

1           **PARALLEL DOMAIN DECOMPOSITION TECHNIQUES APPLIED TO**  
2           **MULTIVARIATE FUNCTIONAL APPROXIMATION OF DISCRETE DATA \***

3           VIJAY S. MAHADEVAN<sup>†</sup>, DAVID LENZ<sup>‡</sup>, IULIAN GRINDEANU<sup>§</sup>, AND THOMAS PETERKA<sup>¶</sup>

4           **Abstract.** Compactly expressing large-scale datasets through Multivariate Functional Approximations  
5 (MFA) can be critically important for analysis and visualization to drive scientific discovery. Tackling  
6 such problems requires scalable data partitioning approaches to compute MFA representations in amenable  
7 wall clock times. We introduce a fully parallel scheme to reduce the total work per task in combination  
8 with an overlapping additive Schwarz based iterative scheme to compute MFA with tensor expansion of  
9 non-uniform B-spline (NURBS) bases, while preserving full degree continuity across subdomain boundaries.  
10 While previous work on MFA has been successfully proven to be effective, the computational complexity of  
11 encoding large datasets on a single process can be severely prohibitive. Parallel algorithms for generating  
12 reconstructions from the MFA have had to rely on postprocessing techniques to blend discontinuities across  
13 subdomain boundaries. In contrast, a robust constrained minimization infrastructure to impose higher-  
14 order continuity directly on the MFA representation is presented here. We demonstrate effectiveness of  
15 the parallel approach with domain decomposition solvers, to minimize the subdomain error residuals of  
16 the decoded MFA, and more specifically to recover continuity across non-matching boundaries at scale.  
17 The analysis of the presented scheme for analytical and scientific datasets in 1-, 2- and 3-dimensions are  
18 presented. Extensive strong and weak scalability performance are also demonstrated for scientific 2D and 3D  
19 datasets to evaluate the parallel speedup of the MPI-based algorithm implementation on large computing  
20 machines.

21           **Key words.** functional approximation, domain decomposition, scalable methods, additive schwarz

22           **MSC codes.** 65D05, 65D15, 65Y05

23           **1. Introduction.** Large-scale discrete data analysis of various scientific computational sim-  
24 ulations often require high-order continuous functional representations that have to be evaluated  
25 anywhere in the domain. Such expansions described as Multivariate Functional Approximations  
26 (MFA) [12] in arbitrary dimensions allow the original discrete data to be compressed, and expressed  
27 in a compact closed form, in addition to supporting higher-order derivative queries (without further  
28 approximations such as finite differences) for complex data analysis tasks. One particular option  
29 is to use the variations of the NURBS bases [30] for the MFA *encoding* of scientific data. Due to  
30 the potentially large datasets that need to be encoded into MFA, the need for computationally  
31 efficient algorithms (in both time and memory) to parallelize the work is critically important. It is  
32 also essential to guarantee that the solution smoothness in the reconstructed (or *decoded*) dataset  
33 is consistently preserved when transitioning from a single MFA block to multiple blocks during  
34 parallelization.

35           Achieving improved performance without sacrificing discretization accuracy requires an infra-  
36 structure that is consistent in the error metrics of the decoded data and an algorithm that remains  
37 efficient in the limit of large number of tasks. In this paper, we will utilize domain decomposition  
38 (DD) techniques [33] with data partitioning strategies to produce scalable algorithms to compute  
39 the MFA that minimizes the solution reconstruction error to reproduce a given dataset within user-

---

\*Submitted to editors .

Funding: This work was funded by the Early Career Research Program, Department of Energy, USA under contract no. XXX-XXXX.

<sup>†</sup>Argonne National Laboratory, Lemont, IL (mahadevan@anl.gov)

<sup>‡</sup>Argonne National Laboratory, Lemont, IL (dlenz@anl.gov)

<sup>§</sup>Argonne National Laboratory, Lemont, IL (iulian@anl.gov)

<sup>¶</sup>Argonne National Laboratory, Lemont, IL (tpeterka@mcs.anl.gov)

40 specified tolerances. In such partitioned analysis, it is imperative to ensure that the continuity of  
 41 the encoded and decoded data across subdomain interfaces are maintained, and remain consistent  
 42 with the degree of underlying expansion bases used in MFA [29]. This is due to the fact that in-  
 43 dependently computing MFA approximations in individual subdomains do not guarantee even  $C^0$   
 44 regularity in either the MFA space or in the reconstructed data. In order to tackle this issue, we rely  
 45 on an iterative Schwarz-type DD scheme to ensure that continuity is enforced, and the overall error  
 46 stays bounded as the number of subdomains are increased (or as the subdomain size decreases).

47 In addition to remaining efficient, we also require the devised algorithms to extend naturally  
 48 to arbitrary dimensional settings and to handle large datasets. We next discuss some of the related  
 49 work in the literature that have been explored for reconstruction of scattered data, and approaches  
 50 to make these algorithms scalable in order to motivate the ideas presented in the paper.

51 **Literature Review.** Domain decomposition techniques in general rely on the idea of split-  
 52 ting a larger domain of interest into smaller partitions or subdomains, which results in coupled  
 53 Degrees-of-Freedom (DoF) at their common interfaces. Typical applications of DD in Boundary-  
 54 Value problems (BVP) [33, 23] have been successfully employed to efficiently compute the solution  
 55 of large, discretized PDEs in a scalable manner. DD techniques for parallel approximation of scat-  
 56 tered data have been explored previously with Radial Basis Functions (RBF) [24], yielding good  
 57 scalability and closely recovering the underlying solution profiles. In general, overlapping mul-  
 58 tiplicative and additive Schwarz [34] iterative techniques for RBF [22] have proven successful to  
 59 tackle large-scale problems. Additionally, the use of Restricted variants of Additive-Schwarz (RAS)  
 60 method as preconditioners, with Krylov iterative solvers, can yield iterative schemes [36] with  $O(N)$   
 61 computational complexity, as opposed to the typical  $O(N \log(N))$  complexity with traditional RBF  
 62 reconstructions [1]. The extensions of these ideas to NURBS bases exposes a way to fully parallelize  
 63 traditional, serial MFA computations.

64 Application of DD schemes and NURBS bases with isogeometric analysis (IGA) [8] to high-  
 65 fidelity modeling of nonlinear Partial Differential Equations (PDEs) have enjoyed recent success  
 66 [13, 25, 9] at scale, but many implementations lack full support to handle multiple geometric  
 67 patches in a distributed memory setting due to non-trivial requirements on continuity constraints at  
 68 patch boundaries. Directly imposing higher-order geometric continuity in IGA requires specialized  
 69 parameterizations in order to preserve the approximation properties [21] and can be difficult to  
 70 parallelize [19] generally.

71 In a similar vein, using NURBS bases to compute the MFA in parallel, while maintaining higher-  
 72 order continuity across subdomain patches has not been fully explored previously. To overcome  
 73 some of these issues with discontinuities along NURBS patches, Zhang et al. [37] proposed to  
 74 use a gradient projection scheme to constrain the value ( $C^0$ ), the gradient ( $C^1$ ), and the Hessian  
 75 ( $C^2$ ) at a small number of test points for optimal shape recovery. Such a constrained projection  
 76 yields coupled systems of equations for control point data for local patches, and results in a global  
 77 minimization problem that needs to be solved.

78 Alternatively, it is possible to create a constrained recovery during the actual post-processing  
 79 stage i.e., during the decoding stage of the MFA through standard blending techniques [18], in order  
 80 to recover continuity in the decoded data. However, the underlying MFA representation remains  
 81 discontinuous, and would become more so with increasing number of subdomains without the  
 82 ability to recover higher-order derivatives along these boundaries. Moreover, selecting the amount  
 83 of overlaps and resulting width of the blending region relies strongly on a heuristic, which can be  
 84 problematic for general problem settings.

85 In contrast, we propose extensions to the constrained solvers used by Zhang et al. [37] and Xu et

al. [35], and introduce a two-level, DD-based, parallel iterative scheme to enforce the true degree of continuity. The outer iteration utilizes the RAS method [17], with efficient inner subdomain solvers that can handle linear Least-Squares systems to minimize the decoded residual within acceptable error tolerances. The inner subdomain solves can utilize adaptive MFA computations as well [27] with knot insertions and deletions to recover better reconstructions. Such an iterative solver has low memory requirements that scales weakly with growing number of subdomains, and necessitates only nearest-neighbor communication of the interface data once per outer iteration to converge towards consistent MFA solutions.

**Structure of the paper.** The paper is organized as follows. Section (2) presents the theory and necessary details about the subdomain solvers, and the domain decomposition approach used to converge the boundary continuities across MFA patches. Next, in Section (3), the DD solver is applied to several 1D, 2D and 3D synthetic and real-world datasets to verify error convergence, and the parallel scalability of the iterative algorithm for decreasing subdomain sizes is demonstrated. Finally, key observations from the parallel MFA solver and future extensions to more complex cases with spatial adaptivity is also explored in Section (4).

**2. Approach.** With motivations to accelerate the computation of an accurate MFA representation scalably, we utilize a data decomposition approach with overlapping subdomains to create shared layers of piecewise accurate functional reconstructions. This is similar to a multipatch approach typically taken in IGA computations [8, 9]. However, in order to ensure that higher-order continuity across domain boundaries are preserved, an outer iteration loop is inevitable to converge the shared unknowns across the interfaces. These global iterations guarantee consistent MFA encodings in parallel, without which the representations will not even ensure  $C^0$  regularity.

In this section, we first provide an illustrative example by formulating the constrained minimization problem to be solved in each subdomain and explain the iterative methodology used in the current work to converge the shared degrees-of-freedoms (DoFs). We will also introduce the idea of using open vs closed knots, which are clamped or free respectively at subdomain boundaries and discuss the advantages of using one approach over the other.

**2.1. Numerical Background.** A  $p$ -th degree NURBS or B-spline curve [30] is defined using the Cox-deBoor functions for each subdomain as

$$(2.1) \quad \vec{C}(u) = \sum_{i=0}^n R_{i,p}(u) \vec{P}(i), \quad \forall u \in \Omega$$

$$(2.2) \quad R_{i,p}(u) = \frac{N_{i,p}(u)W_i}{\sum_{i=0}^n N_{i,p}(u)W_i}$$

where  $R_{i,p}(u)$  are the piecewise rational functions with  $\vec{P}$  control points of size  $n$ ,  $W_i$  are the control point weights, with the  $p$ -th degree B-spline bases  $N_{i,p}(u)$  defined on a knot-vector  $u$ . Note that exact high-order derivatives of these NURBS basis defined in Equation (2.2) can also be evaluated without any approximation errors at the control point locations using the Cox-deBoor recurrence relations [12].

Given a set of input points  $\vec{Q}$  that need to be encoded into a MFA, with the weights  $W = 1$  (B-spline representations) for simplicity, the unconstrained minimization problem to compute the optimal set of control point locations within a subdomain can be posed as a solution to a linear Least-Squares (LSQ) system that minimizes the net error of the B-spline approximation.

$$126 \quad (2.3) \quad \arg \min_{\vec{P} \in \mathbb{R}^n} E = \left\| \vec{Q} - R\vec{P} \right\|_{L_2}, \quad R \in \mathbb{R}^{m \times n}, \vec{Q} \in \mathbb{R}^m$$

127 An appropriate LSQ solver such as the one based on Cholesky decomposition or the more ef-  
 128 ficient  $\ell$ -BFGS scheme [38] can compute the control point solution  $\vec{P}$  that minimizes the residual  
 129 error  $\vec{E}$  for the given input data  $\vec{Q}$  and MFA representation of degree  $p$ . Note that the minimization  
 130 procedure is independent as there are no constraints explicitly specified in Equation (2.3). How-  
 131 ever, in order to recover high-order continuity across subdomain patches, computing unconstrained  
 132 solutions is insufficient. At a minimum, the DoFs on the shared subdomain boundaries have to be  
 133 converged to recover  $C^0$  continuity for the decoded solution data ( $R\vec{P}$ ). A straightforward approach  
 134 to achieve  $C^0$  continuity is by ensuring the common control point data  $\vec{P}$  at subdomain interfaces  
 135 are clamped with repeated knots, in addition to using clamping at the global domain boundaries.  
 136 In this scheme, the control points exactly interpolate (are clamped to) input data points at the  
 137 subdomain interface boundaries.

138 More generally, the constrained minimization problem to recover continuity can be formulated  
 139 as

$$140 \quad (2.4) \quad R\vec{P} = \vec{Q} \mid \mathcal{C}\vec{P} = \vec{D},$$

141 where  $\mathcal{C}$  is the constraint matrix imposing continuity restrictions on the control points  $P$  along with  
 142 its derivatives, with data exchanged from neighboring domains stored in  $\mathcal{D}$  defined on the interface  
 143  $\partial\Omega_{i,j}$  shared by subdomains  $i$  and  $j$  [30].

144 With the use of penalized constraints ( $\mathcal{C}$ ) and Lagrange multipliers [16, 28], the solution to  
 145 the constrained LSQ problem can recover optimal control point values. Such an approach requires  
 146 in general a good spatial distribution of  $\vec{Q}$ , and yields only low-order continuous approximations  
 147 ( $C^0$  or  $C^1$  at best) when the solution remains smooth across the subdomain interfaces. It should  
 148 also be noted that as the number of subdomains increases, the global solution being computed  
 149 becomes further constrained, and more interpolatory due to clamped DoFs. Moreover, the MFA  
 150 solution computed becomes dependent on the number of subdomains used to decompose the prob-  
 151 lem; i.e., the global control point data  $\vec{P}$  recovers different reconstructions as a function of number  
 152 of subdomains ( $\mathcal{N}$ ) used.

153 While the numerics and implementation of the multipatch MFA can be much simpler with  
 154 clamped knots on all subdomain boundaries, ensuring higher-order continuity would require that  
 155 all  $p$  derivatives of the approximation match as well. As a continuous extension, one could relax the  
 156 interpolatory behavior of clamped knot boundaries by reducing the number of repeated knots, and  
 157 instead use floating (or unclamped) knots at internal subdomain boundary interfaces by sharing knot  
 158 spans between subdomains. This modification allows us to recover fully consistent ( $C^0$  to  $C^{p-1}$ )  
 159 continuous MFA reconstructions within the solution procedure detailed for the global constrained  
 160 minimization problem Equation (2.4).

161 **2.2. Shared Knot Spans at Subdomain Interfaces.** In this scheme, we utilize floating  
 162 (unclamped) shared knot spans near all interior subdomains such that the high-order continuity  
 163 and consistency of the reconstructed solution with respect to  $\mathcal{N}$  are preserved.

164 For the purpose of illustration and to explain the proposed solver methodology, let us consider  
 165 a simple one dimensional domain ( $\Omega$ ) with two subdomains ( $\mathcal{N} = 2$ ) as shown in Fig. (1), where  
 166  $\Omega_1$  and  $\Omega_2$  represent the subdomains that share an interface  $\partial\Omega_{1,2}$ . In Fig. (1), the layout of the

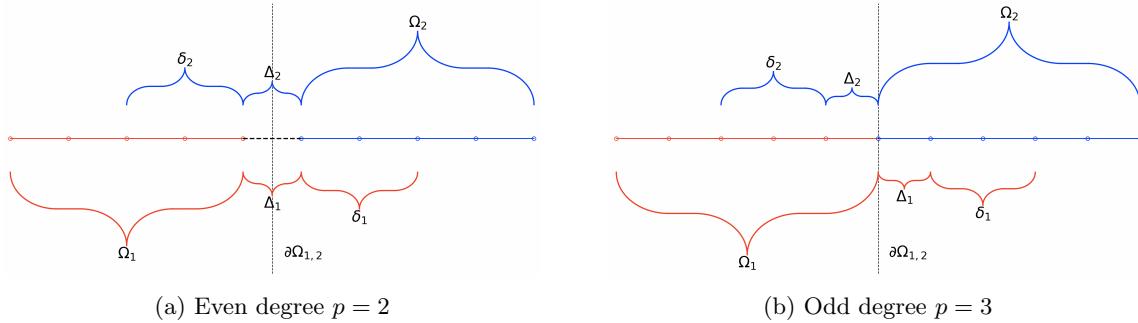


Fig. 1: Illustration: 1-D parallel partitioned domain with floating (unclamped) interior knots and augmented spans ( $|\delta| = 2$ )

knot spans for both an even degree ( $p = 2$ ) and odd degree ( $p = 3$ ) are shown. For generality, we also introduce here an overlap layer  $\Delta_1$  and  $\Delta_2$  on each subdomain that share the knot spans with its adjacent subdomain, and an optional augmented layer  $\delta_1$  and  $\delta_2$  that has a connotation similar to that of an overlap region in traditional DD schemes [33]. Note that in order to reconstruct the input data in  $\Omega_i, \forall i \in [1, 2]$ , the knot spans must mandatorily include  $\Delta_i$  regions. This  $\Delta_i$  overlap region is required by definition to maintain partition of unity of a NURBS or a B-spline curve in order to evaluate Equation (2.2).  $\Delta_i$  represents the repeated knots along clamped global domain boundary and the shared knots between two subdomains in the unclamped interior boundaries. For arbitrary degree  $p$ , the number of knot spans in  $\Delta_i$  is  $\frac{p-1}{2}$  and  $\frac{p}{2}$  for odd and even  $p$  respectively. In multidimensional tensor product expansions, these shared spans are replaced by shared layers of spans along the subdomain interfaces. The  $\delta_i$  regions are additional, and optional, shared regions of knot spans that can help improve error convergence in a manner similar to overlap regions in DD methods used for PDE solvers [17].

Now, the constrained minimization problem for the two subdomain case can be written as

$$(2.5) \quad \left[ \begin{array}{c|c} R_1(\Omega_1) & \lambda_{1,2}(\Delta_1 + \delta_1) \\ \hline \lambda_{2,1}(\Delta_2 + \delta_2) & R_2(\Omega_2) \end{array} \right] \left[ \begin{array}{c} \vec{P}_1 \\ \vec{P}_2 \end{array} \right] = \left[ \begin{array}{c} \vec{Q}_1 \\ \vec{Q}_2 \end{array} \right]$$

where the diagonal operators  $R_1$  and  $R_2$  are the piecewise rational functions that minimize the local subdomain residuals in  $\Omega_j, \forall j \in [1, 2]$ , while the off-diagonal blocks  $\lambda_{1,2}$  and  $\lambda_{2,1}$  represent the coupling terms between the subdomains near the interface  $\partial\Omega_{1,2}$ . This coupling term provides the constraints on the shared control point data, and higher-order derivatives as needed to recover smoothness and enforce continuity along subdomain boundaries. For higher dimensional problems, the constraints on the control points must include both face neighbor and diagonal neighbor contributions to accurately determine the globally consistent minimization problem.

The coupling blocks  $\lambda_{i,j}$  can be viewed as Lagrange multipliers that explicitly couple the control point DoFs across a subdomain interface ( $\vec{P}_1 \cap \vec{P}_2$ ) such that continuity is preserved in a weak sense [30]. Using appropriate Schur complements to eliminate the coupled DoF contributions in each subdomain, with  $\lambda_{i,j}$  evaluated at *lagged* iterates of adjacent subdomains, the set of coupled constrained equations in Equation (2.5) can be completely decoupled for each subdomain. This modified system resembles a block-Jacobi operator of the global system. The scheme introduced

here in the illustration is akin to the Jacobi-Schwarz method [17] that is equivalent to restricted-Additive-Schwarz (RAS) scheme [33].

The overlapping Schwarz solvers [23, 33] have been proven to be efficient and scalable for PDE solvers compared with nonoverlapping variants [34]. We utilize ideas in that vein as a natural extension in terms of overlapping and extending knot spans in the MFA computation in order to produce better reconstructions of the underlying data. This user-specified, additional overlap is described by  $\delta_j, \forall j \in [1, 2]$  in Fig. (1). The amount of data overlap utilized for computing the functional approximation can directly affect the global convergence speed of the iterative scheme, and the scalability of the overall algorithm [4]. Additionally, we expect the residual errors  $\vec{E}$  from the approximation to remain bounded as the number of subdomain increase with appropriate overlap regions.

For the 1D scenario illustrated here, the control point DoF vector can be represented by two separate parts based on the local support of the basis expansion. This is shown below for a two subdomain case with  $p = 2$  and  $|\delta| = 1$ , where the operator  $|\mathcal{D}|$  represents the number of knot spans in any underlying domain  $\mathcal{D}$ .

(2.6)

$$\vec{P}_1 = \begin{bmatrix} P_1(1) \\ P_1(2) \\ \vdots \\ P_1(m) \\ P_2(1) \\ P_2(2) \end{bmatrix}, \vec{P}_2 = \begin{bmatrix} P_2(1) \\ P_2(2) \\ \vdots \\ P_2(n) \\ P_1(m-1) \\ P_1(m) \end{bmatrix}$$

211

where  $m, n$  are the number of control points in  $\Omega_1$  and  $\Omega_2$  respectively. Note that higher degree expansions (for e.g.,  $p > 2$ ) will require more support points from the adjacent subdomains in order to decode the MFA up to  $\partial\Omega_{1,2}$ . This implies that  $P_i(\Delta_i)$  directly provides a measure of the required cost of communication with adjacent subdomains.

In this scheme, the coupled data chunks,  $\vec{P}_2$  and  $\vec{P}_1$  for subdomains  $\Omega_1$  and  $\Omega_2$  respectively at  $\partial\Omega_{1,2}$  are exchanged simultaneously before the local domain solves are computed. One key advantage with such a DD scheme is that it only requires nearest neighbor exchange of data, which keeps communication costs bounded as number of subdomains increases [34, 17], while providing opportunities to interlace recomputation of the constrained control point solution. Note that in a RAS iterative scheme, nearest neighbor exchanges can be performed compactly per dimension and direction, thereby minimizing communication costs and eliminating expensive global collectives.

**2.2.1. Overlap: Augmenting Knot Spans.** One of the key metrics of interest is that the parallel solver infrastructure does not amplify any approximation errors unresolved by the tensor product NURBS or B-spline mesh. Since the local decoupled subdomain solution is encoded accurately (within user tolerance) in each individual subdomain without any data communication (i.e., embarrassingly parallel), imposing the constraints for the shared DoFs in  $\Delta$  should ensure the error change is bounded. However, as the control point data across subdomains become synchronized, numerical artifacts, especially for high-degree ( $p > 1$ ) basis reconstructions at subdomain interfaces can become dominant sources of error. A key metric to consider in all experiments is to validate that the multiple subdomain case produces the same error profile of a single subdomain case, in order to ensure convergence of the solvers to the same unique solution, independent of  $\mathcal{N}$ .

233 It is well known that overlapping DD methods using RAS and its variants [17] provide better  
 234 stability and convergence properties as both the solver and as preconditioner when used in the  
 235 context of accelerating PDE solvers. Likewise, we will use overlap regions  $\delta$ , as illustrated in  
 236 Fig. (2) for a 2D problem with  $p = 3$  and  $N = 4$ , for increase the size of the local problem ( $\Omega$ )  
 237 and to improve the accuracy of the domain decomposed approximation. We note that for the extra  
 238 augmented overlap regions  $\delta$ , the control point data  $\vec{P}$  are shared and uniformly weighted by the  
 239 solution to the neighboring subdomain minimization problem.

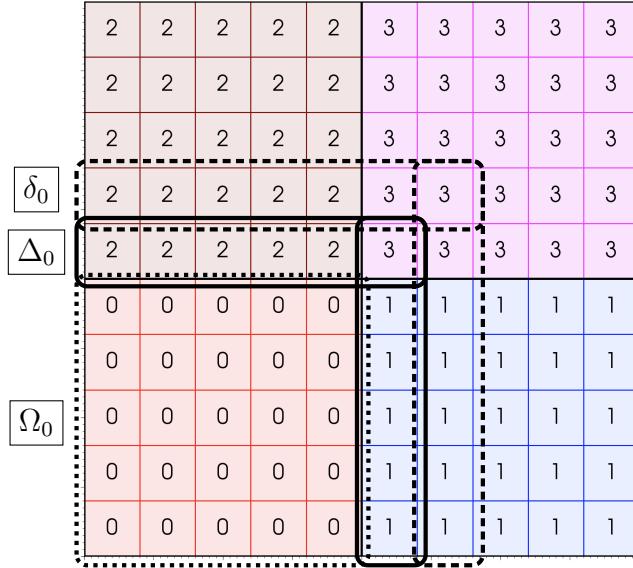


Fig. 2: 2D domain with  $N = 4$ ,  $p = 3$  and the augmented overlap  $|\delta| = 1$

240 **2.2.2. Note on Performance Characteristics.** The volume of messages exchanged between  
 241 subdomains depends on several computational factors.

- 242 1. **Clamping:** If the boundary knots are pinned, or if they have a floating knot description  
     at subdomain interfaces depending on whether  $C^0$  or  $C^{p-1}$  continuity is required,
- 243 2. **Parity:** Whether the MFA degree of expansion is odd or even, which determines the range  
     of common knot spans shared between adjacent domains as given by  $\Delta$  (refer to Fig. (1)),
- 244 3. **Overlap:** The amount of augmented overlap ( $\delta$ ), which determines the number of addi-  
     tional coupled data layers to be communicated between neighboring domains, both in terms  
     of the input span space  $\vec{Q}$ , and control point DoFs  $\vec{P}$  (refer to Fig. (2)).

245 At convergence, the interface data at  $\partial\Omega_{1,2}$  will satisfy the higher-order continuity prescriptions  
 246 specified by the user, thereby guaranteeing full regularity of  $C^{p-1}$ . The illustration in Fig. (1), and  
 247 the methodology description in this section can be generalized and extended to arbitrary dimen-  
 248 sions in the tensor-product setting (with the parametric domain represented by a  $d$ -dimensional  
 249 hypercube), and will serve as the basis to describe the local subdomain solvers in the following sub-  
 250 sections. Using knot insertion and removal strategies, individual subdomains can also be adapted  
 251 to resolve fast varying solutions and to reduce decoded error to be within user-specified tolerances.

256 While adaptivity has not been fully explored in the current work, enabling variable resolutions  
 257 in different dimensions is a natural extension of the work that will still preserve  $C^p$  continuity.  
 258 The implementation of the presented approach with domain decomposition strategies, combined  
 259 with overlapping RAS scheme yields a scalable scheme that will be demonstrated to be suitable for  
 260 tackling large-scale data analysis problems.

261 We next present the parallel MFA computation workflow that will be used with domain-  
 262 decomposed subdomain partitions.

263 **2.3. Solver Workflow.** Computing the functional approximation to large-scale datasets re-  
 264 quires efficient solvers at two levels: firstly, the local decoupled subdomain solver for Equation (2.3),  
 265 and next, the constrained minimization problem in Equation (2.4). Hence, the global problem re-  
 266 duces to a series of local minimization problems in each subdomain.

$$267 \quad (2.7) \quad \begin{aligned} & \arg \min_{\vec{P} \in \mathbb{R}^n} \left\| \vec{Q}_\ell - R_\ell \vec{P}_\ell \right\|_{L_2}, \quad R_\ell \in \mathbb{R}^{m \times n}, \vec{Q}_\ell \in \mathbb{R}^m, \vec{P}_\ell \in \Omega_i \\ & \text{subject to } \sum_{\partial\Omega_{i,j}} \left[ \mathcal{F}_{ij}(\vec{P}_i(\Delta_i), \vec{P}_j(\Omega_j)) + \mathcal{F}_{ij}(\vec{P}_i(\delta_i), \vec{P}_j(\Omega_j)) \right]^2 = 0, \quad \forall i, j \in [1, \dots, \mathcal{N}] \end{aligned}$$

268 where  $\Omega_j$  are the neighboring subdomains of  $\Omega_i$ ,  $\mathcal{F}_{ij}(a, b)$  is the jump term across the shared  
 269 interface DoFs  $a$  and  $b$  defined on subdomains  $i$  and  $j$  respectively.

270 **2.3.1. Subdomain Solvers.** For the linear LSQ solvers that can be used to compute local  
 271 subdomain control point solution  $\vec{P}$ , there are a variety of choices available. Direct methods like  
 272 Singular Value decomposition or Cholesky decomposition operating on the normal equations [3] can  
 273 compute optimal values. Alternatively, the iterative LSQ solvers such as orthogonal decomposition  
 274 methods based on QR and QZ factorizations are more stable, especially when the normal form of  
 275 the operator,  $R^T R$ , is ill-conditioned.

276 **2.3.2. Restricted Additive-Schwarz Solvers.** The outer RAS iterations work together  
 277 with nearest neighbor communication procedures to exchange shared DoF data between adjacent  
 278 subdomains. This is an important step to ensure that  $\vec{P}$  data computed through the LSQ procedure  
 279 are consistent and high-order continuous across subdomain boundaries. The final minimized control  
 280 point solution is achieved when the interface solutions match on all  $\partial\Omega_{i,j} \in \Omega$  rendering zero jump  
 281 residuals ( $\mathcal{F}_{ij}$ ) on  $\Delta$  and  $\delta$  shared domains in Equation (2.7).

282 It is also important to note that unlike the blending approaches that can be directly applied on  
 283 decoded data [18], the numerical error with this constrained iterative scheme is not bounded by the  
 284 original partitioned, unconstrained least-squares solution; i.e., imposing boundary constraints does  
 285 not create artificial numerical pinning of the control point data as we converge towards continuity  
 286 recovery. A solution to address this issue is to increase the amount of overlap range to ensure uniform  
 287 convergence to the true single-subdomain solution error, even as the number of subdomains ( $\mathcal{N}$ )  
 288 increases.

289 The nonoverlapping and overlapping RAS scheme applied to the computation of MFA exhibits  
 290 scalable convergence properties in the limit of decreasing subdomain size (i.e., as  $\mathcal{N} \rightarrow \infty$ ). This  
 291 is a favorable property for strong scaling, especially when tackling large datasets, as the net com-  
 292 putational cost always remains bounded. This behavior can be explained by the nature of how the  
 293 RAS iterative procedure resolves the shared DoFs.

294 By using a weighted averaging procedure for all shared DoFs that reside in the  $\Delta$  and  $\delta$  domains,

295 each outer iteration resolves any disparity in  $\vec{P}$ . The DoFs values on shared vertices ( $d > 0$ ), edges  
 296 ( $d > 1$ ) and faces ( $d > 2$ ) are resolved in the following order in consequent RAS iterations.

- 297    1. DoFs shared between two adjacent, neighboring subdomains; e.g., direct interface data  
 298        between say  $\Omega_0$  and  $\Omega_1$  in overlap regions  $\Delta_0 \cup \delta_0$  and  $\Delta_1 \cup \delta_1$  in Fig. (2),  
 299    2. DoFs shared by multiple (more than 2) neighboring subdomains; e.g., diagonal corners that  
 300        result from  $\mathcal{S}_0 \cap \mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3$  in Fig. (2), where  $\mathcal{S}_i = \Omega_i \cup \Delta_i \cup \delta_i$ .

301 Given both of these specific DoF groups, the overlapping RAS scheme applied to MFA computation  
 302 *always converges in 3 outer iterations*, which includes an iteration to perform convergence checks.

303 In the current study, a uniform weighting procedure has been used to converge the shared DoF  
 304 between different subdomains, where the weights for each shared DoF is assigned as  $w_i = \frac{1}{n_s}$ , where  
 305  $n_s$  is the number of subdomains containing DoF  $i$  within its domain  $\mathcal{S}$ . This weighing procedure can  
 306 be trivially replaced by Shephard's functions, especially in the context of adaptive discretizations  
 307 with variable knot displacements.

---

**Algorithm 2.1** Domain Decomposed MFA Solver
 

---

**Input:** Dataset and coordinates

**Parameters:**  $\mathcal{N}, p$

**Setup:** Decompose domain into blocks with DIY

**Solve:** Unconstrained local LSQ problem

**while** not converged **do**

$\vec{P}(\Omega_i \cap (\Delta_j + \delta_j)) \rightarrow$  enqueue outgoing constraints

    Exchange constraints with all nearest neighbor blocks

$\vec{P}(\Delta) \leftarrow$  dequeue incoming constraints

    Enforce constraints for  $\vec{P} \in \Delta$  and  $\delta$  domains

    Update local error  $E_i := \|\vec{Q}_i - R\vec{P}_i\|_{L_2}$

    Check if converged

    Write MFA to disk for analysis and visualization

---

308    **2.4. Implementation.** The DD techniques presented here for MFA computation are primarily  
 309 written in Python, with main dependencies on SciPy for spline evaluations and linear algebra  
 310 routines. Additionally, the drivers utilize the Python bindings for the DIY [26] C++ library. DIY  
 311 is a programming model and runtime for block-parallel analytics on distributed-memory machines,  
 312 built on MPI-3 [15]. Rather than programming for process parallelism directly in MPI, the pro-  
 313 gramming model in DIY is based on block parallelism: data are decomposed into subdomains called  
 314 blocks; blocks are assigned to processing elements (processes or threads) and the computation is  
 315 described over these blocks, and communication between blocks is defined by reusable patterns.  
 316 The Python bindings to DIY utilize PyBind11 [20] and MPI4Py [10] to expose the interfaces in the  
 317 C++ library. In our implementation, PyDIY is exclusively used to manage the data decomposi-  
 318 tion, including specifications to share an interface  $\partial\Omega_{i,j}$  and ghost layers that represent the  $\Delta \cup \delta$   
 319 overlapping domains.

320    The overall approach is sketched in Algorithm (2.1). We begin by decomposing the domain  
 321 into a set of regular blocks aligned with the principal axes of the global domain. Before enforcing  
 322 constraints, the local subdomain solves are performed completely decoupled so that the discontinu-  
 323 ous MFA to represent the partitioned input data is computed. The control point solution from this  
 324 decoupled LSQ problem solver is then used as the DoF data that needs to be constrained with RAS

325 iterative method. We then begin iterating over the blocks to converge the shared DoFs through the  
 326 linear constraints described in Section (2.3).

327 At the start of each iteration, the control point constraints are exchanged between neighboring  
 328 blocks in a regular nearest-neighbor communication pattern. This is sufficient to update the con-  
 329 straints  $\vec{P}(\Delta \cup \delta)$ . DIY sends and receives the constraint data to neighboring blocks based on the  
 330 parallel data decomposition. The nonlinear residual error in each subdomain is a function of the  
 331 tensor product mesh resolution and degree  $p$ . At convergence, we expect to recover the subdomain  
 332 error that is identical to the single subdomain case.

333 The final result, as described in Algorithm (2.1), is a global MFA that retains high-order  
 334 continuity and accuracy of a single subdomain solve, but with excellent parallel efficiency to reduce  
 335 total time to solution as the number of subdomains increases.

336 **3. Results.** To demonstrate the effectiveness of the iterative algorithm for MFA computation,  
 337 we devised a series of analytical closed form functionals and utilized real-world scientific datasets  
 338 in both 2- and 3-dimensions obtained from high-fidelity simulations. All runs shown in this section  
 339 were performed using the Python drivers written specifically for this work using the DIY domain  
 340 decomposition infrastructure.

341 **3.1. 1-D Results.** In this section, detailed analysis on the convergence and accuracy of  
 342 various MFA continuity recovery approaches are presented.

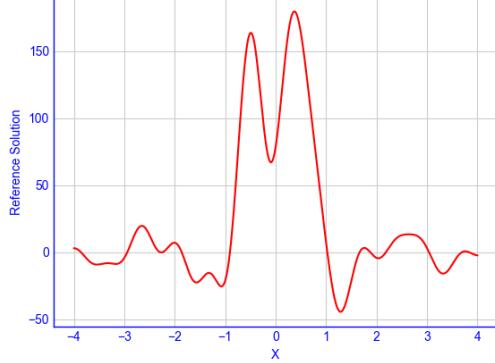
343 **3.1.1. Comparison of Clamped vs Floating Boundary Knots.** To demonstrate the  
 344 choice of using floating knots vs the low-order ( $C^0$ ) continuous clamped knots at subdomain bound-  
 345 aries, we choose an analytical closed form reference solution of the form:

$$346 \quad (3.1) \quad F(x) = \text{sinc}(x) + \text{sinc}(2x - 1) + \text{sinc}(3x + 1.5), \forall x \in \Omega = [-4, 4]$$

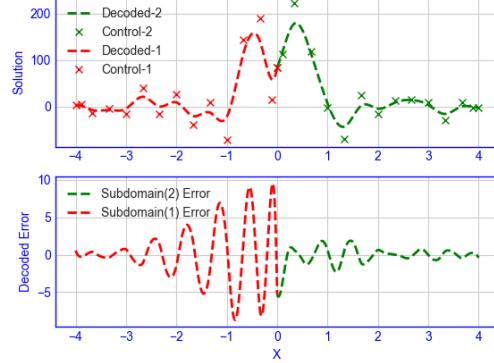
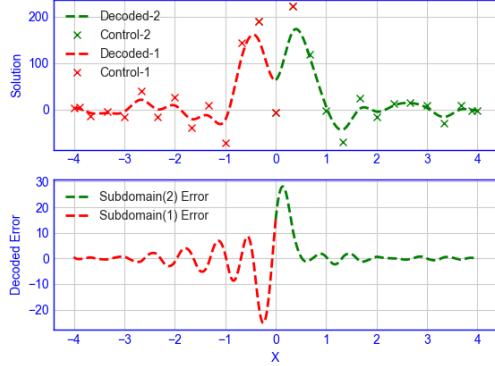
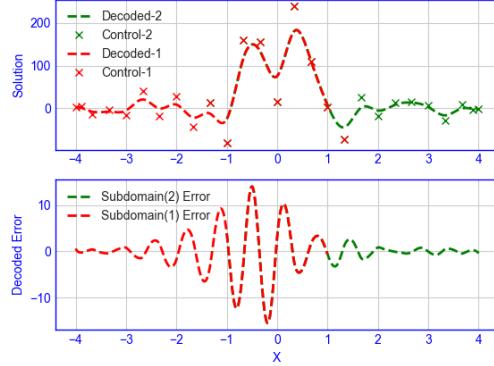
347 The reference solution  $F(x)$ , the results from the clamped knots, and floating knots with and  
 348 without augmented overlap regions  $\delta = p$  are shown in Fig. (3). The figures show the recovered  
 349 solutions and the corresponding decoded error from MFA evaluation for a  $\mathcal{N} = 2$  and  $p = 3$  case. It  
 350 is evident that the net error profile in the fully clamped subdivision in this example shows lower error  
 351 as compared to the floating knot experiments. However, it is imperative to note that the former  
 352 only shows  $C^0$  regularity, while the floating knots fully recover high-order continuity at subdomain  
 353 interfaces. Moreover, the use of augmented overlap regions ( $\delta = 3$ ) produce error profiles that  
 354 resemble a single subdomain error profile in the domain, which is one of the key metrics of interest.  
 355 These behaviors and conclusions extend to multi-dimensional setting as well.

356 To further demonstrate the continuity recovery behavior, we plot the error profile  $\vec{E}$  for these  
 357 approaches in Fig. (4). The unconstrained and decoupled LSQ solution procedure in the top  
 358 shows that the reconstructed solution is discontinuous at the interface, as expected. Using the  
 359 fully clamped approach that yields lower overall absolute error showcases only  $C^0$  continuity at the  
 360 interface, which may or not be sufficient depending on the use case utilizing the MFA representation.  
 361 Finally, the bottom plot shows the smooth error profile from using the floating knots at the interface  
 362 with full recovery of high-order continuity. We again emphasize that one could recover  $C^0$  to  $C^{p-1}$   
 363 continuity with this approach by choosing to use floating knots vs varying number of repeated knots  
 364 at the interface.

365 **3.1.2. Error Convergence and Overlap Experiments.** To determine the effect of using  
 366 augmented or overlapped knot span regions ( $\delta$ ) as the number of subdomains  $\mathcal{N}$  are increased, we



(a) Input analytical 1D solution profile

(b) Clamped  $C^0$  continuous decoded solution(c) Floating knots at  $\Omega_{1,2}$  recovering  $C^{p-1}$  continuity(d) Floating knots at  $\Omega_{1,2}$  with  $|\delta| = 3$ Fig. 3: 1-D analytical sinc dataset with 10000 input points with  $\mathcal{N} = 2$  and  $p = 3$ 

use a fully symmetric double-sinc function on a single subdomain as the reference solution as shown in Fig. (5a), and with  $\mathcal{N} = 5$  for different values of augmented spans ( $|\delta| = 0$  and  $|\delta| = 3$ ).

It is evident from Fig. (5c) that when there is no augmented knot spans used in a multi-subdomain solver, the decoding of data at subdomain boundaries are influenced by contributions from both adjacent domain DoFs, which are enforced to be  $C_{p-1}$  continuous by the constrained minimization solver. However, as we increase the number of overlap regions in terms of both the underlying data and the local bases support spans, the error profiles as shown in Fig. (5d) approaches the reference profile (with  $\mathcal{N} = 1$ ) shown in Fig. (5b). Heuristically, for many of the problems tested, using  $|\delta| = p$  provides optimal convergence as number of subdomains increase, even though increasing this parameter to  $|\delta| = 2p$  or higher will in general always improve the

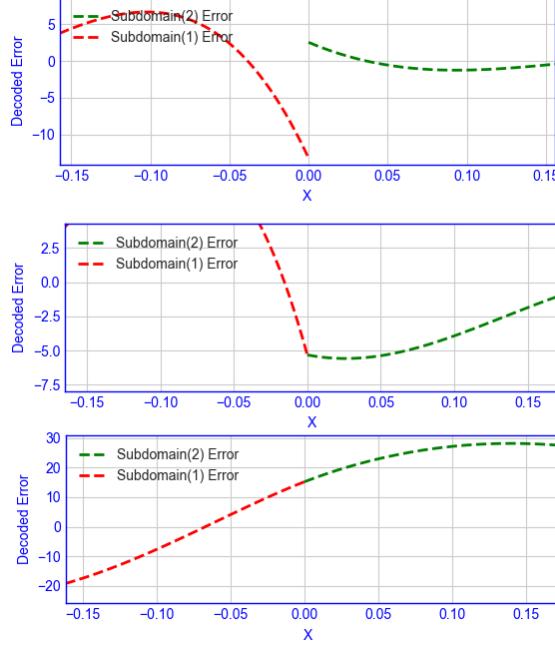


Fig. 4: Zoomed error plots at interface  $\Omega_{12}$  for 1-D analytical dataset with  $10^4$  input points with  $\mathcal{N} = 2$  and  $p = 3$ . Top: unconstrained and decoupled subdomains (discontinuous), middle: clamped interface knots, and bottom: floating interface knots.

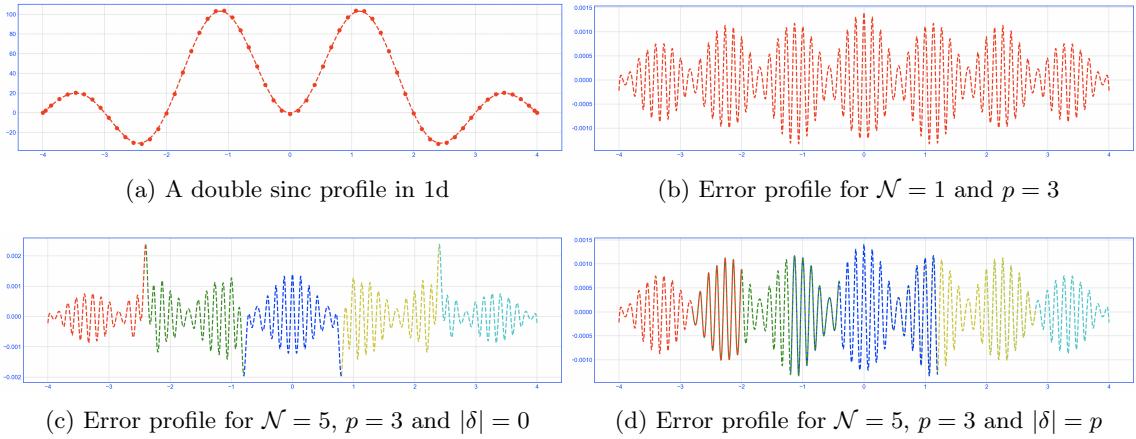


Fig. 5: Demonstration of error convergence, and effect of the overlapping spans to minimize numerical artifacts

numerical accuracy at the cost of higher communication costs between neighboring subdomains.

### 3.2. Multi-dimensional Problem Cases.

**3.2.1. 2D Problem.** In order to verify the reconstruction of real world data, we choose a 3D Large-Eddy Simulation code (Nek5000 [14]) applied to a double-pipe flow problem. We take a 2D slice (with  $|\Omega| = 200 \times 200$ ) along the midplane in axial direction and use the dataset for our first test study here. The reference solution and the converged, reconstructed solution with  $\mathcal{N} = 5 \times 5 = 25$  subdomains with  $p = 6$  and  $|\delta| = 2p$  is shown in Fig. (6). The error norms are well converged with 20 floating control points per subdomain, yielding a net compression of 4x in representing the data with continuous derivatives everywhere in the domain  $\Omega$ .

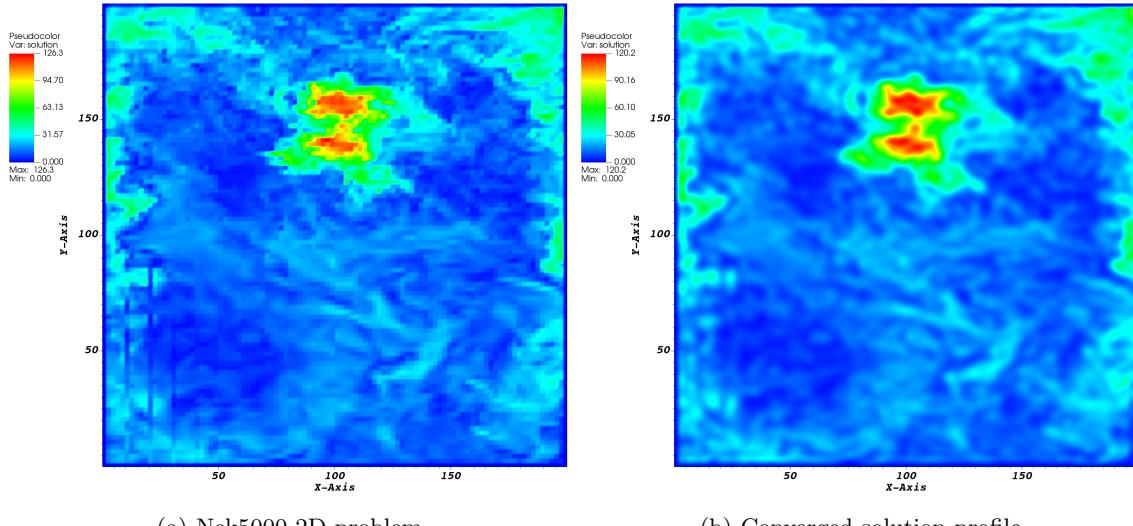


Fig. 6: 2-D slice of the Nek5000 3D dataset ( $200 \times 200$ ): reference profile and B-spline MFA with  $p = 6$  and  $\mathcal{N} = 5 \times 5$

**3.2.2. 3D Problem.** Next, we present S3D, a turbulent combustion dataset generated by an S3D simulation [7] of fuel jet combustion in the presence of an external cross-flow. The 3D domain has the span  $|\Omega| = 704 \times 540 \times 550$ , with the raw data containing the components of the vector field. We choose to use the magnitude of this velocity field in our reconstruction study shown below in Fig. (7). While uniform refinement in knot spans does yield sufficient error reductions in most subdomains, utilizing adaptive error resolution with knot insertions and removals for MFA as previously used here [27] for the S3D problem can provide better reconstructions in addition to the iterative scheme introduced here. This extension will be pursued in the future.

**3.2.3. Error Convergence.** We utilize synthetic datasets shown in Equation (3.3) to perform error convergence studies and to determine areas of maximal error that iteratively are resolved between neighboring subdomains.

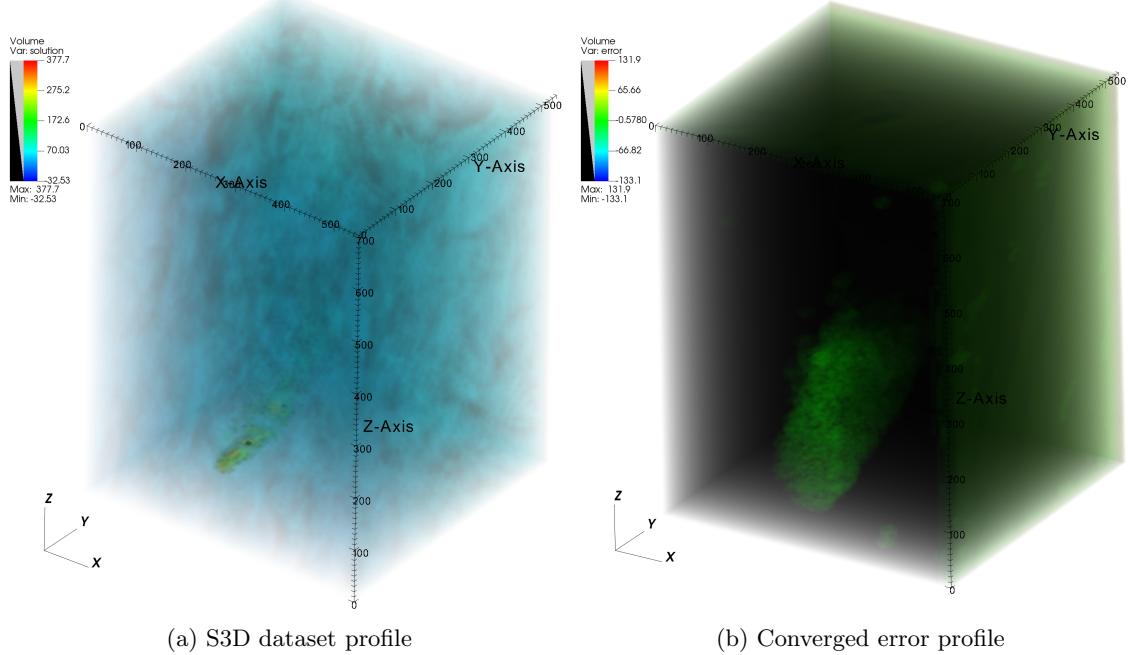


Fig. 7: Volume rendered S3D dataset: profile and error resolution with  $\mathcal{N} = 4 \times 4 \times 4 = 64$ ,  $p = 4$  and  $|\delta| = 4$

$$(3.2) F(x, y) = \text{sinc}(\sqrt{x^2 + y^2}) + \text{sinc}(2(x - 2)^2 + 2(y + 2)^2), \forall (x, y) \in \Omega = [-4, 4]^2,$$

$$(3.3) g(x, y, z) = \text{sinc}(\sqrt{x^2 + y^2 + z^2}) + \text{sinc}(2(x - 2)^2 + (y + 2)^2 + (z - 2)^2), \forall (x, y, z) \in \Omega = [-4, 4]^3$$

We plot the change in error between subsequent iterations of the RAS scheme for both the 2D and 3D problem cases, with  $|\delta| = 0$  in Fig. (8). This clearly demonstrates that the interface values between two neighboring subdomains are resolved first, and DoFs that are shared by multiple subdomains are resolved next. In all cases, the iterations converge in 3 steps, independent of  $\mathcal{N}$  or  $\delta$ .

**3.2.4. Parallel Scalability.** To demonstrate the parallel performance of the implemented RAS iterative scheme for MFA computation with continuity preservation, we employ both closed-form synthetic and real simulation datasets in 2D and 3D. In the following sections, we present both strong scaling and weak scaling studies performed on the Theta Cray XC40 supercomputer operated by the Argonne Leadership Computing Facility (ALCF), which provides 4,392 KNL compute nodes with 64 compute cores and 192 GB DDR4 RAM per node. The interconnect is based on the Aries Dragonfly high speed network.

**Strong Scaling.** We consider both 2D and 3D problems to demonstrate the scaling behavior of the presented parallel MFA computational algorithm in Algorithm (2.1). One key consideration

413 that drove selection of the subdomain sizes, and the floating knot span descriptions, is motivated  
 414 by the metric to recover the original error profile from a single subdomain case. Verification studies  
 415 were performed during this strong scaling test to ensure that the local subdomain errors computed  
 416 on a single task, and on different process counts remain the same at convergence. This verification  
 417 is important to reiterate the fact that the approximation error due to the constrained solves to  
 418 recover higher-order continuity does not significantly affect the error metrics for the MFA as  $\mathcal{N}$   
 419 increases. For this reason, we used synthetic datasets generated with closed form equations for 2D  
 420 and 3D studies as shown in Equation (3.3).

421 The 2D strong scaling tests were performed on tasks in the span of [1,16384] increasing by a  
 422 factor of  $2^2 = 4$ , and the 3D tests use tasks [1,32768] increasing by a factor of  $2^3 = 8$ . In order to  
 423 also better understand the effects of using augmented overlap regions ( $\delta$ ) on scalability, two cases  
 424 with the choice of  $|\delta| = 0$  and  $|\delta| = p$  are shown in Fig. (9). The Python driver utilized DIY to  
 425 handle block decompositions and rank assignments, as the total number of tasks used in the parallel  
 426 run was increased. We measured the overall computational time for setting up the problem, the  
 427 initial subdomain solves, and the consequent RAS iteration cycle to convergence, which includes  
 428 the nearest neighbor communication at each iteration. We also show the time for decoding the  
 429 MFA that is used to measure the errors in each subdomain and the overall total that includes the  
 430 effort spent on each of these various tasks. This task-wise breakdown helps us clearly visualize the  
 431 steps that scale linearly.

432 As expected, the RAS iterative scheme shows excellent scalability for the chosen datasets, and  
 433 the overall time to compute the MFA in parallel as  $\mathcal{N}$  increases, was reduced at a nearly ideal rate  
 434 upto  $10^4$  MPI tasks, while ensuring  $C^{p-1}$  continuity in the subdomain interfaces. The effect of  
 435 using augmented overlaps has an *insignificant effect* on the overall scaling efficiency of the solution  
 436 method. It is important to note that the dominant computational time is driven by the decoupled  
 437 LSQ solution computation and decoding operations, which are embarrassingly parallel as the size  
 438 of the subdomains (determined by number of knot spans) decrease in direct proportion to the  
 439 tasks. Given that the scalability of the linear algebraic LSQ solvers [2, 11] and Sparse Matrix  
 440 Vector (SpMV) products used in the decode tasks are well understood, the bottlenecks potentially  
 441 occur only from the nearest neighbor communication for constraint data exchanges, which remain  
 442 significantly small in magnitude up to  $10^4$  tasks tested. The overall strong scaling efficiency remain  
 443 above 70% for both overlapping and non-overlapping 2D problem cases. However, the added setup  
 444 cost and nearest neighbor communication reduce the 3D problem efficiency for the overlapping  
 445 subdomain cases to 40% at 32768 tasks from 85% in the nonoverlapping cases. One reason for this  
 446 efficiency degradation behavior in augmented runs is due to the higher setup cost, which involves  
 447 computing participating intervals in both the input space ( $\Omega$ ) and extra knot spans ( $\delta$ ) that need  
 448 to be exchanged with neighboring subdomains.

449 **Weak Scaling.** Given that the performance of the overlapping and augmented MFA scheme  
 450 was comparable to non-overlapping cases ( $|\delta| = 0$ ), and since the error reduction from having extra  
 451 overlaps always results in better solution recovery, we strictly focus on overlapping cases alone for  
 452 the weak scaling study. Here, the overall work per subdomain is maintained constant, and the  
 453 number of tasks are increased from 1 to 16384 in 2D, and from 1 to 32768 in 3D, similar to the  
 454 strong scaling study. The weak scaling results maintaining an overall MFA compression rate of  
 455  $2^d, \forall d \in [2, 3]$  is shown in Fig. (10).

456 The weak scaling study demonstrates that the overall performance of the RAS iterative scheme  
 457 for large number of subdomains does not significantly affect the parallel efficiency, which are around  
 458 40% in 2D and 24% in 3D at the fine limit tested. The subdomain solve and nearest neighbor data

459 exchange dominates the overall time to solution. However, it is important to note that the actual  
 460 runtime for the MFA computation only grows by a factor of 2, even on 16K processes or more.

461 **Performance Study on S3D Dataset.** Finally, we consider the case of the S3D combustion  
 462 dataset shown in Fig. (7) and measure the strong scaling performance on upto 1024 processes. Using  
 463 parallel MPI-IO implemented with DIY, and exposed through the Python interface, a strong scaling  
 464 performance study was measured on this realistic dataset and shown in Fig. (11). The performance  
 465 and error analysis indicate good speedup to reduce overall time for MFA computation, until nearest  
 466 neighbor communication and data exchanges start dominating the overall workflow.

467 **4. Conclusions.** We have presented a scalable DD approach to tackle the issue of discontinuous  
 468 MFA representations when performing the computations in parallel. The Restricted Additive  
 469 Schwarz (RAS) method is a natural algorithmic fit for data analysis problems to create efficient  
 470 MFA solutions in parallel. Through the use of Schwarz-based iterative schemes, combined with con-  
 471 strained local subdomain LSQ solvers, the two-level iterative technique has been shown to be robust  
 472 in converging to the compressed functional representation of the given data, without sacrificing the  
 473 approximation accuracy measured on a single subdomain of equivalent control point resolution.  
 474 Combining NURBS-based adaptivity with a-posteriori error measures [27], and ensuring higher-  
 475 order continuity across block boundaries, a scalable infrastructure can be developed based on the  
 476 Algorithm (2.1).

477 The PyDIY based Python implementations for 1-, 2- and 3-dimensional problems have been  
 478 shown here to resolve complex solution profiles and gradient variations, even under decreasing  
 479 subdomain sizes run on large-scale datasets. Additionally, we have demonstrated that the use of  
 480 overlap layers  $\delta$  can definitely improve the overall MFA accuracy and with a slightly higher one-time  
 481 setup cost that gets amortized in the overall computation time. We have determined that for all  
 482 the problems tested,  $|\delta| = p$  to  $|\delta| = 2p$  is optimal in terms of error recovery and computational  
 483 cost even for 3D problems up to 32768 tasks.

484 The strong scalability of the algorithm was also demonstrated for a large 3D combustion dataset  
 485 with 209M data points. The presented iterative scheme provides very good strong scalability for  
 486 both 2D and 3D problems tested, and the parallel efficiency degrades only when the cost of nearest  
 487 neighbor subdomain data exchanges start to creep up beyond the cost of the local constrained  
 488 subdomain solve. Given that scaling characteristics of these processes are well understood in the  
 489 literature, the parallel speedups behave predictably well at scale on large computing machines  
 490 tested.

491 Another natural way to ensure continuity across adaptively resolved NURBS or B-spline patches  
 492 would be to use T-splines [32], which are specifically designed for merging higher-dimensional sur-  
 493 faces with non-matching knot locations. The presented RASM-based solver approach can be easily  
 494 extended to this adaptive case to impose constraints across subdomain patch boundaries, while  
 495 local constraints within each block can be imposed with appropriate T-spline basis modifications.

496 Within this infrastructure, we can also utilize a multilevel MFA representation that hierarchi-  
 497 cally refines the approximation at each level [31] by decreasing the number of subdomains used,  
 498 similar to ideas in standard multilevel methods [6]. Such computations involving multilevel MFA  
 499 such as the ones using hierarchical B-splines [5] can significantly reduce the computational cost  
 500 of the local subdomain solvers, which dominate the total computational time. With appropriate  
 501 choices of prolongation and restriction operators, the RAS iterative scheme can be used with a  
 502 multilevel subdomain solver to efficiently produce accurate and compact functional approximation  
 503 of given data, especially in higher dimensions.

- [1] R. K. BEATSON, W. LIGHT, AND S. BILLINGS, *Fast solution of the radial basis function interpolation equations: Domain decomposition methods*, SIAM Journal on Scientific Computing, 22 (2001), pp. 1717–1740.
- [2] M. BENZI, *Preconditioning techniques for large linear systems: a survey*, Journal of computational Physics, 182 (2002), pp. 418–477.
- [3] Å. BJÖRCK, *Numerical methods for least squares problems*, SIAM, 1996.
- [4] P. E. BJØRSTAD AND O. B. WIDLUND, *To overlap or not to overlap: A note on a domain decomposition method for elliptic problems*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 1053–1061.
- [5] P. BORNEMANN AND F. CIRAK, *A subdivision-based implementation of the hierarchical b-spline finite element method*, Computer Methods in Applied Mechanics and Engineering, 253 (2013), pp. 584–598.
- [6] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of computation, 31 (1977), pp. 333–390.
- [7] J. H. CHEN, A. CHOUDHARY, B. DE SUPINSKI, M. DEVRIES, E. R. HAWKES, S. KLASKY, W.-K. LIAO, K.-L. MA, J. MELLOR-CRUMMEY, N. PODHORSZKI, ET AL., *Terascale direct numerical simulations of turbulent combustion using s3d*, Computational Science & Discovery, 2 (2009), p. 015001.
- [8] J. A. COTTRELL, T. J. HUGHES, AND Y. BAZILEVS, *Isogeometric analysis: toward integration of CAD and FEA*, John Wiley & Sons, 2009.
- [9] L. DALCIN, N. COLLIER, P. VIGNAL, A. CÓRTES, AND V. M. CALO, *Petiga: A framework for high-performance isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 308 (2016), pp. 151–181.
- [10] L. D. DALCIN, R. R. PAZ, P. A. KLER, AND A. COSIMO, *Parallel Distributed Computing using Python*, Advances in Water Resources, 34 (2011), pp. 1124–1139.
- [11] L. D. DALCIN, R. R. PAZ, P. A. KLER, AND A. COSIMO, *Parallel distributed computing using python*, Advances in Water Resources, 34 (2011), pp. 1124–1139.
- [12] C. DE BOOR AND R. DEVORE, *Approximation by smooth multivariate splines*, Transactions of the American Mathematical Society, 276 (1983), pp. 775–788.
- [13] L. DEDÈ AND A. QUARTERONI, *Isogeometric analysis for second order partial differential equations on surfaces*, Computer Methods in Applied Mechanics and Engineering, 284 (2015), pp. 807–834.
- [14] M. O. DEVILLE, P. F. FISCHER, AND E. H. MUND, *High-order methods for incompressible fluid flow*, vol. 9, Cambridge university press, 2002.
- [15] J. DONGARRA ET AL., *MPI: A Message-Passing Interface Standard Version 3.0*, High Performance Computing Center Stuttgart (HLRS), (2013).
- [16] W. DORNISCH AND S. KLINKEL, *Boundary conditions and multi-patch connections in isogeometric analysis*, PAMM, 11 (2011), pp. 207–208.
- [17] E. EFSTATHIOU AND M. J. GANDER, *Why restricted additive schwarz converges faster than additive schwarz*, BIT Numerical Mathematics, 43 (2003), pp. 945–959.
- [18] I. GRINDEANU, T. PETERKA, V. S. MAHADEVAN, AND Y. S. NASHED, *Scalable, high-order continuity across block boundaries of functional approximations computed in parallel*, in 2019 IEEE International Conference on Cluster Computing (CLUSTER), IEEE, 2019, pp. 1–9.
- [19] C. HOFER AND U. LANGER, *Fast multipatch isogeometric analysis solvers*, (2018).
- [20] W. JAKOB, J. RHINELANDER, AND D. MOLDOVAN, *pybind11—Seamless Operability Between C++ 11 and Python*, 2017.
- [21] M. KAPL, G. SANGALLI, AND T. TAKACS, *Construction of analysis-suitable g1 planar multi-patch parameterizations*, Computer-Aided Design, 97 (2018), pp. 41–55.
- [22] J. LI AND Y. HON, *Domain decomposition for radial basis meshless methods*, Numerical Methods for Partial Differential Equations: An International Journal, 20 (2004), pp. 450–462.
- [23] P.-L. LIONS, *On the schwarz alternating method. i*, in First international symposium on domain decomposition methods for partial differential equations, vol. 1, Paris, France, 1988, p. 42.
- [24] N. MAI-DUY AND T. TRAN-CONG, *Approximation of function and its derivatives using radial basis function networks*, Applied Mathematical Modelling, 27 (2003), pp. 197–220.
- [25] F. MARINI, *Parallel additive schwarz preconditioning for isogeometric analysis*, (2015).
- [26] D. MOROZOV AND T. PETERKA, *Block-Parallel Data Analysis with DIY2*, in Proceedings of the 2016 IEEE Large Data Analysis and Visualization Symposium LDAV’16, Baltimore, MD, USA, 2016.
- [27] Y. S. NASHED, T. PETERKA, V. MAHADEVAN, AND I. GRINDEANU, *Rational approximation of scientific data*, in International Conference on Computational Science, Springer, 2019, pp. 18–31.
- [28] K. PAUL, C. ZIMMERMANN, T. X. DUONG, AND R. A. SAUER, *Isogeometric continuity constraints for multi-patch shells governed by fourth-order deformation and phase field models*, Computer Methods in Applied Mechanics and Engineering, 370 (2020), p. 113219.

- 561 [29] T. PETERKA, S. YOUSSEF, I. GRINDEANU, V. S. MAHADEVAN, R. YEH, X. TRICOCHE, ET AL., *Foundations of*  
 562 *multivariate functional approximation for scientific data*, in 2018 IEEE 8th Symposium on Large Data  
 563 Analysis and Visualization (LDAV), 2018, pp. 61–71.
- 564 [30] L. PIEGL AND W. TILLER, *The NURBS book*, Springer Science & Business Media, 2012.
- 565 [31] D. SCHILLINGER, J. A. EVANS, A. REALI, M. A. SCOTT, AND T. J. HUGHES, *Isogeometric collocation: Cost*  
 566 *comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations*, Computer  
 567 Methods in Applied Mechanics and Engineering, 267 (2013), pp. 170–232.
- 568 [32] T. W. SEDERBERG, D. L. CARDON, G. T. FINNIGAN, N. S. NORTH, J. ZHENG, AND T. LYCHE, *T-spline simplifi-*  
 569 *cation and local refinement*, ACM transactions on graphics (TOG), 23 (2004), pp. 276–283.
- 570 [33] B. SMITH, P. BJORSTAD, AND W. GROPP, *Domain decomposition: parallel multilevel methods for elliptic partial*  
 571 *differential equations*, Cambridge university press, 2004.
- 572 [34] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive*  
 573 *schwarz preconditioning*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2402–2425.
- 574 [35] S. XU, W. JAHN, AND J.-D. MÜLLER, *Cad-based shape optimisation with cfd using a discrete adjoint*, Interna-  
 575 *tional Journal for Numerical Methods in Fluids*, 74 (2014), pp. 153–168.
- 576 [36] R. YOKOTA, L. A. BARBA, AND M. G. KNEPLEY, *Petrbf—a parallel o (n) algorithm for radial basis func-*  
 577 *tion interpolation with gaussians*, Computer Methods in Applied Mechanics and Engineering, 199 (2010),  
 578 pp. 1793–1804.
- 579 [37] X. ZHANG, Y. WANG, M. GUGALA, AND J.-D. MÜLLER, *Geometric continuity constraints for adjacent nurbs*  
 580 *patches in shape optimisation*, in ECCOMAS Congress, vol. 2, 2016, p. 9316.
- 581 [38] W. ZHENG, P. BO, Y. LIU, AND W. WANG, *Fast b-spline curve fitting by l-bfgs*, Computer Aided Geometric  
 582 Design, 29 (2012), pp. 448–462.

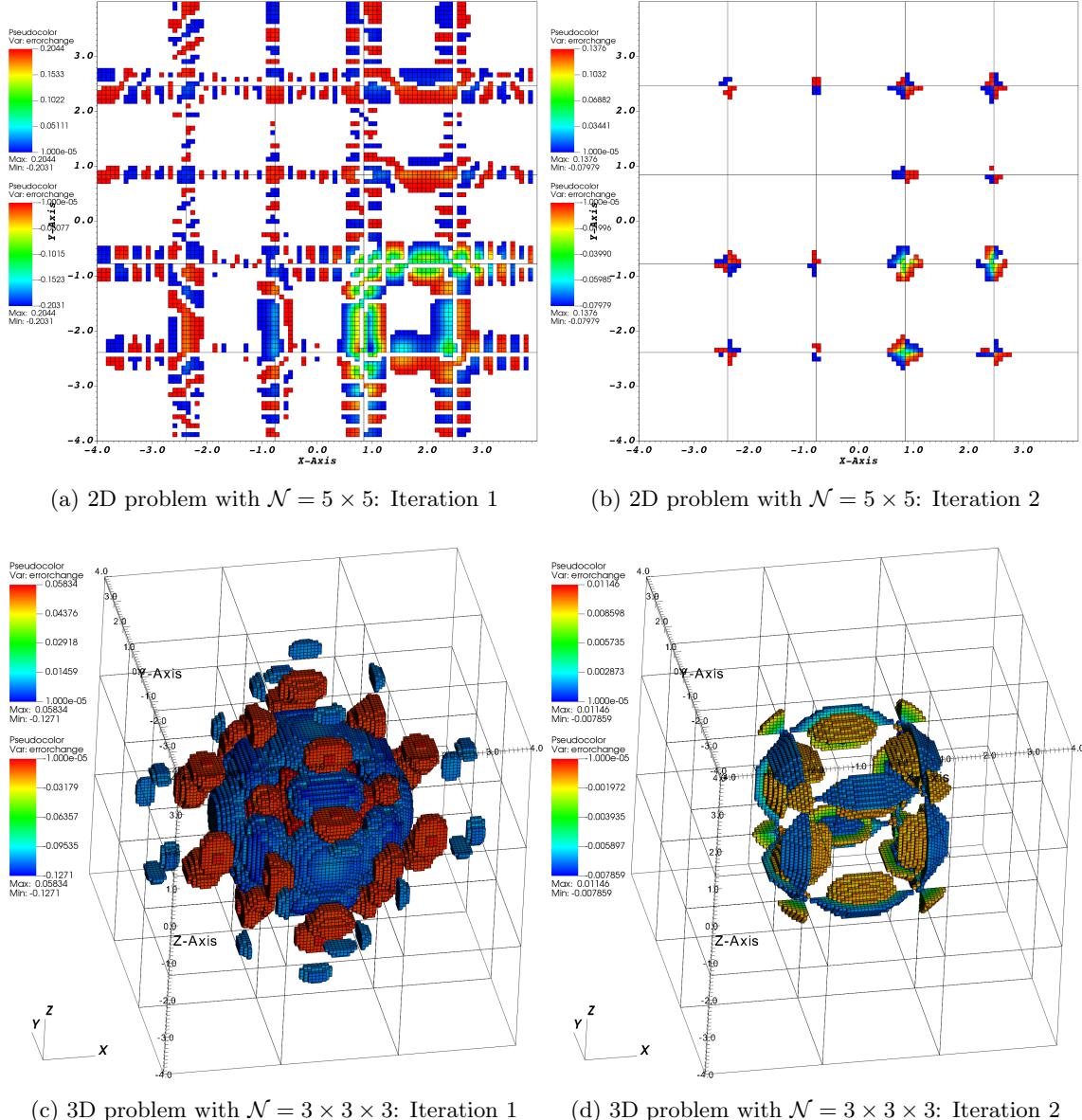
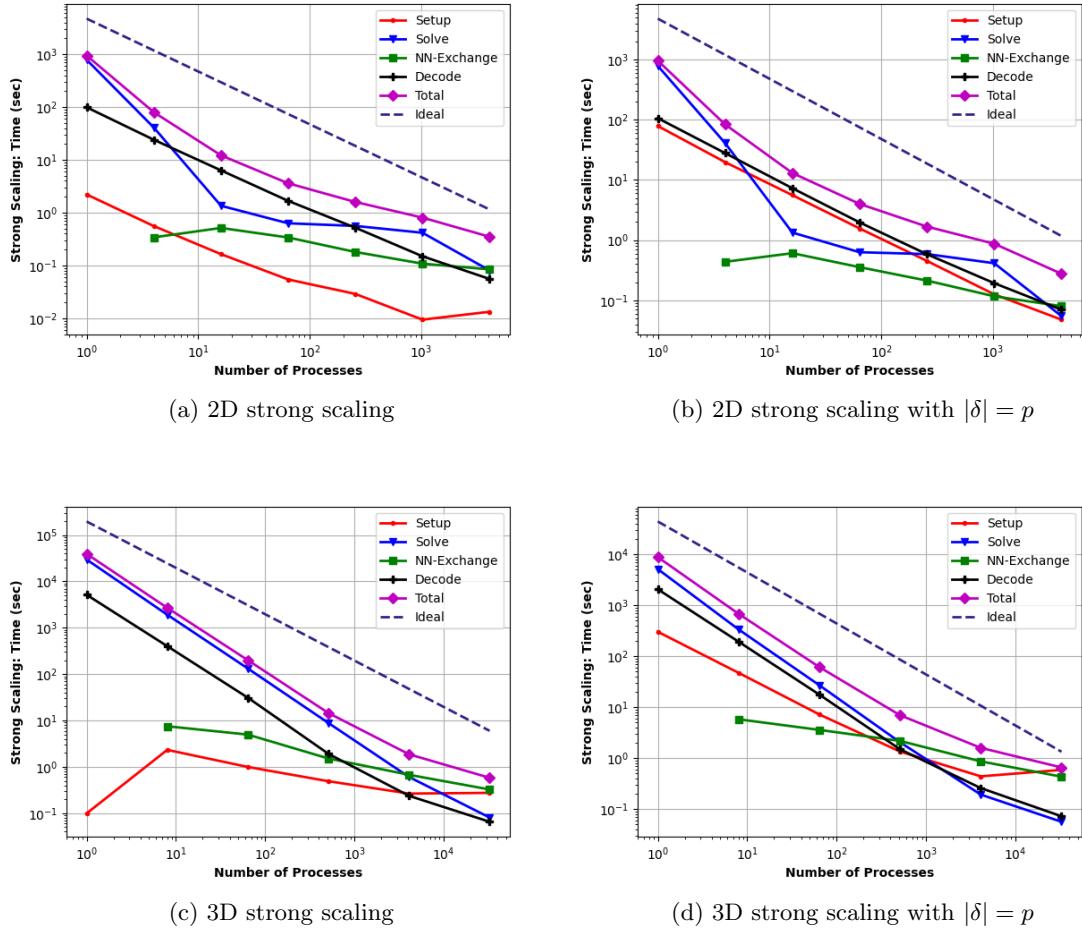
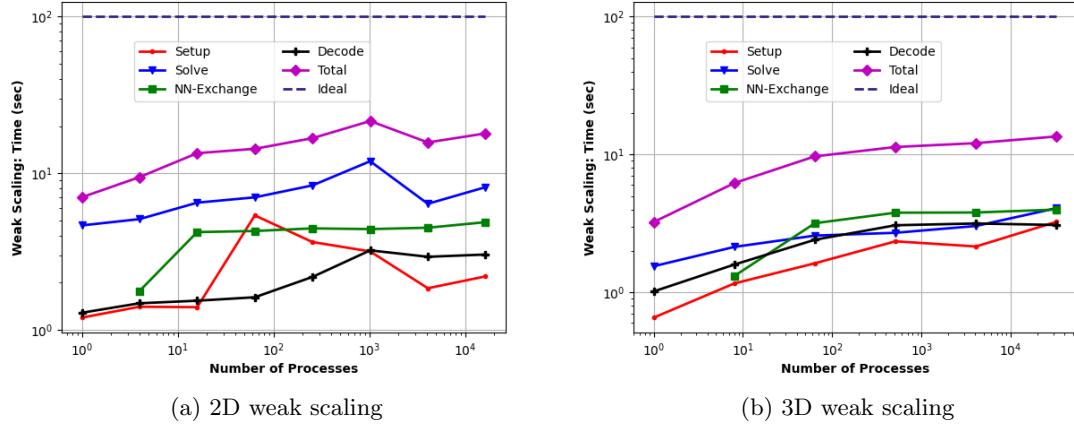
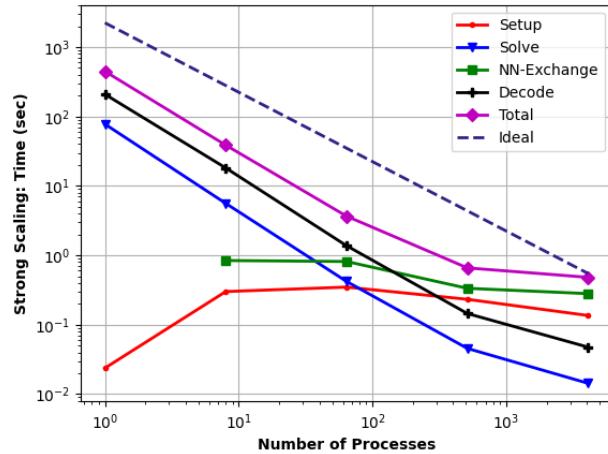


Fig. 8: Error convergence for 2D and 3D problems on the first and second iterate

Fig. 9: Strong scaling performance of RAS solver with  $p = 3$

Fig. 10: Weak scaling performance of RAS solver with  $p = 3$  and  $|\delta| = p$ Fig. 11: Strong scaling for the 3D S3D dataset with  $p = 5$