

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

3 VIJAY S. MAHADEVAN[†], DAVID LENZ[‡], IULIAN GRINDEANU[§], AND THOMAS PETERKA[¶]

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) [13] 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 [31] 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 decomposi-
38 tion (DD) techniques [34] with data partitioning strategies to produce scalable MFA computation
39 algorithms tha minimizes the reconstruction error when reproducing a given dataset. In such par-

*Submitted to editors .

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

[†]Argonne National Laboratory, Lemont, IL (mahadevan@anl.gov)

[‡]Argonne National Laboratory, Lemont, IL (dlenz@anl.gov)

[§]Argonne National Laboratory, Lemont, IL (iulian@anl.gov)

[¶]Argonne National Laboratory, Lemont, IL (tpeterka@mcs.anl.gov)

tioned analysis, it is imperative to ensure that the continuity of the encoded and decoded data across subdomain interfaces are maintained, and remain consistent with the degree of underlying expansion bases used in MFA [30]. This is due to the fact that independently computing MFA approximations in individual subdomains do not guarantee even C^0 regularity in either the MFA space or in the reconstructed data. In order to tackle this issue, we rely on an iterative Schwarz-type DD scheme to ensure that continuity is enforced, and the overall error stays bounded as the number of subdomains are increased (or as the subdomain size decreases).

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

Literature Review. Domain decomposition techniques in general rely on the idea of splitting a larger domain of interest into smaller partitions or subdomains, which results in coupled Degrees-of-Freedom (DoF) at their common interfaces. Typical applications of DD in Boundary-Value problems (BVP) [34, 24] have been successfully employed to efficiently compute the solution of large, discretized PDEs in a scalable manner. DD techniques for parallel approximation of scattered data have been explored previously with Radial Basis Functions (RBF) [25], yielding good scalability and closely recovering the underlying solution profiles. In general, overlapping multiplicative and additive Schwarz [35] iterative techniques for RBF [23] have proven successful to tackle large-scale problems. Additionally, the use of Restricted variants of Additive-Schwarz (RAS) method as preconditioners, with Krylov iterative solvers, can yield iterative schemes [37] with $O(N)$ computational complexity, as opposed to the typical $O(N \log(N))$ complexity with traditional RBF reconstructions [1]. The extensions of these ideas to NURBS bases exposes a way to fully parallelize traditional, serial MFA computations.

Combining the application of DD schemes and NURBS bases with isogeometric analysis (IGA) [8, 9] for high-fidelity modeling of nonlinear Partial Differential Equations (PDEs) [14, 26, 10] have enjoyed recent success at scale. However, many of these implementations lack full support to handle multiple geometric patches in a distributed memory setting due to non-trivial requirements on continuity constraints at patch boundaries. Directly imposing higher-order geometric continuity in IGA requires specialized parameterizations in order to preserve the approximation properties [22], which can be difficult to parallelize [20] generally. In a similar vein, using NURBS bases to compute the MFA in parallel, while maintaining higher-order continuity across subdomain patches has not been fully explored previously.

To overcome some of these issues with discontinuities along NURBS patches, Zhang et al. [38] proposed to use a gradient projection scheme to constrain the value (C^0), the gradient (C^1), and the Hessian (C^2) at a small number of test points for optimal shape recovery. Such a constrained projection yields coupled systems of equations for control point data for local patches, and results in a global minimization problem that needs to be solved.

Alternatively, it is possible to create a constrained recovery during the actual post-processing stage i.e., during the decoding stage of the MFA through standard blending techniques [19], in order to recover continuity in the decoded data. However, the underlying MFA representation remains discontinuous, and would become more so with increasing number of subdomains without the ability to recover higher-order derivatives along these boundaries. Moreover, selecting the amount of overlaps and resulting width of the blending region relies strongly on a heuristic, which can be problematic for general problem settings.

In contrast, we propose extensions to the constrained solvers used by Zhang et al. [38] and

Xu et al. [36], and introduce a two-level, DD-based, parallel iterative scheme to enforce the true degree of continuity, independent of p , unlike the low-order constraints used previously [38]. The outer iteration utilizes the RAS method [18], with efficient inner subdomain solvers that can handle linear Least-Squares systems to minimize the decoded residual within acceptable error tolerances. 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 presented 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, 10]. 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 DoFs. We will also introduce the idea of using open vs closed knots, which are clamped or floating 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 [31] 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 [13]. This property becomes especially important when performing analysis and in-situ visualization directly based on the NURBS representation of underlying data.

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
 128 efficient ℓ -BFGS scheme [39] 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 can be performed independently on each subdomain without dependencies as there are no
 131 constraints explicitly specified in Equation (2.3). However, in order to recover high-order continuity
 132 across subdomain patches, computing unconstrained solutions is insufficient. At a minimum, the
 133 DoFs on the shared subdomain boundaries have to be converged to recover C^0 continuity for the
 134 decoded solution data ($R\vec{P}$).

135 More generally, the constrained minimization problem to recover continuity [31] can be formulated as
 136

$$137 \quad (2.4) \quad R\vec{P} = \vec{Q} \mid \mathcal{C}\vec{P} = \vec{\mathcal{D}},$$

138 where \mathcal{C} is the constraint matrix imposing continuity restrictions on the control points \vec{P} along with
 139 its derivatives, with data exchanged from neighboring domains stored in \mathcal{D} , around the neighbor-
 140 hood of the interface $\partial\Omega_{i,j}$ shared by subdomains i and j . With the use of penalized constraints (\mathcal{C})
 141 and Lagrange multipliers [17, 29], the solution to the constrained LSQ problem can recover optimal
 142 control point values.

143 A straightforward approach to achieve C^0 continuity in the recovered solution is by ensuring
 144 that the common control point data \vec{P} at subdomain interfaces are clamped with repeated knots,
 145 in addition to using clamping at the global domain boundaries. In this scheme, the control points
 146 exactly interpolate (are clamped to) input data points at the subdomain interface boundaries. Such
 147 an approach requires in general a good spatial distribution of \vec{Q} , and yields only low-order contin-
 148 uous approximations (C^0 or C^1 at best) when the solution remains smooth across the subdomain
 149 interfaces. It should also be noted that as the number of subdomains increases, the global solu-
 150 tion being computed becomes further constrained, and more interpolatory due to clamped DoFs.
 151 Moreover, the MFA solution computed becomes dependent on the number of subdomains used to
 152 decompose the problem; i.e., the global control point data \vec{P} recovers different reconstructions as a
 153 function of number of subdomains (\mathcal{N}) used.

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

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

165 For the purpose of illustration and to explain the proposed solver methodology, let us consider
 166 a simple one dimensional domain (Ω) with two subdomains ($\mathcal{N} = 2$) as shown in Fig. (1), where Ω_1

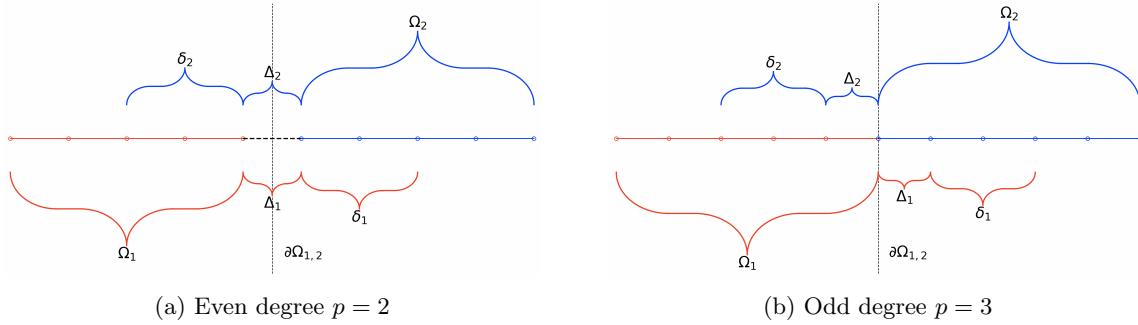


Fig. 1: Illustration: 1-D parallel partitioned domain with floating (unclamped) interior knots and augmented spans ($|\delta| = 2$)

and Ω_2 represent the subdomains that share an interface $\partial\Omega_{1,2}$. In Fig. (1), the layout of the 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 Δ_1 and Δ_2 on each subdomain that represents the set of shared knot spans with its adjacent subdomain (for internal boundaries), and an optional augmented layer δ_1 and δ_2 that has a connotation similar to that of an overlap region in traditional DD schemes [34]. Note that in order to reconstruct the input data in $\Omega_i, \forall i \in [1, 2]$, the knot spans must mandatorily include Δ_i regions. This Δ_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). For generality, Δ_i represents the repeated knots along clamped global domain boundaries, and the shared knots between two subdomains in the unclamped interior boundaries. For arbitrary degree p , the number of knot spans in Δ_i are $\frac{p-1}{2}$ and $\frac{p}{2}$ for odd and even p respectively. In multidimensional tensor product expansions, these shared spans are replaced by shared layers of knot spans along the subdomain interfaces. The δ_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 [18, 34].

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

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

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 [31]. 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

195 constrained equations in Equation (2.5) can be completely decoupled for each subdomain. This
 196 modified system resembles a block-Jacobi operator of the global system. The scheme illustrated in
 197 this section follows ideas similar to the Jacobi-Schwarz method [18] and the overlapping, restricted-
 198 Additive-Schwarz (RAS) scheme [35].

199 The control point DoF vector can be represented by three separate parts based on the local
 200 support of the basis expansion. The control point vector \vec{P} is in general given as $P =$
 201 $[\vec{P}(\Omega); \vec{P}(\Delta); \vec{P}(\delta)]$. For the 1D scenario illustrated here, this is shown below for a two subdo-
 202 main case with $p = 3$ and $|\delta| = 1$, where the operator $|\mathcal{D}|$ represents the number of knot spans in
 203 any underlying domain \mathcal{D} .

204 (2.6)

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

205
 206 where m, n are the number of control points in Ω_1 and Ω_2 respectively. Note that higher degree
 207 expansions (for e.g., $p > 3$) will require more support points in Δ from adjacent subdomains in
 208 order to decode the MFA up to $\partial\Omega_{1,2}$. This implies that $P_i(\Delta_i)$ directly provides a measure of the
 209 required cost of communication with adjacent subdomains.

210 In the above description, the coupled data chunks, $\vec{P}_1(\Delta_1)$ and $\vec{P}_2(\Delta_2)$ belonging to adjacent
 211 subdomains near $\partial\Omega_{1,2}$ are exchanged simultaneously before the local domain solves are computed.
 212 One key advantage with such a DD scheme is that only nearest neighbor exchange of data is
 213 required, which keeps communication costs bounded as the number of subdomains \mathcal{N} increase
 214 [35, 18], while providing opportunities to interlace recomputation of the constrained control point
 215 solution. Note that in a RAS iterative scheme, nearest neighbor exchanges can be performed
 216 compactly per dimension and direction, thereby minimizing communication costs and eliminating
 217 expensive global collectives.

218 **Overlap: Augmenting Knot Spans.** One of the key metrics of interest is that the parallel
 219 solver infrastructure does not amplify any approximation errors unresolved by the tensor product
 220 NURBS or B-spline mesh. Since the local decoupled subdomain solution is encoded accurately
 221 to satisfy Equation (2.3) in each individual subdomain without any data communication (i.e.,
 222 embarrassingly parallel), imposing the constraints for the shared DoFs in Δ should ensure the error
 223 change is bounded. However, as the control point data across subdomains become synchronized,
 224 numerical artifacts, especially for high-degree ($p > 2$) basis reconstructions at subdomain interfaces
 225 can become dominant sources of error. A key metric to consider in all experiments is to validate
 226 that the multiple subdomain case produces the same error profile as a single subdomain case, in
 227 order to ensure convergence of the solvers to the same unique solution, independent of \mathcal{N} .

228 For many problem domains, overlapping variants of Schwarz solvers [24, 18] have been proven
 229 to be more stable, efficient and scalable compared to nonoverlapping variants [4, 35]. We utilize
 230 the concept of overlap regions by sharing additional knot spans between subdomains in order to
 231 produce better MFA reconstructions of the underlying data. This user-specified, additional overlap
 232 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 conditioning in the subdomain solver, and the scalability of the overall algorithm. 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 better clarity, we will use overlap regions δ , as illustrated in Fig. (2) for a 2D problem with $p = 3$ and $N = 4$, to increase the size of the local problem (Ω), and to improve the accuracy of domain decomposed approximations. We note that the control point data \vec{P} in both the Δ and δ overlap regions are shared and uniformly weighted (averaged) by \vec{P} computed in the neighboring subdomains.

Note that in the 2D schematic, shared data in δ regions are always exchanged between neighboring subdomains and never locally computed. The set of $\vec{P}(\delta)$ are explicitly used only to impose constraints that contribute to the reconstruction of datasets. It is also important to note that when \vec{P} DoFs are *multishared* between subdomains, then shared data between multiple $\Omega_j, \forall j \in [0, 3]$ need to be exchanged in order to compute $\vec{P}_i(\delta_i), \forall i \in [0, 3]$.

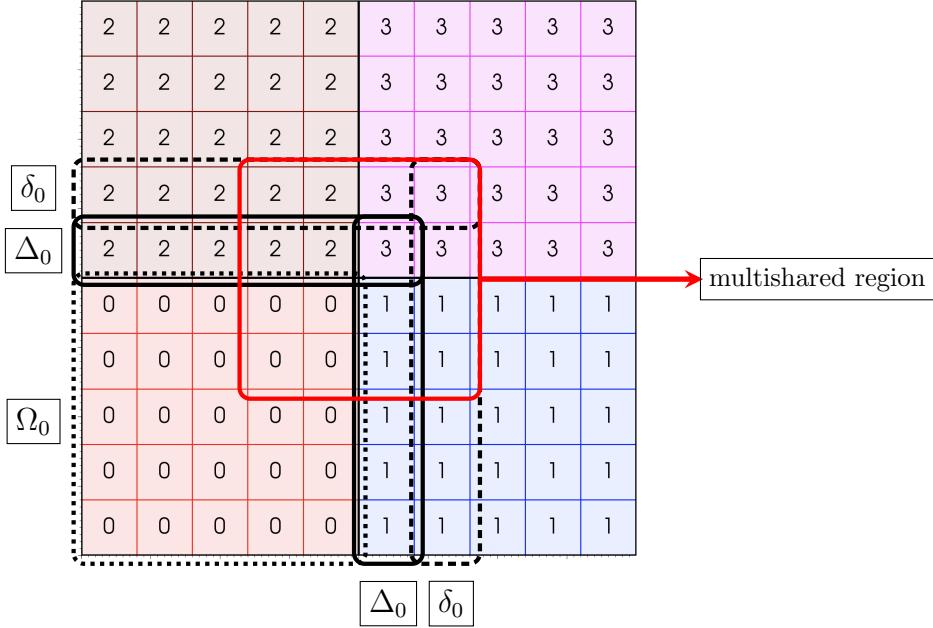


Fig. 2: 2D subdomains with $N = 4$, $p = 3$ and the augmented overlap $|\delta| = 1$ showing local subdomains Ω_i , mandatory overlap for floating knots Δ_i , optional overlap regions δ_i , where $i \in [0, N - 1]$, and multishared DoF regions that couple multiple subdomains (marked in red).

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

2.3. Solver Workflow. Computing the functional approximation to large-scale datasets requires efficient solvers at two levels: firstly, the local decoupled subdomain solver for Equation (2.3), and next, the constrained minimization problem in Equation (2.4). Hence, the global problem re-

duces to a series of local minimization problems in each subdomain.

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

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

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

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

It is also important to note that unlike the blending approaches that can be directly applied on decoded data [19], the numerical error with this constrained iterative scheme is not bounded by the original partitioned, unconstrained least-squares solution; i.e., imposing boundary constraints does not create artificial numerical pinning of the control point data as we converge towards continuity recovery. A solution to address this issue is to increase the amount of overlap range to ensure uniform convergence to the true single-subdomain solution error, even as the number of subdomains (\mathcal{N}) increases.

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

By using a weighted averaging procedure for all shared DoFs that reside in the Δ and δ domains, each outer iteration resolves any disparity in \vec{P} . The DoFs values on shared vertices ($d > 0$), edges ($d > 1$) and faces ($d > 2$) are resolved in the following order in consequent RAS iterations.

1. DoFs shared between two adjacent, neighboring subdomains; e.g., direct interface data between say Ω_0 and Ω_1 in overlap regions $\Delta_0 \cup \delta_0$ and $\Delta_1 \cup \delta_1$ in Fig. (2),
2. DoFs shared by multiple (more than 2) neighboring subdomains; e.g., diagonal corners that result from $\mathcal{S}_0 \cap \mathcal{S}_1 \cap \mathcal{S}_2 \cap \mathcal{S}_3$ in Fig. (2), where $\mathcal{S}_i = \Omega_i \cup \Delta_i \cup \delta_i$.

Given both of these specific DoF groups, the overlapping RAS scheme applied to MFA computation *always converges in 2 outer iterations* for 2D and 3D, and a single iteration in 1D. This result is demonstrated in Section (3).

In the current study, a uniform weighting procedure has been used to converge the shared DoF between different subdomains, where the weights for each shared DoF is assigned as $w_i = \frac{1}{n_s}$,

292 with n_s being the number of subdomains containing DoF i within its domain \mathcal{S} . This weighing
 293 procedure can be trivially replaced by Shephard's functions, especially in the context of adaptive
 294 discretizations with variable knot displacements.

295 **2.3.3. Note on Performance Characteristics.** The volume of messages exchanged between
 296 subdomains depends on several computational factors.

- 297 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,
- 299 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 Δ (refer to Fig. (1)),
- 301 3. **Overlap:** The amount of augmented overlap (δ), which determines the number of additional
 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. (1) and Fig. (2)).

304 At convergence, the interface data at $\partial\Omega_{1,2}$ will satisfy the higher-order continuity prescriptions
 305 specified by the user, thereby guaranteeing full regularity of C^{p-1} . The illustration in Fig. (1), and
 306 the methodology description in this section can be generalized and extended to arbitrary dimensions
 307 in the tensor-product setting (with the parametric domain represented by a d -dimensional hyper-
 308 cube) as shown in Fig. (2) for 2D. Using knot insertion and removal strategies based on deCasteljau
 309 subdivision procedures [31], individual subdomains can also be adapted to resolve fast varying solu-
 310 tions and to reduce decoded error to be within user-specified tolerances [28]. While adaptivity has
 311 not been fully explored in the current work, enabling variable resolutions in different dimensions is
 312 a natural extension of the work that will still preserve high-order continuity. The implementation
 313 of the presented approach with domain decomposition strategies, combined with overlapping RAS
 314 scheme yields a scalable scheme that will be demonstrated to be suitable for tackling large-scale
 315 data analysis problems.

Algorithm 2.1 Domain Decomposed MFA Solver

Input: Dataset and coordinates

Parameters: \mathcal{N}, p, m

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 δ 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

316 **2.4. Implementation.** The DD techniques presented here for MFA computation are primar-
 317 ily written in Python, with main dependencies on SciPy for spline evaluations and linear algebra
 318 routines. Additionally, the drivers utilize the Python bindings for the DIY [27] C++ library. DIY
 319 is a programming model and runtime for block-parallel analytics on distributed-memory machines,
 320 built on MPI-3 [16]. Rather than programming for process parallelism directly in MPI, the pro-

gramming model in DIY is based on block parallelism: data are decomposed into subdomains called blocks; blocks are assigned to processing elements (processes or threads) and the computation is described over these blocks, and communication between blocks is defined by reusable patterns. The Python bindings to DIY utilize PyBind11 [21] and MPI4Py [11] to expose the interfaces in the C++ library. In our implementation, PyDIY is exclusively used to manage the data decomposition, including specifications to share an interface $\partial\Omega_{i,j}$ and ghost layers that represent the $\Delta \cup \delta$ overlapping domains.

The overall approach is sketched in Algorithm (2.1). We begin by decomposing the domain into a set of regular blocks aligned with the principal axes of the global domain. Before enforcing constraints, the local subdomain solves are performed completely decoupled so that the discontinuous MFA to represent the partitioned input data is computed. The control point solution from this decoupled LSQ problem solver is then used as the DoF data that needs to be constrained with RAS iterative method. We then begin iterating over the blocks to converge the shared DoFs through the linear constraints described in Section (2.3).

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

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

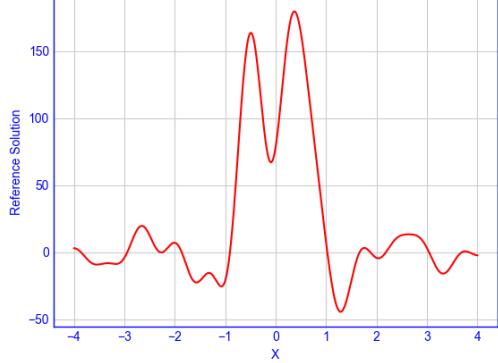
3. Results. To demonstrate the effectiveness of the iterative algorithm for MFA computation, we devised a series of analytical closed form functionals and utilized real-world scientific datasets in both 2- and 3-dimensions obtained from high-fidelity simulations. All runs shown in this section were performed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

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

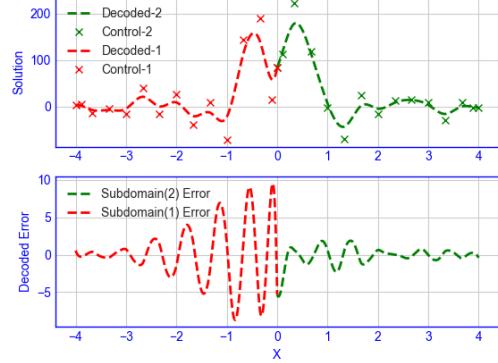
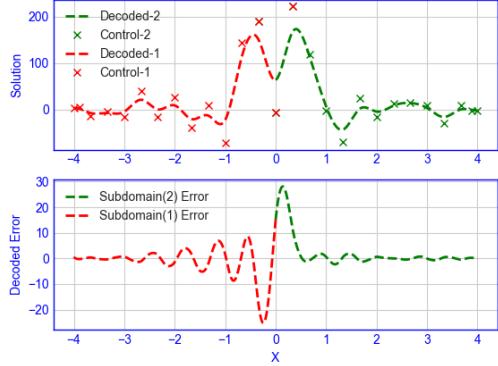
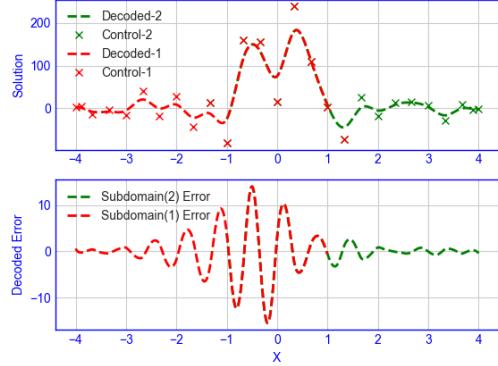
3.1.1. Comparison of Clamped vs Floating Boundary Knots. To demonstrate the choice of using floating knots vs the low-order (C^0) continuous clamped knots at subdomain boundaries, we choose an analytical closed form reference solution of the form:

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

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



(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: 1D analytical sinc dataset with 10000 input points with $\mathcal{N} = 2$ and $p = 3$

364 To further demonstrate the continuity recovery behavior, we plot the error profile \vec{E} for these
 365 approaches in Fig. (4), zoomed in around the interface $\Omega_{1,2}$. The unconstrained and decoupled
 366 LSQ solution procedure in the top shows that the reconstructed solution is discontinuous at the
 367 interface, as expected. Using the fully clamped approach that yields lower overall absolute error
 368 showcases only C^0 continuity at the interface, which may or not be sufficient depending on the
 369 use case utilizing the MFA representation. Finally, the bottom plot shows the smooth error profile
 370 from using the floating knots at the interface with full recovery of high-order continuity. We again
 371 emphasize that one could recover C^0 to C^{p-1} continuity with this approach by choosing to use
 372 floating knots vs varying number of repeated knots at the interface.

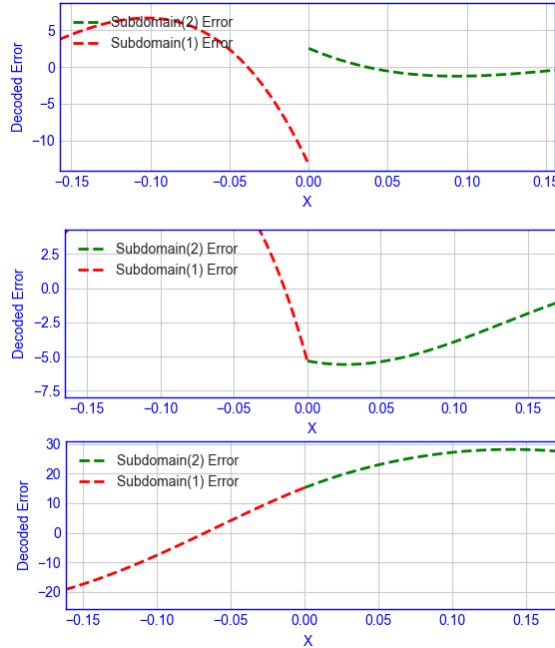


Fig. 4: Zoomed error plots at interface Ω_{12} for 1D analytical dataset with 10^4 input points with $\mathcal{N} = 2$ and $p = 3$. Top: unconstrained and decoupled subdomains (discontinuous), middle: clamped interface knots, and bottom: floating interface knots.

3.1.2. Error Convergence and Overlap Experiments. To determine the effect of using augmented or overlapped knot span regions (δ) as the number of subdomains \mathcal{N} are increased, we use a fully symmetric double-sinc function as shown below in Equation (3.2) 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 $|\delta| = 3$).

$$378 \quad (3.2) \quad F(x) = \text{sinc}(x+1) + \text{sinc}(x-1), \forall x \in \Omega = [-4, 4]$$

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

3.2. Multi-dimensional Problem Cases. In this section, we present some results from applying parallel MFA for multidimensional problem cases.

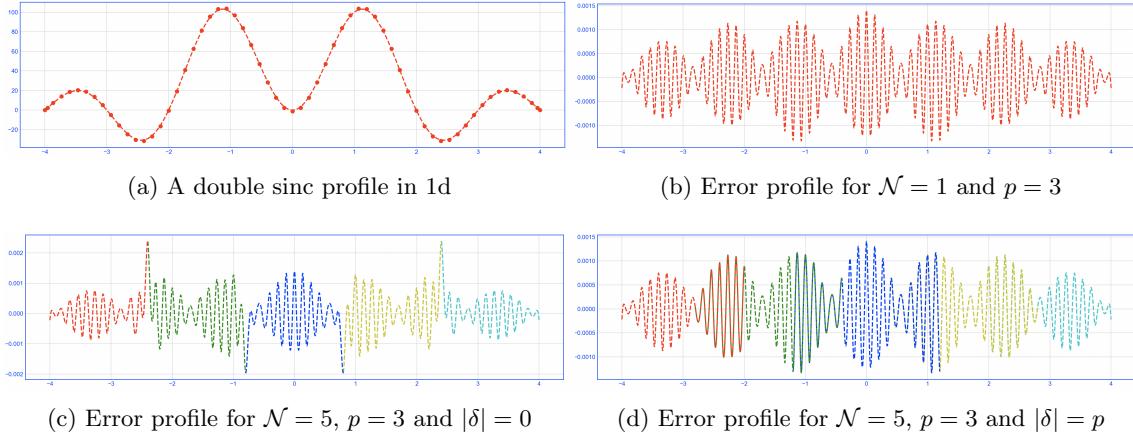


Fig. 5: Demonstration of error convergence, and effect of the overlapping spans to minimize numerical artifacts

390 **3.2.1. 2D Problem.** In order to verify the reconstruction of real world data, we choose a
 391 3D Large-Eddy Simulation code (Nek5000 [15]) applied to a double-pipe flow problem **Tom, need**
 392 **more details about dataset**. We take a 2D slice (with $|\Omega| = 200 \times 200$) along the midplane in axial
 393 direction and use the dataset for our first test study here. The reference solution and the converged,
 394 reconstructed solution with $\mathcal{N} = 5 \times 5 = 25$ subdomains with $p = 6$ and $|\delta| = 2p$ is shown in Fig. (6).
 395 The error norms are well converged with 20 floating control points per subdomain, yielding a net
 396 compression of 4x in representing the data with continuous derivatives everywhere in the domain
 397 Ω .

398 **3.2.2. 3D Problem.** Next, we present S3D, a turbulent combustion dataset generated from
 399 the fuel jet combustion simulation in the presence of an external cross-flow [7]. The 3D domain has
 400 the span $|\Omega| = 704 \times 540 \times 550$, with the raw data containing the components of the vector field. We
 401 choose to use the magnitude of this velocity field in our reconstruction study shown below in Fig. (7).
 402 While uniform refinement in knot spans does yield sufficient error reductions in most subdomains,
 403 utilizing adaptive error resolution with knot insertions and removals for MFA as previously used
 404 here on a single subdomain case [28] for the S3D problem can provide better reconstructions in
 405 addition to the iterative scheme introduced here. This extension will be pursued in the future.
 406 However, the experiments demonstrate that the MFA computations produce reconstructed data
 407 and numerical errors that are consistent and convergent for arbitrary \mathcal{N}, p and values of $|\delta|$.

408 **3.2.3. Error Convergence.** We utilize synthetic datasets shown in Equation (3.5) to perform
 409 error convergence studies and to determine areas of maximal error that iteratively are resolved
 410 between neighboring subdomains.

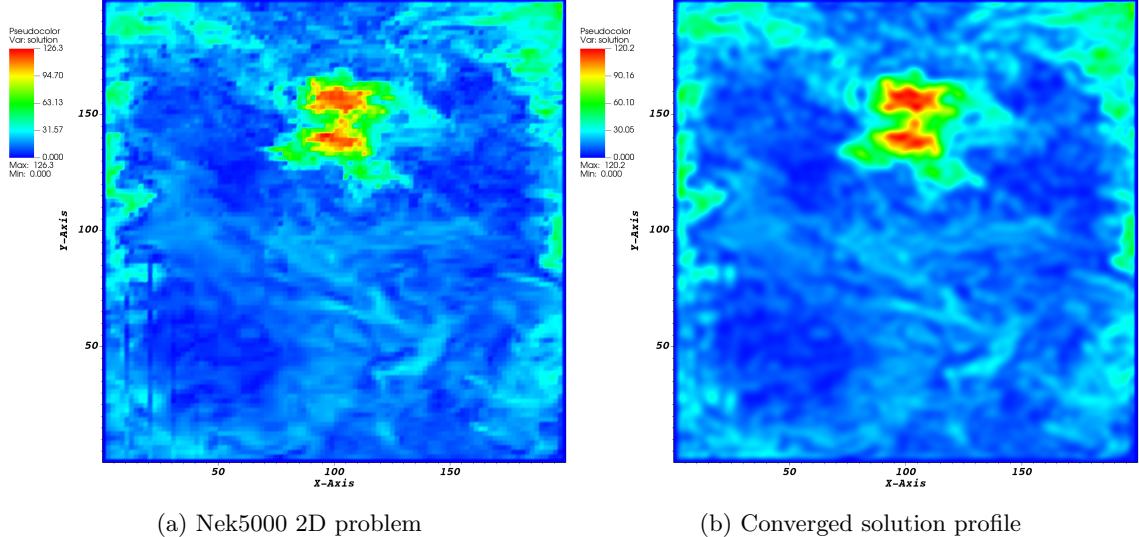


Fig. 6: 2D slice of the Nek5000 3D dataset (200×200): reference profile and B-spline MFA with $p = 6$, $\mathcal{N} = 5 \times 5$ and $|\delta| = p$

$$\begin{aligned}
 411 \quad (3.3) \quad F(x, y) &= \text{sinc}(\sqrt{x^2 + y^2}) + \text{sinc}(2(x - 2)^2 + 2(y + 2)^2), \\
 412 \quad (3.4) \quad F(x, y, z) &= \text{sinc}(\sqrt{x^2 + y^2 + z^2}) + \text{sinc}(2(x - 2)^2 + (y + 2)^2 + (z - 2)^2), \\
 413 \quad (3.5) \quad \forall \Omega &= (x, y, z) \in [-4, 4]^d.
 \end{aligned}$$

414 We plot the change in error \vec{E} between subsequent iterations of the RAS scheme for both the
 415 2D and 3D problem cases, with $|\delta| = 0$ in Fig. (8). This clearly demonstrates that the interface
 416 values between two neighboring subdomains are resolved first, and DoFs that are shared by multiple
 417 subdomains are resolved next. In all cases, the iterations converge in 2 steps, independent of \mathcal{N} or
 418 δ .

419 **3.2.4. Parallel Scalability.** To demonstrate the parallel performance of the implemented
 420 RAS iterative scheme for MFA computation with continuity preservation, we employed both closed-
 421 form synthetic and real simulation datasets in 2D and 3D. In the following sections, we present both
 422 strong scaling and weak scaling studies performed on the Theta Cray XC40 supercomputer operated
 423 by the Argonne Leadership Computing Facility (ALCF), which provides 4,392 KNL compute nodes
 424 with 64 compute cores and 192 GB DDR4 RAM per node. The interconnect is based on the Aries
 425 Dragonfly high speed network.

426 **Strong Scaling.** We consider both 2D and 3D problems to demonstrate the scaling behavior
 427 of the presented parallel MFA computational algorithm in Algorithm (2.1). One key consideration
 428 that drove selection of the subdomain sizes, and the floating knot span descriptions, is motivated
 429 by the metric to recover the original error profile from a single subdomain case. Verification studies

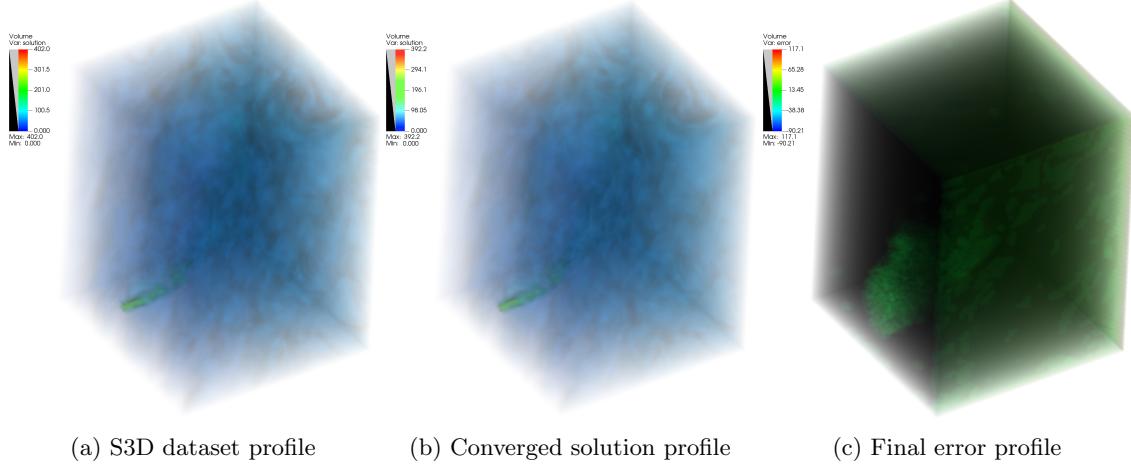


Fig. 7: Volume rendered S3D dataset: reference profile, converged MFA decoded profile and the corresponding reconstruction error with $\mathcal{N} = 8 \times 8 \times 8 = 512$, $p = 3$ and $|\delta| = 2p = 6$

430 were performed during this strong scaling test to ensure that the local subdomain errors computed
 431 on a single task, and on different process counts remain the same at convergence. This verification
 432 is important to reiterate the fact that the approximation error due to the constrained solves to
 433 recover higher-order continuity does not significantly affect the error metrics for the MFA as \mathcal{N}
 434 increases. For this reason, we used synthetic datasets generated with closed form equations for 2D
 435 and 3D studies as shown in Equation (3.5).

436 The strong scaling tests were performed on 1 to 16384 tasks in 2D, increasing by a factor of
 437 $2^2 = 4$ and the 3D tests were executed on 1 to 32768 tasks, increasing by a factor of $2^3 = 8$. In
 438 order to also better understand the effects of using augmented overlap regions (δ) on scalability, two
 439 cases with the choice of $|\delta| = 0$ and $|\delta| = p$ are shown in Fig. (9). The Python driver utilized DIY to
 440 handle block decompositions and rank assignments, as the total number of tasks used in the parallel
 441 run was increased. We measured the overall computational time for setting up the problem, the
 442 initial subdomain solves, and the consequent RAS iteration cycle to convergence, which includes
 443 the nearest neighbor communication at each iteration. We also show the time for decoding the
 444 MFA that is used to measure the errors in each subdomain and the overall total that includes the
 445 effort spent on each of these various tasks. This task-wise breakdown helps us clearly visualize the
 446 steps that scale linearly and the ones that do not.

447 As expected, the RAS iterative scheme shows excellent scalability for the chosen datasets, and
 448 the overall time to compute the MFA in parallel was reduced at a nearly ideal rate upto 10^4 MPI
 449 tasks as \mathcal{N} increases, while ensuring C^{p-1} continuity in the subdomain interfaces. The effect of using
 450 augmented overlaps has only a *small effect* on the overall scaling efficiency of the solution method.
 451 It is important to note that the dominant computational time is driven by the decoupled LSQ
 452 solution computation and decoding operations, which are embarrassingly parallel as the size of the
 453 subdomains (determined by number of knot spans) decrease in direct proportion to the tasks. Given
 454 that the scalability of the linear algebraic LSQ solvers [2, 12] and Sparse Matrix Vector (SpMV)

455 products used in the decode tasks are well understood, the bottlenecks potentially occur only from
 456 the nearest neighbor communication for constraint data exchanges, which remain significantly small
 457 in relative magnitude up to 10^4 tasks tested. The overall strong scaling efficiency remains above 70%
 458 for both overlapping and non-overlapping 2D problem cases. However, the added setup cost and
 459 nearest neighbor communication reduce the 3D problem efficiency for the overlapping subdomain
 460 cases to 40% at 32768 tasks from 85% in the nonoverlapping cases.

461 The parallel efficiency degradation behavior in augmented runs due to the high setup cost
 462 occurs due to the current choice of implementations to determine intervals in both the input space
 463 data (\vec{Q}) and extra knot spans (δ) that need to be exchanged with neighboring subdomains. This
 464 setup cost is purely local in the case when $|\delta| = 0$, in contrast to the communication dominated
 465 setup when $|\delta| > 0$.

466 **Weak Scaling.** Given that the performance of the overlapping and augmented MFA scheme
 467 was comparable to non-overlapping cases ($|\delta| = 0$), and since the error reduction from having extra
 468 overlaps always results in better solution recovery, we strictly focus on overlapping cases alone for
 469 the weak scaling study. Here, the overall work per subdomain is maintained constant, and the
 470 number of tasks are increased from 1 to 16384 in 2D, and from 1 to 32768 in 3D, similar to the
 471 strong scaling study. The weak scaling results maintaining an overall MFA compression rate of
 472 $2^d, \forall d \in [2, 3]$ is shown in Fig. (10).

473 The weak scaling study demonstrates that the overall performance of the RAS iterative scheme
 474 for large number of subdomains does not significantly affect the parallel efficiency, which are around
 475 40% in 2D and 24% in 3D at the fine limit tested. The subdomain solve and nearest neighbor data
 476 exchange dominate the overall time to solution. However, it is important to note that the actual
 477 runtime for the MFA computation only grows by a factor of 2, even on 16K processes or more.

478 **Performance Study on S3D Dataset.** Finally, we consider the case of the S3D combustion
 479 dataset shown in Fig. (7) and measure the strong scaling performance on upto 1024 processes.
 480 Using parallel MPI-IO implemented with DIY, and exposed through the Python interface, a strong
 481 scaling performance study was measured on this realistic dataset and shown in Fig. (11). Note that
 482 the IO cost for reading the chunk of data required on each task is included in the setup time shown
 483 in the figure.

484 The performance and error analysis indicate good speedup to reduce overall time for MFA
 485 computation, until nearest neighbor communication and data exchanges start dominating the overall
 486 workflow. These results show similar behavior to the strong scaling studies performed on synthetic
 487 datasets and provide confirmation on the feasibility of the presented approach for tackling real-world
 488 large datasets.

489 **4. Conclusions.** We have presented a scalable DD approach to tackle the issue of discontinuous
 490 MFA representations when performing the computations in parallel. The Restricted Additive
 491 Schwarz (RAS) method is a natural algorithmic fit for data analysis problems to create efficient
 492 MFA solutions in parallel. Through the use of Schwarz-based iterative schemes, combined with
 493 constrained local subdomain LSQ solvers, the two-level iterative technique has been shown to be robust
 494 in converging to the compressed functional representation of the given data, without sacrificing the
 495 approximation accuracy measured on a single subdomain of equivalent control point resolution.
 496 Combining NURBS-based adaptivity with a-posteriori error measures [28], and ensuring higher-
 497 order continuity across block boundaries, a scalable infrastructure can be developed based on the
 498 Algorithm (2.1).

499 The PyDIY based Python implementations for 1-, 2- and 3-dimensional problems have been

500 shown here to resolve complex solution profiles and gradient variations, even under decreasing
 501 subdomain sizes run on large-scale datasets. Additionally, we have demonstrated that the use of
 502 overlap layers δ can definitely improve the overall MFA accuracy and with a slightly higher one-time
 503 setup cost that gets amortized in the overall computation time. We have determined that for all
 504 the problems tested, including real datasets, $|\delta| = p$ to $|\delta| = 2p$ is optimal in terms of error recovery
 505 and computational cost even for 3D problems up to 32768 tasks.

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

512 Another natural way to ensure continuity across adaptively resolved NURBS or B-spline patches
 513 would be to use T-splines [33], which are specifically designed for merging higher-dimensional sur-
 514 faces with non-matching knot locations. The presented RASM-based solver approach can be easily
 515 extended to this adaptive case to impose constraints across subdomain patch boundaries, while
 516 local constraints within each block can be imposed with appropriate T-spline basis modifications.

517 Within this infrastructure, we can also utilize a multilevel MFA representation that hierarchi-
 518 cally refines the approximation at each level [32] by decreasing the number of subdomains used,
 519 similar to ideas in standard multilevel methods [6]. Such computations involving multilevel MFA
 520 such as the ones using hierarchical B-splines [5] can significantly reduce the computational cost
 521 of the local subdomain solvers, which dominate the total computational time. With appropriate
 522 choices of prolongation and restriction operators, the iterative scheme can be used with a multilevel
 523 subdomain solver to efficiently produce accurate and compact functional approximation of given
 524 data, especially in higher dimensions.

525

REFERENCES

- 526 [1] R. K. BEATSON, W. LIGHT, AND S. BILLINGS, *Fast solution of the radial basis function interpolation equations:*
 527 *Domain decomposition methods*, SIAM Journal on Scientific Computing, 22 (2001), pp. 1717–1740.
- 528 [2] M. BENZI, *Preconditioning techniques for large linear systems: a survey*, Journal of computational Physics,
 529 182 (2002), pp. 418–477.
- 530 [3] Å. BJÖRCK, *Numerical methods for least squares problems*, SIAM, 1996.
- 531 [4] P. E. BJØRSTAD AND O. B. WIDLUND, *To overlap or not to overlap: A note on a domain decomposition method*
 532 *for elliptic problems*, SIAM Journal on Scientific and Statistical Computing, 10 (1989), pp. 1053–1061.
- 533 [5] P. BORNEMANN AND F. CIRAK, *A subdivision-based implementation of the hierarchical b-spline finite element*
 534 *method*, Computer Methods in Applied Mechanics and Engineering, 253 (2013), pp. 584–598.
- 535 [6] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of computation, 31 (1977),
 536 pp. 333–390.
- 537 [7] J. H. CHEN, A. CHOUDHARY, B. DE SUPINSKI, M. DEVRIES, E. R. HAWKES, S. KLASKY, W.-K. LIAO, K.-L.
 538 MA, J. MELLOR-CRUMMEY, N. PODHORSZKI, ET AL., *Terascale direct numerical simulations of turbulent*
 539 *combustion using s3d*, Computational Science & Discovery, 2 (2009), p. 015001.
- 540 [8] J. A. COTTRELL, T. J. HUGHES, AND Y. BAZILEVS, *Isogeometric analysis: toward integration of CAD and*
 541 *FEA*, John Wiley & Sons, 2009.
- 542 [9] L. B. DA VEIGA, D. CHO, L. F. PAVARINO, AND S. SCACCHI, *Overlapping schwarz methods for isogeometric*
 543 *analysis*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 1394–1416.
- 544 [10] L. DALCIN, N. COLLIER, P. VIGNAL, A. CÔRTES, AND V. M. CALO, *Petiga: A framework for high-performance*
 545 *isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 308 (2016), pp. 151–181.
- 546 [11] L. D. DALCIN, R. R. PAZ, P. A. KLER, AND A. COSIMO, *Parallel Distributed Computing using Python*, Advances
 547 *in Water Resources*, 34 (2011), pp. 1124–1139.
- 548 [12] L. D. DALCIN, R. R. PAZ, P. A. KLER, AND A. COSIMO, *Parallel distributed computing using python*, Advances

- 549 in Water Resources, 34 (2011), pp. 1124–1139.
- 550 [13] C. DE BOOR AND R. DEVORE, *Approximation by smooth multivariate splines*, Transactions of the American
551 Mathematical Society, 276 (1983), pp. 775–788.
- 552 [14] L. DEDÈ AND A. QUARTERONI, *Isogeometric analysis for second order partial differential equations on surfaces*,
553 Computer Methods in Applied Mechanics and Engineering, 284 (2015), pp. 807–834.
- 554 [15] M. O. DEVILLE, P. F. FISCHER, AND E. H. MUND, *High-order methods for incompressible fluid flow*, vol. 9,
555 Cambridge university press, 2002.
- 556 [16] J. DONGARRA ET AL., *MPI: A Message-Passing Interface Standard Version 3.0*, High Performance Computing
557 Center Stuttgart (HLRS), (2013).
- 558 [17] W. DORNISCH AND S. KLINKEL, *Boundary conditions and multi-patch connections in isogeometric analysis*,
559 PAMM, 11 (2011), pp. 207–208.
- 560 [18] E. EFSTATIOU AND M. J. GANDER, *Why restricted additive schwarz converges faster than additive schwarz*,
561 BIT Numerical Mathematics, 43 (2003), pp. 945–959.
- 562 [19] I. GRINDEANU, T. PETERKA, V. S. MAHADEVAN, AND Y. S. NASHED, *Scalable, high-order continuity across block
563 boundaries of functional approximations computed in parallel*, in 2019 IEEE International Conference on
564 Cluster Computing (CLUSTER), IEEE, 2019, pp. 1–9.
- 565 [20] C. HOFER AND U. LANGER, *Fast multipatch isogeometric analysis solvers*, (2018).
- 566 [21] W. JAKOB, J. RHINELANDER, AND D. MOLDOVAN, *pybind11—Seamless Operability Between C++ 11 and Python*,
567 2017.
- 568 [22] M. KAPL, G. SANGALLI, AND T. TAKACS, *Construction of analysis-suitable g1 planar multi-patch parameteriza-
569 tions*, Computer-Aided Design, 97 (2018), pp. 41–55.
- 570 [23] J. LI AND Y. HON, *Domain decomposition for radial basis meshless methods*, Numerical Methods for Partial
571 Differential Equations: An International Journal, 20 (2004), pp. 450–462.
- 572 [24] P.-L. LIONS, *On the schwarz alternating method. i*, in First international symposium on domain decomposition
573 methods for partial differential equations, vol. 1, Paris, France, 1988, p. 42.
- 574 [25] N. MAI-DUY AND T. TRAN-CONG, *Approximation of function and its derivatives using radial basis function
575 networks*, Applied Mathematical Modelling, 27 (2003), pp. 197–220.
- 576 [26] F. MARINI, *Parallel additive schwarz preconditioning for isogeometric analysis*, (2015).
- 577 [27] D. MOROZOV AND T. PETERKA, *Block-Parallel Data Analysis with DIY2*, in Proceedings of the 2016 IEEE
578 Large Data Analysis and Visualization Symposium LDAV’16, Baltimore, MD, USA, 2016.
- 579 [28] Y. S. NASHED, T. PETERKA, V. MAHADEVAN, AND I. GRINDEANU, *Rational approximation of scientific data*,
580 in International Conference on Computational Science, Springer, 2019, pp. 18–31.
- 581 [29] K. PAUL, C. ZIMMERMANN, T. X. DUONG, AND R. A. SAUER, *Isogeometric continuity constraints for multi-
582 patch shells governed by fourth-order deformation and phase field models*, Computer Methods in Applied
583 Mechanics and Engineering, 370 (2020), p. 113219.
- 584 [30] T. PETERKA, S. YOUSSEF, I. GRINDEANU, V. S. MAHADEVAN, R. YEH, X. TRICOCHE, ET AL., *Foundations of
585 multivariate functional approximation for scientific data*, in 2018 IEEE 8th Symposium on Large Data
586 Analysis and Visualization (LDAV), 2018, pp. 61–71.
- 587 [31] L. PIEGL AND W. TILLER, *The NURBS book*, Springer Science & Business Media, 2012.
- 588 [32] D. SCHILLINGER, J. A. EVANS, A. REALI, M. A. SCOTT, AND T. J. HUGHES, *Isogeometric collocation: Cost
589 comparison with galerkin methods and extension to adaptive hierarchical nurbs discretizations*, Computer
590 Methods in Applied Mechanics and Engineering, 267 (2013), pp. 170–232.
- 591 [33] T. W. SEDERBERG, D. L. CARDON, G. T. FINNIGAN, N. S. NORTH, J. ZHENG, AND T. LYCHE, *T-spline simplifi-
592 cation and local refinement*, ACM transactions on graphics (TOG), 23 (2004), pp. 276–283.
- 593 [34] B. SMITH, P. BJORSTAD, AND W. GROPP, *Domain decomposition: parallel multilevel methods for elliptic partial
594 differential equations*, Cambridge university press, 2004.
- 595 [35] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive
596 schwarz preconditioning*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2402–2425.
- 597 [36] S. XU, W. JAHN, AND J.-D. MÜLLER, *Cad-based shape optimisation with cfd using a discrete adjoint*, Interna-
598 tional Journal for Numerical Methods in Fluids, 74 (2014), pp. 153–168.
- 599 [37] R. YOKOTA, L. A. BARBA, AND M. G. KNEPLEY, *Petrbf—a parallel o (n) algorithm for radial basis func-
600 tion interpolation with gaussians*, Computer Methods in Applied Mechanics and Engineering, 199 (2010),
601 pp. 1793–1804.
- 602 [38] X. ZHANG, Y. WANG, M. GUGALA, AND J.-D. MÜLLER, *Geometric continuity constraints for adjacent nurbs
603 patches in shape optimisation*, in ECCOMAS Congress, vol. 2, 2016, p. 9316.
- 604 [39] W. ZHENG, P. BO, Y. LIU, AND W. WANG, *Fast b-spline curve fitting by l-bfgs*, Computer Aided Geometric
605 Design, 29 (2012), pp. 448–462.

606 **Government License.** The submitted manuscript has been created by UChicago Argonne,
607 LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of En-
608 ergy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The U.S.
609 Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable
610 worldwide license in said article to reproduce, prepare derivative works, distribute copies to the pub-
611 lic, and perform publicly and display publicly, by or on behalf of the Government. The Department
612 of Energy will provide public access to these results of federally sponsored research in accordance
613 with the DOE Public Access Plan. <http://energy.gov/downloads/doe-public-access-plan>.

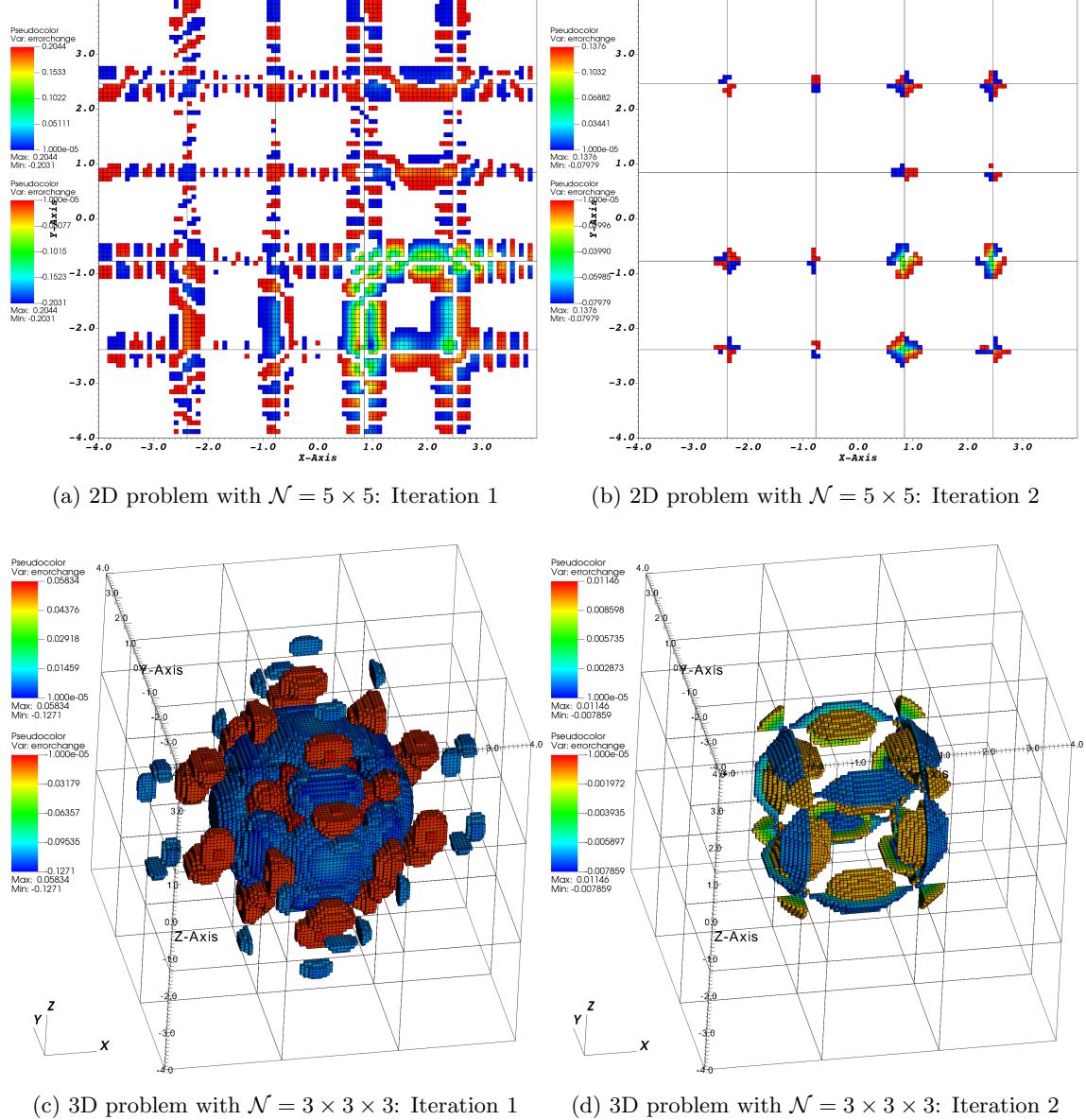
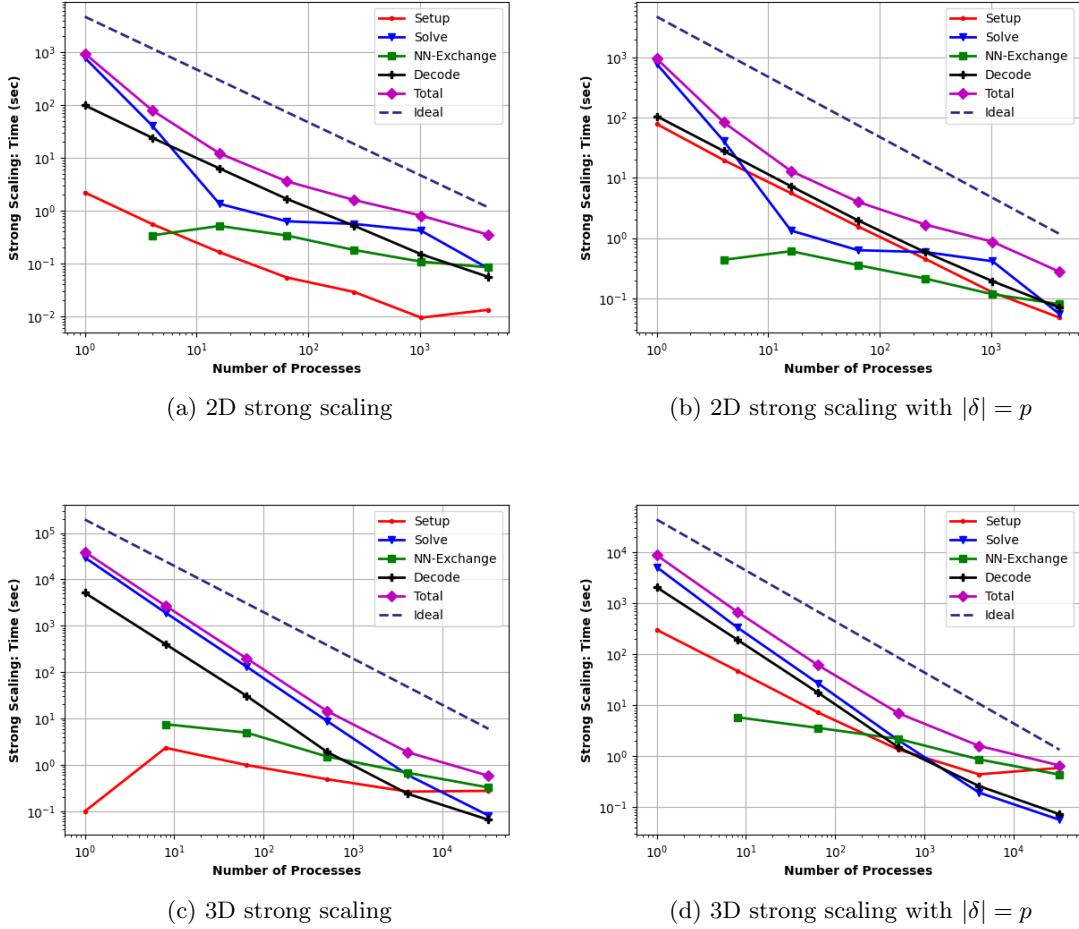
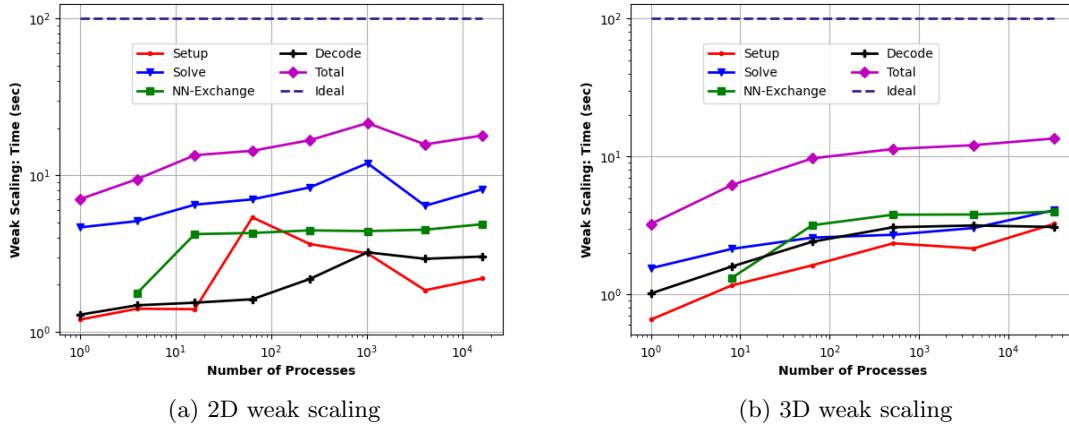
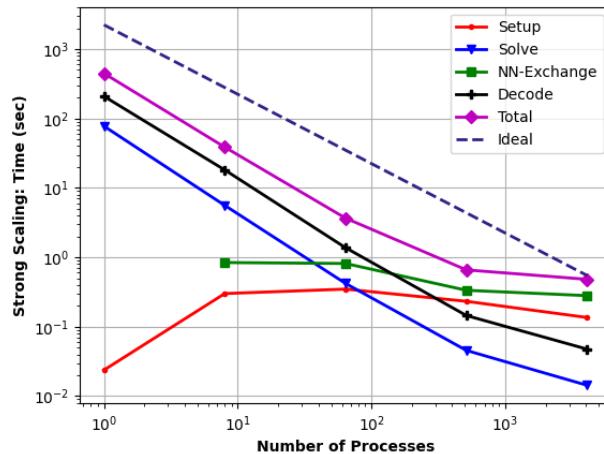


Fig. 8: Error convergence for 2D and 3D problems on the first and second iterate

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 S3D dataset with $p = 5$