Exascale **Q**uantification of **U**ncertainties for
Technology and Science Simulation

D3.1 Report on nonlinear domain decomposition preconditioners and release of the solvers

Document information table

Contract number:	800898
Project acronym:	ExaQUte
Project Coordinator:	CIMNE
Document Responsible Partner:	UPC
Deliverable Type:	Report, Other
Dissemination Level:	Public
Related WP & Task:	WP3, Task 3.2
Status:	Final Version

Authoring

Prepared by:				
Authors	Partner	Modified Page/Sections	Version	Comments
Santiago Badia	CIMNE			
Alberto F. Martín	CIMNE			
Contributors				
Javier Principe	UPC			

Change Log

Versions	Modified Page/Sections	Comments

Approval

Approved by:				
	Name	Partner	Date	OK
Task leader	Santiago Badia	CIMNE	29/01/2019	Yes
WP leader	A. F. Martín	CIMNE	29/01/2019	Yes
Coordinator	Javier Principe	UPC	30/01/2019	Yes

Table of contents

List of Figures

Nomenclature / Acronym list

Acronym	Meaning
ExaQUte	EXAscale Quantification of Uncertainties for Technology and Science Simulation
BDDC	Balancing Domain Decomposition by Constraints
HPC	High performance computing
PDE	Partial differential equation
MN-IV	Marenostrum-IV supercomputer
FE	finite element
DOF	degree of freedom
PDE	partial differential equation
MPI	Message Passing Interface
BSC	Barcelona Supercomputing Center
HTS	High Temperature Superconductors
PB	physics-based
DD	domain decomposition
PB-BDDC	physics-based BDDC
rPB-BDDC	relaxed PB-BDDC

1 Introduction

In this deliverable we provide the details related to the design, implementation, and scalability analysis of two nonlinear domain decomposition (DD) methods that have been released in the frame of the `FEMPAR` project [4]. First, we introduce the linear solvers being used in Sect. 2. Next, we will consider nonlinear domain decomposition solvers in Sect. 3. Two different nonlinear solvers are proposed. Sect. 4 is devoted to the numerical analysis of these algorithms. After a thorough analysis for the p-Laplacian problem, the method that provides better results will be used for the simulation of complex and highly nonlinear electromagnetics problems.

2 The BDDC (linear) preconditioner

Balancing Domain Decomposition by Constraints (BDDC) preconditioners [10] belong to the family of non-overlapping DD methods [26]. They can be understood as an evolution of the earlier Balancing DD method [16]. These methods rely on the definition of a finite element (FE) space with relaxed inter-element continuity, which is defined by choosing some quantities to be continuous across subdomain interfaces, i.e., the *coarse* or *primal* degrees of freedom (DOFs). Then, the continuity of the solution at the interface between subdomains is restored with an averaging operator. The method has two properties that make it an outstanding candidate for extreme scale computing, namely it allows for aggressive coarsening and computations among the different levels can be performed in parallel. Outstanding scalability results have been achieved by an implementation in the scientific computing software `FEMPAR` [4], which exploits these two properties in up to almost half a million cores and two million subdomains (MPI tasks) [1]. Another work showing excellent scalability properties up to two hundred thousand cores is [27], which is implemented in the software project PETSc [6].

Modern BDDC methods [21] propose coarse space enrichment techniques that adapt to the variation of coefficients of the problem [9, 15, 17, 23–25], where coarse DOFs are adaptively selected by solving generalized eigenvalue problems. This approach is backed up by rigorous mathematical theory and has been numerically shown to be robust for general heterogeneous problems. On the other hand, several different scalings have been proposed for the averaging operator in the literature to improve the lack of robustness of the *cardinality* (i.e., arithmetic mean) scaling for coefficient jumps. The *stiffness*[1] scaling takes more information into account but can lead to poor preconditioner performance with mildly varying coefficients [22]. The most robust approach up-to-date is the *deluxe* scaling, first introduced in [11] for 3D problems in curl-conforming spaces. It is based on the solution of local auxiliary Dirichlet problems to compute efficient averaging operators [12, 19, 21, 27–29], involving dense matrices per subdomain vertex/edge/face. However, to solve eigenvalue and auxiliary problems is expensive and extra implementation effort is required as coarse spaces in DD methods are not naturally formulated as eigenfunctions.

One of the lines of research of the authors of this report is to construct robust BDDC preconditioners for problems in curl-conforming spaces that keep the simplicity of the standard BDDC method, i.e., to avoid the spectral solvers of adaptive versions, whereas keeping robustness and low computational cost. In order to do so, we follow the idea

[1] weighted averages with the diagonal entries of the operator for every DOF

of the physics-based BDDC (PB-BDDC) preconditioner, presented in [5] for problems in grad-conforming spaces. Based on the fact that BDDC methods (and DD methods in general) are robust with regard to jumps in the material coefficients when these jumps are aligned with the partition [13, 26], one can use a physics-based (PB)-partition obtained by aggregating elements of the same (or similar) coefficient value. However, using this type of partition can lead to a poor load balancing among subdomains and large interfaces. To overcome this situation, the PB-BDDC respects the original partition (well-balanced) but considers a sub-partition of every subdomain based on the physical coefficients, leading to a partition of the objects into sub-objects defined according to the variation of the coefficients. Consequently, the method is also based on an enrichment of the coarse space but with the great advantage of not requiring to solve eigenvalue or auxiliary problems, i.e., the simplicity of the original BDDC preconditioner is maintained. The PB-BDDC preconditioner turned out to be one order of magnitude faster than the BDDC method with deluxe scaling in [27] for linear elasticity and thermal conductivity problems with high contrast. A relaxed definition of the PB subdomains, where we only require that the maximal contrast of the two physical coefficients is smaller than a predefined thresholds, allows one to extend the range of applicability of the preconditioner to truly heterogeneous materials. The extension of PB-BDDC to electromagnetics applications has been recently developed in [2].

The interested reader is referred to [1, 2, 5] for a detailed exposition of BDDC, PB-BDDC, and relaxed PB-BDDC (rPB-BDDC) methods, resp., which is not the aim of this deliverable.

3 Nonlinear domain decomposition solvers

For nonlinear problems, there are additional mechanisms to boost the scalability of linear solvers, oriented to reduce synchronization at the expense of additional local computations. The Newton method and its inexact versions are the most common linearization procedures. Every nonlinear iteration consists in a linear solver with the (possibly inexact) Jacobian matrix, where all the DD-Krylov solvers discussed above can be used; this approach is usually denoted as Newton-Krylov (NK) DD. However, the first linearize/next distribute approach must be re-thought to increase work load between checkpoints (synchronizations), e.g., inner products at every Krylov solver iteration. One approach is to consider the mesh partition first and linearization next as originally proposed in the ASPIN method in the context of additive Schwarz preconditioners [8]. Nonlinear extensions of two-level BNN, BDDC and FETI-DP methods have also been considered in [14]. The composition of nonlinear solvers of different types have been proved to be essential to reduce time-to-solution in nonlinear problems [7], e.g., combining (quasi-)Newton, Anderson acceleration, and line search schemes.

In this work, we consider first a standard Newton-Krylov BDDC solver, which combines an outer Newton nonlinear solver (possibly with, e.g., cubic backtracking) and a BDDC solver for the linearized problem at every nonlinear iteration. Next, in order to increase parallelism, the nonlinear BDDC method in [14] will be considered, which is just a combination of a Newton nonlinear solver, a nonlinear interior correction, i.e., solving subdomain-wise nonlinear problems in parallel, which will act as linearization point, and finally one of the BDDC linear systems commented above.

4 Numerical experiments

4.1 Experimental environment

The numerical experiments are run at the Marenostrum-IV (MN-IV) supercomputer, hosted by Barcelona Supercomputing Center (BSC). This petascale machine is equipped with 3,456 compute nodes interconnected together with the Intel OPA HPC network. Each node has 2x Intel Xeon Platinum 8160 multi-core CPUs, with 24 cores each (i.e. 48 cores per node) and 96 GBytes of RAM.

With respect to the software, we used `FEMPAR` v1.0.0 [3], compiled with Intel v18.0.5 compilers using system recommended optimization flags. `FEMPAR` was linked against the Intel Message Passing Interface (MPI) Library (v2018.4.274) for message-passing and the BLAS/LAPACK and PARDISO available on the Intel MKL library for optimized dense linear algebra kernels and sparse direct solvers, resp. All floating-point operations were performed in IEEE double precision.

4.2 p-Laplacian problem

4.2.1 Problem description

We study the weak scalability[2] on MN-IV of the nonlinear solver algorithms presented in Sect. 3 when applied to the solution of the following 2D p-Laplacian model problem:

$$-\nabla \cdot (|u|^{p-2} \nabla u) = 0 \quad \text{in}\, \Omega, \tag{1}$$

$$u = g \quad \text{on}\, \partial\Omega, \tag{2}$$

with a rectangle domain $\Omega = [0, l_x] \times [0, l_y]$, and Dirichlet data given by $g = x + y$. This problem has known analytical solution $u = x + y$.

The linearized residual of (1) at a given nonlinear iterate $u = u^*$, i.e., $R(u^*)(v)$, is given by the following expression:

$$R(u^*)_i := \int_\Omega |\nabla u^*|^{p-2} \nabla u^* \cdot \nabla \varphi_i \mathrm{d}\Omega \ , \tag{3}$$

where we have used the basis representation of v. On the other hand, by computing the derivative of this expression i.e., $DR(u^*)(v)$, in the direction w, and by using the the basis representation of v and w, we obtain the following expression for the Jacobian matrix required for the Newton-Raphson nonlinear problem solver:

$$DR(u^*)_{ij} := \int_\Omega |\nabla u^*|^{p-2} \nabla \varphi_i \cdot \nabla \varphi_j \mathrm{d}\Omega + (p-2) \int_\Omega |\nabla u^*|^{p-4} (\nabla u^* \cdot \nabla \varphi_i)(\nabla u^* \cdot \nabla \varphi_j) \mathrm{d}\Omega \ . \tag{4}$$

In the numerical experiments, we also consider an approximate version of the Jacobian matrix given by

$$DR(u^*)_{ij} :\approx \int_\Omega |\nabla u^*|^{p-2} \nabla \varphi_i \cdot \nabla \varphi_j \mathrm{d}\Omega \, , \tag{5}$$

leading to a Quasi-Newton-Raphson nonlinear iterative scheme.

[2] Weak scalability is the ability of a *parallel system* (i.e., a software implementation of an algorithm run on a parallel computer) to efficiently exploit increasing computational resources (i.e., CPU cores, memory, etc.) in the solution of proportionally larger problem sizes. An ideally weakly scalable system maintains its efficiency (i.e., number of iterations, time-to-solution, etc.) as we increase the number of CPU cores in the same proportion as the global problem size, i.e., with a fixed problem size per processor.

4.2.2 Weak scaling study: setup, results and discussion

We consider $p = 3$ in (1) for all experiments reported in this deliverable. For the discretization of this problem, we use a global conforming uniform partition (mesh) of Ω into hexahedra and bilinear FEs i.e., $Q_1(K)$ FEs. This 2D mesh is partitioned uniformly (optimally) into a rectangular grid of $P_x \times P_y$ subdomains. The size of the local FE mesh at each subdomain is $\frac{H}{h} \times \frac{H}{h}$ hexahedral cells. The size of the global FE mesh is therefore equal to $P_x \frac{H}{h} \times P_y \frac{H}{h}$ hexahedra. We in particular scale the subdomain grid $P_x \times P_y$ as $8k \times 6k$, with $k = 1, 2, 3, 4$, and consider four different values for the subdomain size $\left(\frac{H}{h}\right)^2 = 10^2, 20^2, 30^2, 40^2$, i.e., four different fixed loads per core for the weak scaling study. The domain is scaled as $\Omega = [0, 1.33k] \times [0, 1.0k]$, leading to square-shape subdomains.

We set up the nonlinear and linear iterative solvers as follows:

1. The outer Newton-Raphson iteration is stopped whenever the linearized residual at a given iteration satisfies $||R(u^*)||_2 \leq$ `nls_rtol` $* \, ||R(u^0)||_2 +$ `nls_atol`, with u^* being the current solution iterate, u^0 the initial solution iterate, and `nls_rtol` and `nls_atol` the relative and absolute linearized residual tolerances, resp. We in particular used $u^0 = \frac{1}{2}(x + y)$ as initial solution iterate, `nls_rtol` $= 10^{-8}$, and `nls_atol` $= 10^{-14}$.

2. The BDDC-PCG linear solver for the Jacobian linear system at each Newton-Raphson iteration is stopped whenever $||r^k||_2 \leq$ `ils_rtol` $* \, ||r^0||_2 +$ `ils_atol`, with r^k being the linear system residual corresponding to the current iterate, r^0 the initial residual, and `ils_rtol` and `ils_atol` the relative and absolute linear system residual tolerances. We in particular used $x^0 = 0$ as initial solution, `ils_rtol` $= 10^{-6}$, and `ils_atol` $= 10^{-14}$. The BDDC preconditioner is configured such that the BDDC space correction is continuous on corners and edges of the partition of the global mesh into subdomains.

3. In the particular case of the nonlinear BDDC solver, the interior Newton-Raphson iteration at each local subdomain is stopped whenever $||R_i^I(u_i^*)||_2 \leq$ `nlic_rtol` $* \, ||R_i^I(u_i^0)||_2 +$ `nlic_atol`, or a maximum of 5 nonlinear iterations is reached. $R_i^I(u_i^*)$ denotes the linearized residual restricted to the interior DOFs of subdomain Ω_i at the local current iterate u_i^*, and u_i^0 the initial solution iterate at subdomain Ω_i. As required by the nonlinear BDDC solver, u_i^0 is just the restriction of u^* (i.e., the current solution iterate of the outer nonlinear iteration) to the DOFs of subdomain Ω_i. `nlic_rtol` and `nlic_atol` play the role of `nls_rtol` and `nls_atol`, resp., in the case of the interior Newton-Raphson iteration. We used `nlic_rtol` $= 10^{-2}$ and `nlic_rtol` $= 10^{-14}$.

The results of the weak scaling study are provided in Table 1 and 2 for the two different ways of computing the Jacobian matrix, i.e., (4) and (5), resp. In the tables, we provide three different performance indicators for PCG-BDDC-Newton-Raphson and Nonlinear BDDC; see Sect 3. In particular, the columns labelled as "#NL iters." provide the number of outer Newton-Raphson solver iterations, the ones labelled as "#PCG iters.", the minimum and maximum number of PCG-BDDC linear solver iterations among all outer iterations, and the ones labelled as "T", the total computation time in seconds.

Let us first discuss the results in Table 1. First, we can observe almost ideal weak scaling in terms of "#NL iters." and "#PCG iters.". In particular, an almost constant number of iterations is observed when the global problem size is scaled by a factor of 16. For example, with a fixed load per core $\frac{H}{h} = 40$, ”#NL iters.“ and ”#PCG iters.“ scale as 11, 11, 12, and 12, and 6-8, 6-8, 6-8, and 5-8, resp., for the PCG-BDDC-Newton-Raphson nonlinear solver, and as 10, 10, 11, 11, and 7-8, 7-10, 7-12, 7-14, resp., for the nonlinear BDDC solver. These results confirm, on the one hand, optimality of the BDDC preconditioning approach (i.e., condition numbers of the preconditioned Jacobian linear systems bounded by a constant as we scale the global problem size) and, on the other hand, that this optimality is cornerstone for the high scalability of the whole nonlinear iterative scheme. Second, although scalability in terms of computation times is not as ideal, it is still very remarkable. For example, with a fixed load per core $\frac{H}{h} = 40$, “T“ increases from 0.695 to 1.019 seconds for PCG-BDDC-Newton-Raphson, and from 1.434 to 2.194 seconds for nonlinear BDDC. In other words, computation times only mildly increase by a factor of 1.47 and 1.53, resp., when the global problem size is scaled by a factor of 16. Finally, if we compare the performance of both nonlinear solvers, we can observe that PCG-BDDC-Newton-Raphson is up-to twice as fast as nonlinear BDDC. The extra computational cost spent by nonlinear BDDC in the iterative solution of the nonlinear local-to-subdomain problems is not compensated by the very mild reduction in the outer Newton-Raphson iterations, resulting in a higher total time-to-solution. Apart from the former factor, a higher number of PCG-BDDC linear solver iterations of nonlinear BDDC compared to PCG-BDDC-Newton-Raphson also contributes to nonlinear BDDC being slower than PCG-BDDC-Newton-Raphson.

		PCG-BDDC-Newton-Raphson			Nonlinear BDDC		
subdomain grid	$\frac{H}{h}$	#NL	#PCG iters.	T	#NL	#PCG iters.	T
($P_x \times P_y$)		iters.	(min-max)	(secs.)	iters.	(min-max)	(secs.)
8×6	10	9	5-6	0.053	8	6-8	0.081
	20	10	6-7	0.170	9	7-8	0.301
	30	11	6-7	0.395	9	7-8	0.665
	40	11	6-8	0.695	10	7-8	1.434
16×12	10	10	5-6	0.106	9	6-10	0.135
	20	11	5-7	0.241	10	7-10	0.427
	30	11	6-7	0.431	10	7-10	0.969
	40	11	6-8	0.736	10	7-10	1.661
24×18	10	11	5-6	0.185	9	6-12	0.211
	20	11	5-7	0.315	10	7-12	0.559
	30	11	5-7	0.510	11	7-12	1.170
	40	12	6-8	0.886	11	7-12	1.977
32×24	10	11	5-6	0.305	10	6-14	0.385
	20	11	5-7	0.422	10	7-14	0.714
	30	12	5-7	0.736	11	7-14	1.354
	40	12	5-8	1.019	11	7-14	2.194

Table 1: 2D $p = 3$-Laplacian problem results on the MN-IV supercomputer. The Jacobian of the linearized residual was computed as (4).

The results reported in Table 2 reveal a quite different overall picture to the ones reported in Table 1. Although computing (5) is cheaper than (4), the approximate computation of the Jacobian matrix leads to a significant degradation of the outer nonlinear solver convergence speed, i.e., a higher value of ”#NL iters.“ in Table 2 than in Table 1. Due to this factor, both PCG-BDDC-Newton-Raphson and nonlinear BDDC are less efficient (i.e., slower) in Table 2 than in Table 1 for all subdomain grids and local problem sizes tested. There are still several worth-mentioning observations that can be extracted from the results in Table 2. First, for a fixed subdomain grid, the higher the local problem size (i.e., the higher the local mesh resolution), the less ”#NL iters.“, for both PCG-BDDC-Newton-Raphson and nonlinear BDDC. For example, for 8 × 6 subdomains and PCG-BDDC-Newton-Raphson, ”#NL iters.“ drops from 461 to 231 when $\frac{H}{h}$ is scaled from 10 to 40, resp., and from 71 to 50 for nonlinear BDDC. Second, for a fixed subdomain grid, and ”small“ $\frac{H}{h}$, nonlinear BDDC significantly cut downs ”#NL iters.“ compared to PCG-BDDC-Newton-Raphson up-to an extent that the extra computation cost spent by nonlinear BDDC in the iterative solution of the nonlinear local-to-subdomain problems pays off this reduction in the outer Newton-Raphson iterations. Indeed, e.g., in Table 2, nonlinear BDDC is faster than PCG-BDDC-Newton-Raphson with $\frac{H}{h} = 10$ for all subdomain grids considered in the study. Third, for a fixed subdomain grid, ”#NL iters.“ drops down faster with $\frac{H}{h}$ for PCG-BDDC-Newton-Raphson, up-to an extent that, for the largest three subdomain grids, PCG-BDDC-Newton-Raphson ends up being faster than nonlinear BDDC with $\frac{H}{h}$. Finally, if the trend observed in Table 2 is maintained with larger scales up-to an extent that PCG-BDDC-Newton-Raphson (resp., nonlinear BDDC) converges with (5) in a ”close enough“ number of iterations to those with (4), then it might be possible that the approximate Jacobian computation leads to a more efficient (faster) overall nonlinear iteration scheme.

		PCG-BDDC-Newton-Raphson			Nonlinear BDDC		
subdomain grid	$\frac{H}{h}$	#NL	#PCG iters.	T	#NL	#PCG iters.	T
$(P_x \times P_y)$		iters.	(min-max)	(secs.)	iters.	(min-max)	(secs.)
8 × 6	10	461	4-7	2.92	71	7-8	1.02
	20	454	5-8	8.28	60	7-8	2.92
	30	347	5-8	13.2	54	7-9	5.71
	40	231	5-9	16.3	50	7-9	9.43
16 × 12	10	454	4-7	5.20	84	6-10	1.68
	20	231	4-8	5.40	66	7-9	3.53
	30	112	5-9	4.94	56	8-9	6.20
	40	66	5-9	4.92	59	8-9	9.65
24 × 18	10	347	4-7	6.56	97	6-12	2.66
	20	112	4-8	3.56	69	7-11	4.33
	30	53	4-8	2.69	54	8-10	6.63
	40	31	5-9	2.50	44	8-10	9.07
32 × 24	10	231	4-7	6.94	108	6-13	4.23
	20	66	4-8	2.92	66	7-12	4.95
	30	31	4-8	2.00	47	8-11	6.44
	40	18	4-9	1.64	37	8-11	8.34

Table 2: 2D $p = 3$-Laplacian problem results on the MN-IV supercomputer. The Jacobian of the linearized residual was computed as (5).

4.3 Low-frequency electromagnetic response of High Temperature Superconductors

Next, we study the scalability of the algorithm with a practical application, the modelling of HTS. The problem consists in the magnetization of a superconducting cube completely surrounded by a dielectric material (see Fig. 2a), subjected to an external AC magnetic field. The formulation

$$\boldsymbol{\nabla} \times (\alpha \boldsymbol{\nabla} \times \boldsymbol{u}) + \beta \boldsymbol{u} = \boldsymbol{f} \quad \text{in}\, \Omega, \tag{6}$$

$$\boldsymbol{n} \times (\boldsymbol{u} \times \boldsymbol{n}) = 0 \quad \text{on}\, \partial\Omega, \tag{7}$$

where $\alpha \geq 0$, $\beta > 0$ are the resistivity and the magnetic permeability of the materials, respectively, $\boldsymbol{n}$ is a unit normal to the boundary and $\boldsymbol{\nabla}\times$ is the 3D curl operator (see [18]), arises in the time-domain quasi-static approximation of the Maxwell's equations for solving the magnetic field, see [20] for details. Furthermore, the standard Backward Euler method is used to perform time integration over a time interval $[0, T]$, so let us define a time partition $\{0 = t^0, t^1, \ldots, t^N = T\}$ into N time elements. Then, the form Eq. (6) can be used to compute the magnetic field for a particular time t^n, provided the solution on the previous time $\boldsymbol{u}^{n-1}$. The coefficient β is affected by the current time step size $\Delta t = (t^n - t^{n-1})$ as $\beta = \frac{\mu_0}{\Delta t}$, where $\mu_0 = 4\pi \cdot 10^{-7}$ is the magnetic permeability of the vacuum. While the dielectric material is modelled with a constant value for $\alpha = 10^{-3}$, the superconducting material behaviour is modelled with the stiff nonlinear dependence of the resistivity α with the solution as $\alpha = \alpha_0 \left(\frac{\|\boldsymbol{\nabla} \times \boldsymbol{u}\|}{J_c} \right)^m$, with $m = 100$, $J_c = 10^{-8}$ and $\alpha_0 = 10^{-12}$. The equivalence with Eq. (6) is completed by considering the source term $\boldsymbol{f} = \beta \boldsymbol{u}^{n-1}$ and the strong imposition of an external magnetic field $\boldsymbol{u}^n \times \boldsymbol{n} = \boldsymbol{u}_0^n$ over the whole boundary. For the time step t^n, the weak form of the nonlinear problem reads: find $\boldsymbol{u}^n \in X_h$ such that

$$(\alpha(\boldsymbol{u}^n) \boldsymbol{\nabla} \times \boldsymbol{u}^n, \boldsymbol{\nabla} \times \boldsymbol{v}) + \beta(\boldsymbol{u}^n, \boldsymbol{v}) = \beta(\boldsymbol{u}^{n-1}, \boldsymbol{v}) \quad \forall \boldsymbol{v} \in X_h. \tag{8}$$

In order to derive the linearized form with Newton's method we consider the current approximation $\boldsymbol{u}^{n,k}$ and a (small) correction $\delta\boldsymbol{u}^{n,k}$ for the iterate k such that $\boldsymbol{u}^{n,k+1} = \boldsymbol{u}^{n,k} + \delta\boldsymbol{u}^{n,k}$. We plug the expression in Eq. (8), consider a first order Taylor expansion of $\alpha(\boldsymbol{u}^{n,k+1})$ around $\boldsymbol{u}^{n,k}$ and neglect the quadratic terms with respect to $\delta\boldsymbol{u}^{n,k}$, which yields the linearized problem: find $\delta\boldsymbol{u}^{n,k} \in X_h$ such that

$$\mathcal{J}(\boldsymbol{u}^{n,k}, \delta\boldsymbol{u}^{n,k}, \boldsymbol{v}) = -\mathcal{R}(\boldsymbol{u}^{n-1,k}, \boldsymbol{u}^{n,k}, \boldsymbol{v}) \qquad \forall \boldsymbol{v} \in X_h, \tag{9}$$

where

$$\begin{aligned}\mathcal{J}(\boldsymbol{u}^{n,k}, \delta\boldsymbol{u}^{n,k}, \boldsymbol{v}) &= (\alpha(\boldsymbol{u}^{n,k}) \boldsymbol{\nabla} \times \delta\boldsymbol{u}^{n,k}, \boldsymbol{\nabla} \times \boldsymbol{v}) + \beta(\delta\boldsymbol{u}^{n,k}, \boldsymbol{v}) + \qquad \text{(10a)}\\ &\quad + (\alpha'(\boldsymbol{u}^{n,k}) \delta\boldsymbol{u}^{n,k} \boldsymbol{\nabla} \times \boldsymbol{u}^{n,k}, \boldsymbol{\nabla} \times \boldsymbol{v}),\\ \mathcal{R}(\boldsymbol{u}^{n-1,k}, \boldsymbol{u}^{n,k}, \boldsymbol{v}) &= -\beta(\boldsymbol{u}^{n-1}, \boldsymbol{v}) + \beta(\boldsymbol{u}^{n,k}, \boldsymbol{v}) + (\alpha(\boldsymbol{u}^{n,k}) \boldsymbol{\nabla} \times \boldsymbol{u}^{n,k}, \boldsymbol{\nabla} \times \boldsymbol{v}). \qquad \text{(10b)}\end{aligned}$$

Therefore, the rPB-BDDC preconditioner is applied to the linearized problem Eq. (9) at every nonlinear iteration of every time step. We will focus on the performance of the linear solver, and the reader is directed to [20] for a detailed exposition of the composition of the used transient nonlinear solver.

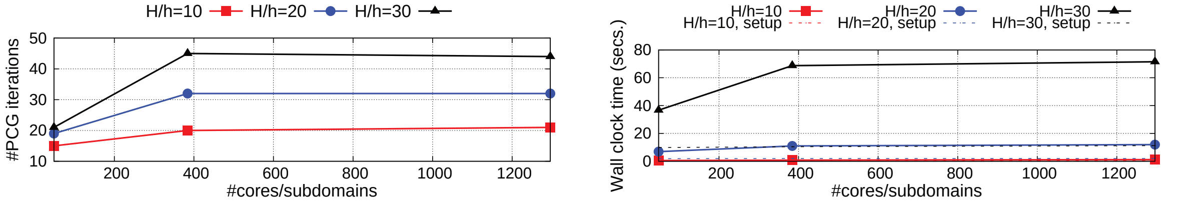
The problem is solved in $\Omega = [0, 40]^3$ mm^3, composed by an outer dielectric Ω_{air} material which includes a concentric superconducting cube Ω_{hts} of size 10 mm such that $\Omega = \Omega_{\text{hts}} \cup \Omega_{\text{air}}$, see Fig. 2a. There is no source term and Dirichlet-type boundary conditions are imposed over the entire boundary as the time-dependent magnetic field

P	# Average iters.	Average size(A_c)	size(A_c) ratio
49	16.8	196.7	$1.31c_0$
385	18.9	2424.5	$2.02c_0$
1297	21.7	7728.7	$1.91c_0$

Table 3: Average metrics for the simulation of the time interval $T = [0, 5]$ ms. c_0 denotes the number of coarse DOFs of the original, geometrical partition.

$\boldsymbol{u}_{ext} = \frac{B_0}{\mu_0}[0, 0, \sin(2\pi\omega t)]$, where $B_0 = 200$ mT and $\omega = 50$ Hz. We solve the problem in the time interval $[0, 5]$ ms, which corresponds to a quarter of a full cycle in the applied $\boldsymbol{u}_{ext}$. Initial conditions are simply $\boldsymbol{u}^0 = \mathbf{0}$. The partition Θ^r_{pb} is obtained in all simulations for $r = 10^2$ (see [2] for details). The nonlinear scheme is stopped when the L_2-norm of the nonlinear residual (Eq. (10b)) is below 10^{-4}, while the convergence criteria for the rPB-BDDC preconditioned linear solver is the reduction of the initial L_2-norm of the residual of the linearized system by 10^{-8}.

We first present weak scalability results for the first set-up and solve with the rPB-BDDC preconditioner in Fig. 1, i.e., the first linearized problem (Eq. (9)) for the first time step. We include results for $\frac{H}{h} = \{10, 20, 30\}$. As expected, the method shows good weak scalability properties in number of iterations (see Fig. 1a) and computing times (see Fig. 1b).



(a) Number of iterations.

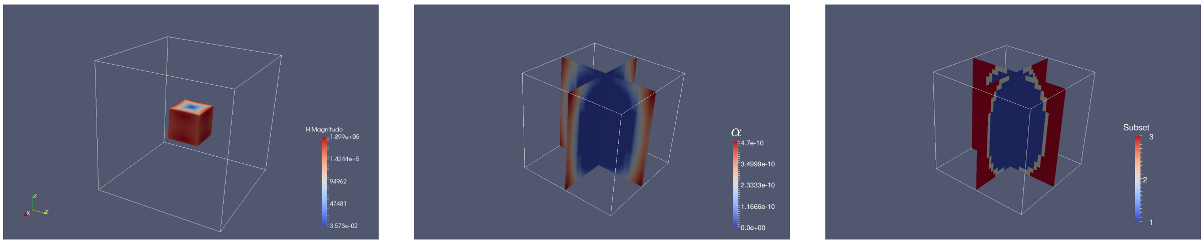
(b) Wall clock time for the total solver and only the set-up phase.

Figure 1: Weak scalability for the first linear solver in the HTS problem with $r = 10^2$.

Next, we present average counters for the total number of linear solver applications for the simulation of the whole time interval $[0, 5]$ ms in Table 3, for a local problem size of $H/h = 10$ and different partitions. The resulting aggregation of cells into subsets based on their physical coefficient α (see [2]) for $t = 4$ ms is depicted in Fig. 2c. We can identify two main regions in the distribution of α (see Fig. 2b): an inner region that is still not magnetized (i.e., with null resistivity) and a surrounding region, separated by a thin layer. Therefore, the selected value for r allows us to capture the behaviour of the different regions in Ω_{hts}. Out of the results in Tab. 3, the most salient property is the (asymptotic) scalability in the average number of iterations. Besides, we show how the coarse problem size for the presented cases is only (approximately) doubled regarding to the size that would be obtained with the partition Θ instead of the Θ_{pb}.

A FEMPAR ’s tutorial program for p-Laplacian problem

In order to illustrate the usage of the nonlinear solvers implemented in this project task, we developed a FEMPAR tutorial program for the p-Laplacian problem. This driver program lets FEMPAR users to play around with several linear and nonlinear solvers and problem



(a) Magnetic field strength in the HTS device surrounded by a dielectric box, for which only the outline is depicted.

(b) Distribution of α in the HTS device.

(c) Subsets arising by the rPB-partition with $r = 10^2$ in the HTS device.

Figure 2: Domain and HTS device for $t = 4$ ms.

and solver parameters through command-line parameter-value arguments. This driver program, together with the nonlinear solvers implemented in this task, will be distributed as part of the source code `Git` repository of `FEMPAR` in a forthcoming release. In Fig. 3 we illustrate an example command-line invocation to this tutorial program.

```
mpiexec --tag-output -np 5 par_test_p_laplacian \
   --STRUCT_HEX_TRIANG_NUM_LEVELS 2 \
   --STRUCT_HEX_TRIANG_NUM_DIMS 2 \
   --STRUCT_HEX_TRIANG_NUM_CELLS_DIM 4 4 0 \
   --STRUCT_HEX_TRIANG_NUM_PARTS_X_DIR 2 2 0 1 1 0 \
   --STRUCT_HEX_TRIANG_DOMAIN_LIMITS 0.0 1.0 0.0 1.0 0.0 0.0 \
   --FES_REF_FE_ORDERS 1 \
   --ILS_RTOL 1.0e-06 \
   --ILS_ATOL 1.0e-14 \
   --ILS_TYPE CG \
   --ILS_MAX_NUM_ITERATIONS 5000 \
   --NLS_RTOL 1.0e-08 \
   --NLS_ATOL 1.0e-14 \
   --NLS_MAX_NUM_ITERATIONS 1000 \
   --enable-nlic .true. \
   --nlic-rtol 1.0e-02 \
   --nlic-atol 1.0e-14 \
   --nlic-max-num-iters 5 \
   --p-lap-exp 3 \
   --enable-full-jacobian .true. \
   --write-solution .true.
```

Figure 3: Example of an invocation to the `FEMPAR` 's tutorial program in charge of the solution of the p-Laplacian benchmark problem.

References

[1] S. Badia, A. F. Martín, and J. Principe. Multilevel balancing domain decomposition at extreme scales. *SIAM Journal on Scientific Computing*, pages C22–C52, 2016. ISSN 1064-8275. doi:10.1137/15M1013511. URL `http://epubs.siam.org/doi/abs/10.1137/15M1013511`.

[2] S. Badia, A. Martín, and M. Olm. Scalable solvers for complex electromagnetics problems. *arXiv:1901.08783*, 2017.

[3] S. Badia, A. F. Martín, and J. Principe. FEMPAR: An Object-Oriented Parallel Finite Element Framework. *Archives of Computational Methods in Engineering*, pages 1–77, 2017.

[4] S. Badia, A. Martín, and J. Principe. FEMPAR Web page. http://www.fempar.org, 2018.

[5] S. Badia, A. F. Martín, and H. Nguyen. Physics-Based Balancing Domain Decomposition by Constraints for Multi-Material Problems. *Journal of Scientific Computing*, pages 1–30, 2018. ISSN 0885-7474. doi:10.1007/s10915-018-0870-z. URL http://link.springer.com/10.1007/s10915-018-0870-z.

[6] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2018.

[7] P. Brune, M. Knepley, B. Smith, and X. Tu. Composing scalable nonlinear algebraic solvers. *SIAM Review*, 57(4):535–565, 2015. ISSN 0036-1445. doi:10.1137/130936725. URL http://epubs.siam.org.recursos.biblioteca.upc.edu/doi/abs/10.1137/130936725.

[8] X. Cai and D. Keyes. Nonlinearly preconditioned inexact newton algorithms. *SIAM Journal on Scientific Computing*, 24(1):183–200, 2002. ISSN 1064-8275. doi:10.1137/S106482750037620X. URL http://epubs.siam.org/doi/abs/10.1137/S106482750037620X.

[9] J. Calvo and O. Widlund. An adaptive choice of primal constraints for BDDC domain decomposition algorithms. *Electronic Transactions on Numerical Analysis*, 45:524–544, 2016. ISSN 1068-9613.

[10] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM Journal on Scientific Computing*, 25(1):246–258, 2003. ISSN 10648275. doi:10.1137/S1064827502412887. URL http://link.aip.org/link/SJOCE3/v25/i1/p246/s1&Agg=doi.

[11] C. R. Dohrmann and O. B. Widlund. Some recent tools and a BDDC algorithm for 3D problems in H(curl). In R. Bank, M. Holst, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering XX*, pages 15–25, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-35275-1.

[12] C. R. Dohrmann and O. B. Widlund. A BDDC algorithm with deluxe scaling for three-dimensional H(curl) problems. *Communications on Pure and Applied Mathematics*, 69(4):745–770, 2016. doi:10.1002/cpa.21574.

[13] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis*, 40(1):159–179, 2002. doi:10.1137/S0036142901388081. URL https://doi.org/10.1137/S0036142901388081.

[14] A. Klawonn, M. Lanser, and O. Rheinbach. Nonlinear {FETI}-{DP} and {BDDC} {Methods}. *SIAM Journal on Scientific Computing*, 36(2):A737—-A765, jan 2014. ISSN 1064-8275. doi:10.1137/130920563. URL http://epubs.siam.org/doi/abs/10.1137/130920563.

[15] A. Klawonn, P. Radtke, and O. Rheinbach. Adaptive coarse spaces for BDDC with a transformation of basis. In T. Dickopf, M. J. Gander, L. Halpern, R. Krause, and L. F. Pavarino, editors, *Domain Decomposition Methods in Science and Engineering XXII*, pages 301–309, Cham, 2016. Springer International Publishing. ISBN 978-3-319-18827-0.

[16] J. Mandel. Balancing domain decomposition. *Communications in Numerical Methods in Engineering*, 9(3):233–241, 1993. ISSN 1099-0887. doi:10.1002/cnm.1640090307.

[17] J. Mandel and B. Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Computer Methods in Applied Mechanics and Engineering*, 196(8):1389–1399, Jan. 2007. ISSN 0045-7825. doi:10.1016/j.cma.2006.03.010.

[18] P. Monk. *Finite Element Methods for Maxwell's Equations.* Oxford Science Publications, 2003.

[19] D. Oh, O. Widlund, Z. Zampini, and C. Dohrmann. BDDC algorithms with deluxe scaling and adaptive selection of primal constraints for Raviart-Thomas vector fields. *Math. Comp.*, 87:659–692, 2018.

[20] M. Olm, S. Badia, and A. Martín. Simulation of high temperature superconductors and experimental validation. *arXiv:1707.09783*, 2017.

[21] C. Pechstein and C. R. Dohrmann. A unified framework for adaptive BDDC. *Electronic Transactions on Numerical Analysis*, 46:273–336, 2017.

[22] C. Pechstein and R. Scheichl. Analysis of FETI methods for multiscale PDEs. Part ii: interface variation. *Numerische Mathematik*, 118(3):485–529, Jul 2011. ISSN 0945-3245. doi:10.1007/s00211-011-0359-2. URL `https://doi.org/10.1007/s00211-011-0359-2`.

[23] B. Sousedík, J. Šístek, and J. Mandel. Adaptive-Multilevel BDDC and its parallel implementation. *Computing*, 95(12):1087–1119, Dec. 2013. ISSN 0010-485X, 1436-5057. doi:10.1007/s00607-013-0293-5.

[24] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. A robust two-level domain decomposition preconditioner for systems of PDEs. *Comptes Rendus Mathematique*, 349(23):1255 – 1259, 2011. ISSN 1631-073X. doi:https://doi.org/10.1016/j.crma.2011.10.021. URL `http://www.sciencedirect.com/science/article/pii/S1631073X11003049`.

[25] N. Spillane, V. Dolean, P. Hauret, F. Nataf, and D. J. Rixen. Solving generalized eigenvalue problems on the interfaces to build a robust two-level FETI method. *Comptes Rendus Mathematique*, 351(5):197 – 201, 2013. ISSN 1631-073X. doi:https://doi.org/10.1016/j.crma.2013.03.010. URL `http://www.sciencedirect.com/science/article/pii/S1631073X1300054X`.

[26] A. Toselli and O. Widlund. *Domain Decomposition Methods: Algorithms and Theory.* Springer-Verlag, Berlin, 2005.

[27] S. Zampini. PCBDDC: A class of robust dual-primal methods in PETSc. *SIAM J. Sci. Comput.*, 38:S282–S306, 2016.

[28] S. Zampini. Adaptive BDDC deluxe methods for H(curl). In C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXIII*, pages 285–292, Cham, 2017. Springer International Publishing. ISBN 978-3-319-52389-7.

[29] S. Zampini and X. Tu. Multilevel balancing domain decomposition by constraints deluxe algorithms with adaptive coarse spaces for flow in porous media. *SIAM Journal on Scientific Computing*, 39(4):A1389–A1415, 2017. doi:10.1137/16M1080653.