysis is also here very close to the numerically calculated optimum in 2D.

4 Conclusion

We showed for a one dimensional discontinuous Galerkin model problem that the optimization of a two grid method leads to a discontinuous interpolation operator, and its performance is superior to using a continuous interpolation operator. The discontinuous interpolation operator allowed us also to cluster the spectrum for our model problem, and thus a Krylov method with this preconditioner becomes a direct solver, converging in the number of iterations corresponding to the number of clusters in exact arithmetic. We showed numerically that this is indeed the case, and that when using the one dimensional optimized parameters in higher spatial dimensions, we still get a spectrum close to the numerically best possible one, even though the spectrum is not clustered any more. We currently investigate if there exist discontinous interpolation operators in 2D that cluster the spectrum, and what their influence is on the Galerkin coarse operator obtained.

References

1. Paola F. Antonietti, Marco Sarti, and Marco Verani. Multigrid algorithms for hp-discontinuous Galerkin discretizations of elliptic problems. *SIAM J. on Numer. Anal.*, 53(1):598–618, 2015.
2. Douglas N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.
3. B. Ayuso de Dios and Zikatanov L. Uniformly convergent iterative methods for discontinuous Galerkin discretizations. *J. Sci. Comput.*, 40:4–36, 2009.
4. Michele Benzi, Martin J. Gander, and Gene H. Golub. Optimization of the hermitian and skew-hermitian splitting iteration for saddle-point problems. *BIT Numerical Mathematics*, 43(5):881–900, 2003.
5. Achi Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.
6. Maksymilian Dryja and Piotr Krzyżanowski. A massively parallel nonoverlapping additive Schwarz method for discontinuous Galerkin discretization of elliptic problems. *Numerische Mathematik*, 132(2):347–367, February 2016.
7. R. P. Fedorenko. The speed of convergence of one iterative process. *Zh. Vychisl. Mat. Mat. Fiz.*, 4:559–564, 1964.
8. X. Feng and O. Karakashian. Two-level non-overlapping Schwarz methods for a discontinuous Galerkin method. *SIAM J. Numer. Anal.*, 39(4):1343–1365, 2001.
9. Martin Jakob Gander and José Pablo Lucero Lorca. Optimization of two-level methods for DG discretizations of reaction-diffusion equations, 2020.
10. P. Hemker. Fourier analysis of grid functions, prolongations and restrictions. *Report NW 98, CWI, Amsterdam*, 1980.
11. P. Hemker, W. Hoffmann, and M. van Raalte. Two-level fourier analysis of a multigrid approach for discontinuous Galerkin discretization. *SIAM J. on Sci. Comp.*, 3(25):1018–1041, 2003.
12. P. W. Hemker, W. Hoffmann, and M. H. van Raalte. Fourier two-level analysis for discontinuous Galerkin discretization with linear elements. *Numerical Linear Algebra with Applications*, 5 – 6(11):473–491, 2004.
13. David Silvester and Andrew Wathen. Fast iterative solution of stabilised Stokes systems part II: Using general block preconditioners. *SIAM J. on Num. Anal.*, 31(5):1352–1367, 1994.