

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

20 **Key words.** functional approximation, domain decomposition, scalable methods, B-spline solvers, additive
21 Schwarz

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

23 **1. Introduction.** Large-scale discrete data analysis of various scientific computational sim-
24 ulations often requires high-order continuous functional representations that have to be evaluated
25 anywhere in the domain. Such expansions described as *Multivariate Functional Approximations*
26 (MFA) [11, 33] in arbitrary dimensions allow the original discrete data to be compressed, and
27 expressed in a compact form, in addition to supporting higher-order derivative queries (without
28 further approximations such as finite differences) for complex data analysis tasks. MFA utilizes
29 approximations of the raw discrete data using a hypervolume of piecewise continuous functions.
30 One particular option is to use the variations of the B-Spline or NURBS bases [32, 31] for the MFA
31 *encoding* of scientific data. The reconstructed data in MFA retains the spatiotemporal contiguity,
32 and statistical distributions, with lesser storage requirements. Due to the potentially large datasets
33 that need to be encoded into MFA, the need for computationally efficient algorithms (in both time
34 and memory) to parallelize the work is critically important. It is also essential to guarantee that
35 the solution smoothness in the reconstructed (or *decoded*) dataset is consistently preserved when
36 transitioning from a single MFA domain to multiple domains during parallelization.

37 Achieving improved performance without sacrificing discretization accuracy requires an infra-
38 structure that is consistent in the error metrics of the decoded data and an algorithm that remains
39 efficient in the limit of large number of parallel tasks. In this paper, we will utilize domain decompo-

*Submitted to editors .

Funding: This work was funded by the Early Career Research Program, Department of Energy, USA under contract no. DE-AC02-06CH11357.

[†]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)

sition (DD) techniques [35] with data partitioning strategies to produce scalable MFA computation algorithms tha minimizes the reconstruction error when reproducing a given dataset. In such partitioned analysis, it is imperative to ensure that the continuity of the encoded and decoded data across subdomain interfaces is maintained, and remain consistent with the degree of underlying expansion bases used in MFA [31]. 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 (DD) 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) [35, 23] have been successfully employed to efficiently compute the solution of large, discretized Partial Differential Equations (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 [36] iterative techniques for RBF [22] 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 [39] 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 B-spline bases exposes a way to fully parallelize traditional, serial MFA computations.

Combining the application of DD schemes and NURBS bases with isogeometric analysis (IGA) [6, 7] for high-fidelity modeling of nonlinear PDEs [12, 26, 8] 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 [21], which can be difficult to parallelize [19] generally. In a similar vein, using B-spline bases to compute the MFA in parallel, while maintaining higher-order continuity across subdomains has not been fully explored previously.

To overcome some of these issues with discontinuities along NURBS or B-spline patches, Zhang et al. [40] 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 [18], 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. [40] and Xu et al. [38], and introduce a two-level, DD-based, parallel iterative scheme to enforce the true degree of continuity, independent of the basis function polynomial degree p , unlike the low-order constraints used previously [40]. 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. 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 DD approach used to converge the boundary continuities across MFA subdomains. 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 are 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 [6, 8]. 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 [32] 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 B-spline 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 [11]. This property becomes especially important when performing analysis and in-situ visualization directly based on the MFA representation of underlying data [37].

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

126 optimal set of control point locations within a subdomain can be posed as a solution to a linear
 127 Least-SQuares (LSQ) system that minimizes the net error of the B-spline approximation.

$$128 \quad (2.3) \quad \arg \min_{\vec{P} \in \mathbb{R}^n} E = \left\| \vec{Q} - R\vec{P} \right\|_{L_2}, \quad R \in \mathbb{R}^{m \times n}, \vec{Q} \in \mathbb{R}^m$$

129 An appropriate LSQ solver such as the one based on Cholesky decomposition or the more
 130 efficient ℓ -BFGS scheme [41] can compute the control point solution \vec{P} that minimizes the residual
 131 error \vec{E} for the given input data \vec{Q} and MFA representation of degree p . Note that the minimization
 132 procedure can be performed independently on each subdomain without dependencies as there are no
 133 constraints explicitly specified in Equation (2.3). However, in order to recover high-order continuity
 134 across subdomain interfaces, computing unconstrained solutions is insufficient. At a minimum, the
 135 DoFs lying on the shared subdomain boundaries have to be converged to recover C^0 continuity for
 136 the decoded solution data ($R\vec{P}$).

137 More generally, the constrained minimization problem to recover continuity [32] can be formulated
 138 as

$$139 \quad (2.4) \quad R\vec{P} = \vec{Q} \quad | \quad \mathcal{C}\vec{P} = \vec{G},$$

140 where \mathcal{C} is the constraint matrix imposing continuity restrictions on the control points \vec{P} along with
 141 its derivatives, with data exchanged from neighboring domains stored in \vec{G} , around the neighbor-
 142 hood of the interface $\Omega_{i,j}$ shared by subdomains i and j . With the use of penalized constraints (\mathcal{C})
 143 and Lagrange multipliers [15, 30], the solution to the constrained LSQ problem can recover optimal
 144 control point values.

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

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

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

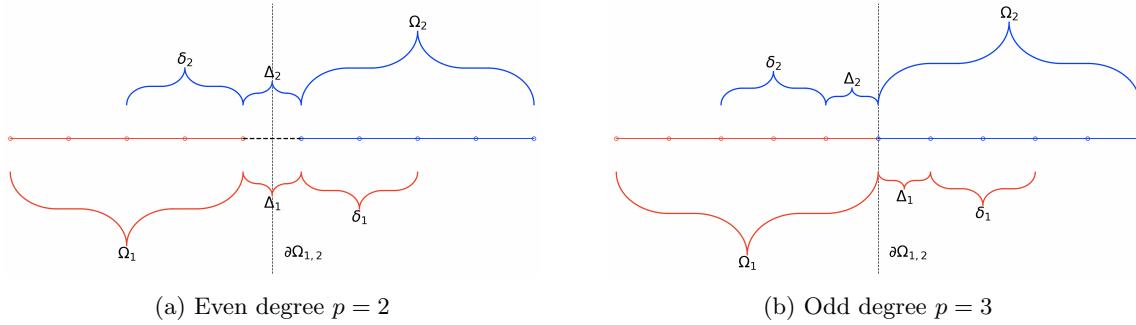


Fig. 1: Illustration: 1D parallel partitioned domain with floating (unclamped) interior knots and augmented spans ($|\delta| = 2$)

For the purpose of illustration and to explain the proposed solver methodology, let us consider a simple one dimensional domain (Ω) with two subdomains ($\mathcal{N} = 2$) as shown in Fig. (1), where Ω_1 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 [35]. 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 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 is given by $\lfloor \frac{p}{2} \rfloor$, where $\lfloor . \rfloor$ represents the floor operator. 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 knot spans that can help improve error convergence in a manner similar to overlap regions in DD methods used for PDE solvers [17, 35].

The control point DoF vector can be represented by three separate parts based on the local support of the basis expansion. The control point vector is in general given as $\vec{P} = [\vec{P}(\Omega); \vec{P}(\Delta); \vec{P}(\delta)]$. For the 1D scenario illustrated, this is shown below for $N = 2$, $p = 3$ and $|\delta| = 1$, where the operator $|\mathcal{D}|$ represents the number of knot spans in any underlying domain \mathcal{D} .

$$187 \quad (2.5) \quad \vec{P}_1(\Omega_1) \xrightarrow{\hspace{1cm}} \vec{P}_1 = \begin{bmatrix} P_1(1) \\ P_1(2) \\ \vdots \\ P_1(m) \end{bmatrix}, \quad \vec{P}_2 = \begin{bmatrix} P_2(1) \\ P_2(2) \\ \vdots \\ P_2(n) \end{bmatrix} \xrightarrow{\hspace{1cm}} \vec{P}_2(\Omega_2)$$

$\vec{P}_1(\Delta_1) \xrightarrow{\hspace{1cm}} \boxed{P_2(1)}$

$\vec{P}_1(\delta_1) \xrightarrow{\hspace{1cm}} \boxed{P_2(2)}$

$\vec{P}_2(\Delta_2) \xrightarrow{\hspace{1cm}} \boxed{P_1(m)}$

$\vec{P}_2(\delta_2) \xrightarrow{\hspace{1cm}} \boxed{P_1(m-1)}$

where m, n are the number of control points in Ω_1 and Ω_2 respectively. Note that higher degree

190 expansions (for e.g., $p > 3$) will require more support points in Δ from adjacent subdomains in
 191 order to decode the MFA up to $\partial\Omega_{1,2}$. This implies that $P_i(\Delta_i)$ in addition to the optional $P_i(\delta_i)$
 192 vectors directly provide a measure of the required cost of communication with adjacent subdomains.
 193 Now, the constrained minimization problem for the two subdomain case can be written as

$$194 \quad (2.6) \quad \left[\begin{array}{c|c} R_1(\Omega_1) & \lambda_{1,2}(\Delta_1 \cup \delta_1) \\ \hline \lambda_{2,1}(\Delta_2 \cup \delta_2) & R_2(\Omega_2) \end{array} \right] \left[\begin{array}{c} \vec{P}_1 \\ \vec{P}_2 \end{array} \right] = \left[\begin{array}{c} \vec{Q}_1 \\ \vec{Q}_2 \end{array} \right]$$

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

202 The coupling blocks $\lambda_{i,j}$ can be viewed as Lagrange multipliers that explicitly couple the control
 203 point DoFs across a subdomain interface ($\vec{P}_1 \cap \vec{P}_2$) such that continuity is preserved in a weak
 204 sense [32]. Using appropriate Schur complements to eliminate the coupled DoF contributions in
 205 each subdomain, with $\lambda_{i,j}$ evaluated at *lagged* iterates of adjacent subdomains, the set of coupled
 206 constrained equations in Equation (2.6) can be completely decoupled for each subdomain. This
 207 modified system resembles a block-Jacobi operator of the global system. The scheme illustrated in
 208 this section follows ideas similar to the Jacobi-Schwarz method [17] and the overlapping, restricted-
 209 Additive-Schwarz (RAS) scheme [36].

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 [36, 17], 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 **Augmenting Knot Spans with Overlap.** One of the key metrics of interest is that the
 219 parallel solver infrastructure does not amplify any approximation errors unresolved by the tensor
 220 product B-spline mesh. Since the local decoupled subdomain solution is encoded accurately to
 221 satisfy Equation (2.3) in each individual subdomain without any data communication (i.e., em-
 222 barassingly 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 [23, 17] have been proven
 229 to be more stable, efficient and scalable compared to nonoverlapping variants [4, 36]. 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. The set of $\vec{P}(\delta)$ are explicitly used only to impose constraints that contribute to the reconstruction of datasets, and hence play a role in the approximation error of MFA. 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 $\vec{P}_i(\delta_i), \forall i \in [0, 3]$.

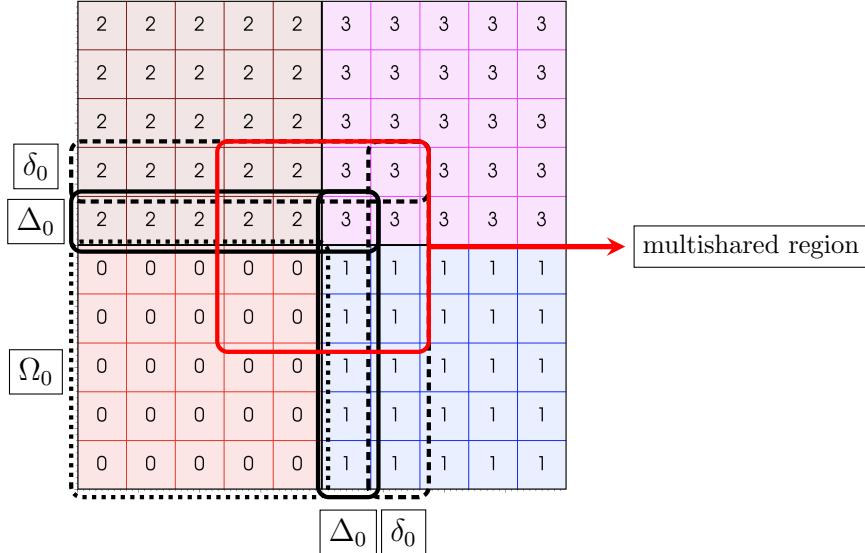


Fig. 2: 2D subdomains with $N = 4$, $p = 3$ and the augmented overlap $|\delta| = 1$ showing local subdomains Ω_i , mandatory overlap for floating knots Δ_i , optional overlap regions δ_i , where $i \in [0, N - 1]$, and multishared DoF regions that couple multiple subdomains (marked in red).

We also introduce a definition for compression ratio (η), which gives the ratio of the total input points in the dataset ($\dim \vec{Q}$) to the total control points ($\dim \vec{P}$) used in the MFA B-spline representation. As $\eta \rightarrow 1$, one can achieve better error residuals in comparison to the reference data for a given degree p , while $\eta \gg 1$ produces smooth approximations with larger error profiles. When we analyze the results in this manuscript, it should be remembered that the reconstruction error is always inversely proportional to η .

Next, the parallel MFA computation workflow that will be used with domain-decomposed subdomain partitions is presented in detail.

255 **2.3. Solver Workflow.** Computing the functional approximation to large-scale datasets re-
 256 quires efficient solvers at two levels: firstly, the local decoupled subdomain solver for Equation (2.3),
 257 and next, the constrained minimization problem in Equation (2.4). Hence, the global problem re-
 258 duces to a series of local minimization problems in each subdomain.

$$259 \quad (2.7) \quad \begin{aligned} & \arg \min_{\vec{P} \in \mathbb{R}^n} \left\| \vec{Q}_\ell - R_\ell \vec{P}_\ell \right\|_{L_2}, \quad R_\ell \in \mathbb{R}^{m \times n}, \vec{Q}_\ell \in \mathbb{R}^m, \vec{P}_\ell \in \Omega_i \\ & \text{subject to } \sum_{\partial\Omega_{i,j}} \left[\mathcal{F}_{ij}(\vec{P}_i(\Delta_i), \vec{P}_j(\Omega_j)) + \mathcal{F}_{ij}(\vec{P}_i(\delta_i), \vec{P}_j(\Omega_j)) \right]^2 = 0, \quad \forall i, j \in [1, \dots, \mathcal{N}] \end{aligned}$$

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

262 **2.3.1. Subdomain Solvers.** For the linear LSQ solvers that can be used to compute local
 263 subdomain control point solution \vec{P} , there are a variety of choices available. Direct methods like
 264 Singular Value decomposition or Cholesky decomposition operating on the normal equations [3] can
 265 compute optimal values. Alternatively, the iterative LSQ solvers such as orthogonal decomposition
 266 methods based on QR and QZ factorizations are more stable, especially when the normal form of
 267 the operator, $R^T R$, is ill-conditioned.

268 **2.3.2. Restricted Additive-Schwarz Solvers.** The outer RAS iterations work together
 269 with nearest neighbor communication procedures to exchange shared DoF data between adjacent
 270 subdomains. This is an important step to ensure that \vec{P} data computed through the LSQ procedure
 271 are consistent and high-order continuous across subdomain boundaries. The final minimized control
 272 point solution is achieved when the interface solutions match on all $\partial\Omega_{i,j} \in \Omega$ rendering zero jump
 273 residuals (\mathcal{F}_{ij}) on Δ and δ shared domains in Equation (2.7).

274 It is also important to note that unlike the blending approaches that can be directly applied
 275 on decoded data [18], the numerical error with the constrained iterative scheme is not bounded by
 276 the original partitioned, unconstrained least-squares solution; i.e., imposing subdomain boundary
 277 constraints can create artificial numerical peaks (non-monotonic) in reconstructed data as we con-
 278 verge towards continuity recovery. A solution to address this issue is to increase the amount of
 279 overlap range to ensure uniform convergence to the true single-subdomain solution error, even as
 280 the number of subdomains (\mathcal{N}) increases.

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

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

- 289 1. Singly-shared (\mathcal{SS}) DoFs that are shared between two adjacent, neighboring subdomains;
 290 e.g., direct interface data belonging to $(\Delta_i \cup \delta_i) \cap \Omega_j$ in Fig. (2), $\forall i \in [1, \mathcal{N}]$ and $j \in [1, \hat{\mathcal{N}}]$,
 291 where $\hat{\mathcal{N}}$ represents the set of nearest neighbors sharing an interface $\Omega_{i,j}$
- 292 2. Multi-shared (\mathcal{MS}) DoFs that are shared by multiple (more than 2) neighboring sub-
 293 domains; 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
 294 $\mathcal{S}_i = \Omega_i \cup \Delta_i \cup \delta_i$.

Given both these specific DoF groups, the overlapping RAS scheme applied to MFA computation *always converges in 2 outer iterations* for problems in 2D and 3D, and a single iteration in 1D (due to lack of \mathcal{MS} DoFs). This result is demonstrated in Section (3). Note that achieving full convergence with high-order continuity in a single iteration is possible by combining constraints for both \mathcal{SS} and \mathcal{MS} DoFs. However, due to complexities in the implementation of constraint matching in Equation (2.7) for this algorithmic optimization, it has not been pursued here.

It is necessary to mention that we use a uniform weighting procedure to converge the shared DoFs between different subdomains, where the weights for each shared DoF is assigned as $w_i = \frac{1}{n_s}$, with n_s being the number of subdomains containing DoF i within its domain \mathcal{S} . This weighing procedure can be trivially replaced by Shephard's functions, especially in the context of adaptive discretizations with variable knot displacements.

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

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,
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)),
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)).

At convergence, the interface data at $\partial\Omega_{1,2}$ will satisfy the higher-order continuity prescriptions specified by the user, thereby guaranteeing full regularity of C^{p-1} . The illustration in Fig. (1), and the methodology description in this section can be generalized and extended to arbitrary dimensions in the tensor-product setting (with the parametric domain represented by a d -dimensional hypercube) as shown in Fig. (2) for 2D. Using knot insertion and removal strategies based on deCasteljau subdivision procedures [32], individual subdomains can also be adapted to resolve fast varying solutions and to reduce decoded error to be within user-specified tolerances [29]. While adaptivity has not been fully explored in the current work, enabling variable resolutions in different dimensions is a natural extension of the work that will still preserve high-order continuity. The implementation of the presented approach with domain decomposition strategies, combined with overlapping RAS scheme yields a scalable scheme that will be demonstrated to be suitable for tackling large-scale data analysis problems.

2.4. Implementation. The DD techniques presented here for MFA computation are primarily implemented in Python, with main dependencies on SciPy for B-spline bases evaluations and linear algebra routines. Additionally, the drivers utilize Python bindings (PyDIY) for the DIY [28] C++ library. DIY is a programming model and runtime for block-parallel analytics on distributed-memory machines, built on MPI-3 [14]. Rather than programming for process parallelism directly in MPI, the programming model in DIY is based on block parallelism. In DIY, data are decomposed into subdomains called blocks. One or more of these 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. PyDIY utilizes PyBind11 [20] and MPI4Py [9] 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

Algorithm 2.1 Domain Decomposed MFA Solver

Input: Dataset and coordinates
Parameters: \mathcal{N}, p, m
Setup: Decompose domain into blocks with DIY, compute R
while $\vec{P}(\Omega)$ not converged **do**
 for $i \leftarrow 1$ to \mathcal{N} **do**
 if $\vec{P}_i = 0$ **then**
 Solve: Unconstrained local LSQ problem
 for $j \leftarrow 1$ to $\hat{\mathcal{N}}$ **do** ▷ where $\hat{\mathcal{N}} = \text{nearest neighbors}$
 – $\vec{P}_i(\Omega_i \cap (\Delta_j \cup \delta_j)) \rightarrow \text{enqueue outgoing constraints}$
 – Exchange shared DoFs with all nearest neighbor blocks
 – $\vec{P}_i(\Delta_i \cup \delta_i) \leftarrow \text{dequeue incoming constraints}$
 – Enforce constraints for $\vec{P}_i(\Delta_i \cup \delta_i) \cap \vec{P}_j, \forall j \in [1, \mathcal{N}]$: Drive $\mathcal{F}_{ij} \rightarrow 0$
 – Update local error $E_i := \|\vec{Q}_i - R\vec{P}_i\|_{L_2}$
 Check convergence metrics
 Write MFA to disk for analysis and visualization

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

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

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

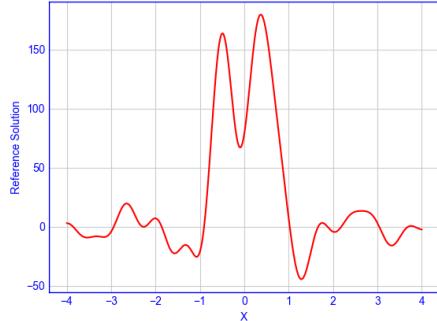
355 **3. Results.** To demonstrate the effectiveness of the iterative algorithm for MFA computation,
356 we devised a series of analytical closed form functionals and utilized real-world scientific datasets
357 in both 2- and 3-dimensions obtained from high-fidelity simulations. All runs shown in this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

360 **3.1. 1D Results.** In this section, detailed analysis on the convergence and accuracy of various
361 MFA continuity recovery approaches are presented.

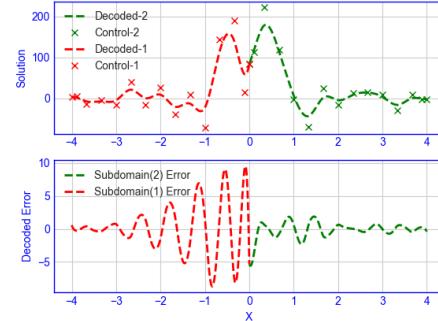
362 **3.1.1. Comparison of Clamped vs Floating Boundary Knots.** To demonstrate the
363 choice of using floating knots vs the low-order (C^0) continuous clamped knots at subdomain bound-
364aries, we choose an analytical closed form reference solution of the form:

365 (3.1) $F(x) = \text{sinc}(x) + \text{sinc}(2x - 1) + \text{sinc}(3x + 1.5), \forall x \in \Omega = [-4, 4]$

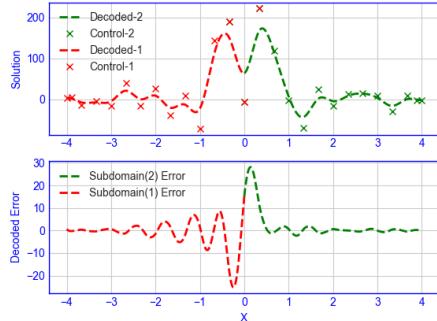
366 The reference solution $F(x)$, the results from the clamped knots, and floating knots with and
 367 without augmented overlap regions $\delta = p$ are shown in Fig. (3). The figures show the recovered
 368 solutions and the corresponding decoded error from MFA evaluation for a $\mathcal{N} = 2$ and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



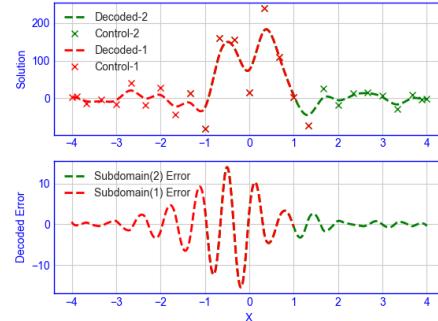
(a) Input analytical 1D solution profile



(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 $p = 3$

375 To further demonstrate the continuity recovery behavior, we plot the error profile \vec{E} for these

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

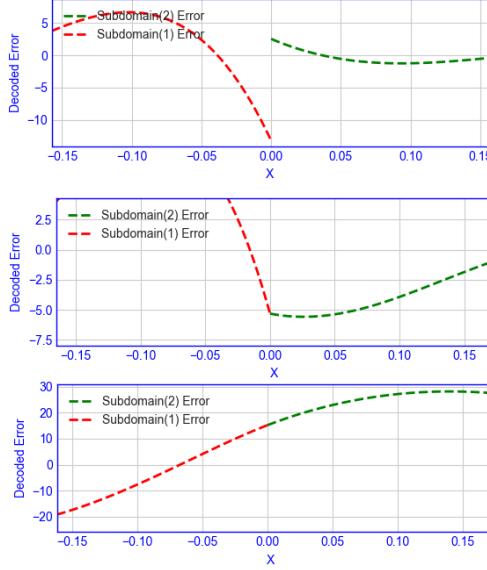


Fig. 4: Zoomed error plots at interface $\Omega_{1,2}$ for 1D analytical dataset with 10^4 input points, $\mathcal{N} = 2$ and $p = 3$. Top: unconstrained and decoupled subdomain profile (discontinuous). Middle: clamped interface knots. Bottom: floating interface knots.

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 $|\delta| = 3$).

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

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

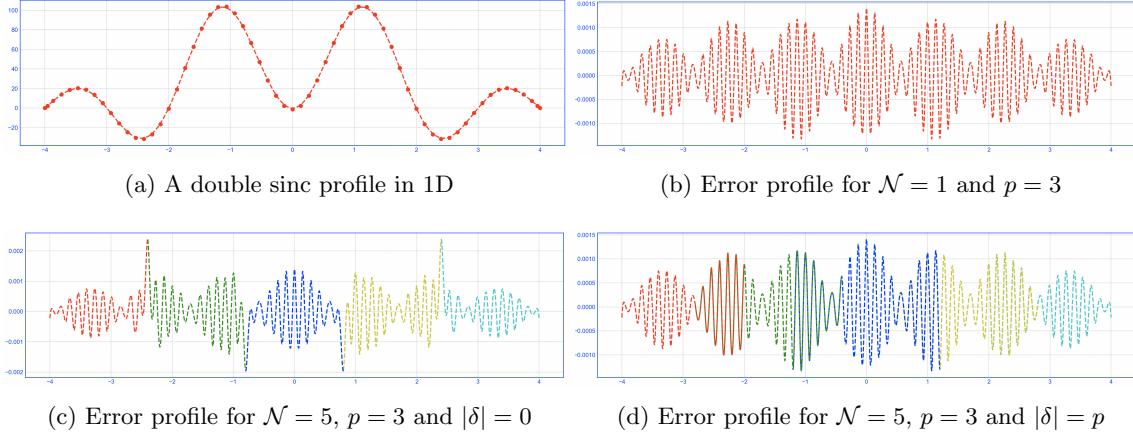


Fig. 5: Demonstration of error convergence, and effect of the overlapping spans to minimize numerical artifacts

394 the underlying data and the local bases support spans, the error profiles as shown in Fig. (5d)
 395 approaches the reference profile (with $\mathcal{N} = 1$) shown in Fig. (5b). Heuristically, for many of the
 396 problems tested, using $|\delta| = p$ provides optimal convergence as number of subdomains increase,
 397 even though increasing this parameter to $|\delta| = 2p$ or higher will in general always improve the
 398 numerical accuracy at the cost of higher communication costs between neighboring subdomains.

399 **3.2. Multi-dimensional Problem Cases.** In this section, we present some results from
 400 applying parallel MFA for multidimensional problem cases.

401 **3.2.1. 2D Problem.** A 3D vector field representing the numerical results of a large-eddy
 402 simulation of Navier-Stokes equations for the MAX experiment [27] is representative of turbulent
 403 mixing and thermal striping that occur in the upper plenum of liquid sodium fast reactors. The
 404 data, generated by the Nek5000 solver [13], have been resampled from their original topology onto
 405 a $200 \times 200 \times 200$ regular grid, and the magnitude of the velocity vector is associated with each 3D
 406 domain point [31]. Out of this dataset, a 2D slice (with $|\Omega| = 200 \times 200$) along the midplane in axial
 407 direction is used for our first study here. The reference solution and the converged, reconstructed
 408 solution with $\mathcal{N} = 5 \times 5 = 25$ subdomains with $p = 6$ and $|\delta| = 2p$ is shown in Fig. (6) for different
 409 compression ratios. Depending on the usecase for MFA reconstruction, the converged error norms
 410 with 20 floating control points per subdomain yielding a net compression of 4X is sufficient to
 411 evaluate continuous derivatives everywhere in the domain Ω . A full MFA representation (with
 412 compression ratio=1) is also shown in Fig. (6c), which can fully reconstruct the original features in
 413 the input dataset in contrast to Fig. (6b) that shows a lossy smoothing of the sharp features in the
 414 original data.

415 **3.2.2. 3D Problem.** Next, we present S3D, a turbulent fuel jet combustion dataset generated
 416 from a simulation in the presence of an external cross-flow [5]. The 3D domain has the span
 417 $|\Omega| = 704 \times 540 \times 550$, with the raw data containing three components of the vector field. We choose
 418 to use the magnitude of this velocity field in our reconstruction study shown below in Fig. (7) with

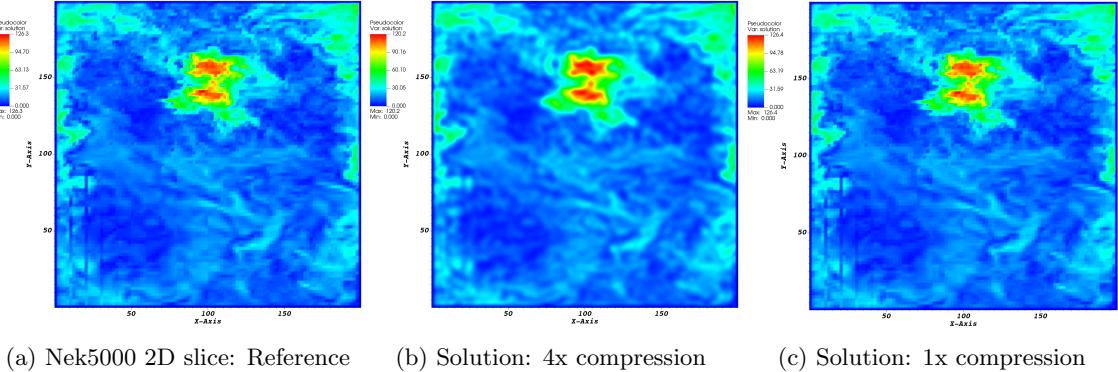


Fig. 6: 2D slice of the Nek5000 3D dataset (200×200): reference profile and B-spline MFA with $p = 6$, $\mathcal{N} = 5 \times 5$, $|\delta| = p$ for 4X compression (middle), and 1X compression (right).

419 209M points. The converged MFA reconstruction shown in the figure with $8^3 = 512$ subdomains
 420 and $n = 35$ per direction in each subdomain yields a net compression ratio of $\frac{(704 \times 540 \times 550)}{(8 \times 35)^3} \approx 9.5$.
 421 While uniform refinement in knot spans does yield sufficient error reductions in most subdomains,
 422 utilizing adaptive error resolution with knot insertions and removals for MFA as previously used
 423 here for single subdomains [29] for the S3D problem can provide better reconstructions in addition
 424 to the iterative scheme introduced here. Such extensions will be pursued in the future. However, the
 425 experiments demonstrate that the MFA computations produce reconstructed data and numerical
 426 errors that are consistent and convergent for arbitrary \mathcal{N}, n, p and values of $|\delta|$, proving the feasibility
 427 of the algorithm.

428 **3.2.3. Error Convergence.** We utilize synthetic datasets shown in Equation (3.3) and Equa-
 429 tion (3.4) to perform error convergence studies and to determine areas of maximal error that iter-
 430 atively are resolved between neighboring subdomains.

431 (3.3) $F(x, y) = \text{sinc}(\sqrt{x^2 + y^2}) + \text{sinc}(2(x - 2)^2 + 2(y + 2)^2), \forall (x, y) \in [-4, 4]^2,$

432 (3.4) $F(x, y, z) = \text{sinc}(\sqrt{x^2 + y^2 + z^2}) + \text{sinc}(2(x - 2)^2 + (y + 2)^2 + (z - 2)^2), \forall (x, y, z) \in [-4, 4]^3.$

433 We plot the change in error \vec{E} between subsequent iterations of the RAS scheme for both 2D and
 434 3D problem cases with $|\delta| = 0$ in Fig. (8). This clearly demonstrates that the \mathcal{SS} interface values
 435 are resolved first, and then \mathcal{MS} DoFs are resolved further. In all cases, the iterations converge in
 436 2 steps, independent of \mathcal{N} or δ .

437 **3.2.4. Parallel Scalability.** To demonstrate the parallel performance of the implemented
 438 RAS iterative scheme for MFA computation with continuity preservation, we employed both closed-
 439 form synthetic and real simulation datasets in 2D and 3D. In the following sections, we present both
 440 strong scaling and weak scaling studies performed on the Theta Cray XC40 supercomputer operated
 441 by the Argonne Leadership Computing Facility (ALCF), which provides 4,392 KNL compute nodes
 442 with 64 compute cores and 192 GB DDR4 RAM per node. The interconnect is based on the Aries
 443 Dragonfly high speed network.

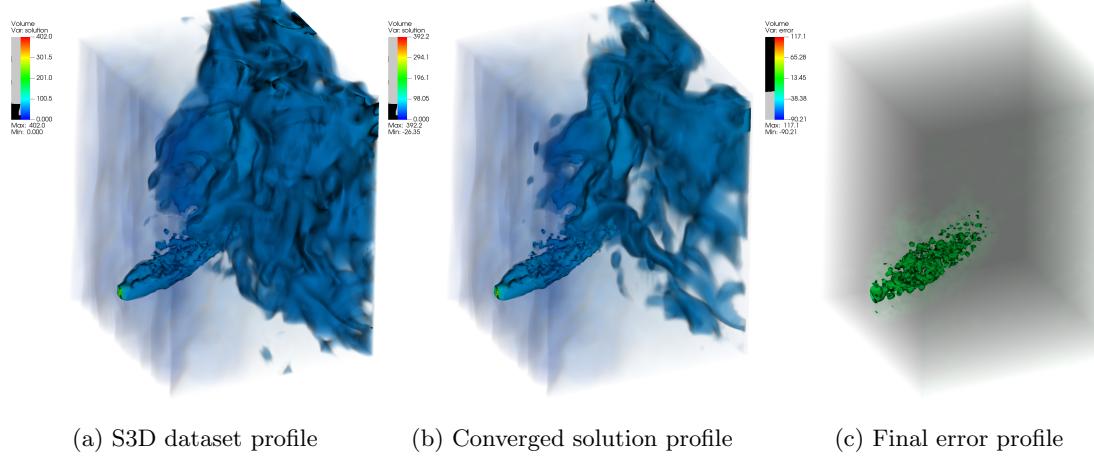


Fig. 7: Volume rendered S3D dataset with compression rate $\eta = 9.5$: reference profile, converged MFA decoded profile and the corresponding reconstruction error with $\mathcal{N} = 8 \times 8 \times 8 = 512$, $p = 3$ and $|\delta| = 2p = 6$

444 **Strong Scaling.** We consider both 2D and 3D problems to demonstrate the scaling behavior
 445 of the presented parallel MFA computational algorithm in Algorithm (2.1). One key consideration
 446 that drove selection of the subdomain sizes, and the floating knot span descriptions, is motivated
 447 by the metric to recover the original error profile from a single subdomain case. Verification studies
 448 were performed during this strong scaling test to ensure that the local subdomain errors computed
 449 on a single task, and on different process counts remain the same at convergence. This verification
 450 is important to reiterate the fact that the approximation error due to the constrained solves to
 451 recover higher-order continuity does not significantly affect the error metrics for the MFA as \mathcal{N}
 452 increases. For this reason, we used synthetic datasets generated with closed form equations for 2D
 453 and 3D studies as shown in Equation (3.3) and Equation (3.4).

454 The strong scaling tests were performed on 1 to 16,384 parallel tasks in 2D, increasing by a
 455 factor of $2^2 = 4$, and the 3D tests were executed on 1 to 32,768 tasks, increasing by a factor of
 456 $2^3 = 8$. We also note that the 2D and 3D case for these studies used 400M and 1.331B input
 457 points respectively, with a corresponding $\eta = 4$ and $\eta = 1.25$ that was maintained constant for
 458 all runs. In order to also better understand the effects of using augmented overlap regions (δ) on
 459 scalability, two cases with the choice of $|\delta| = 0$ and $|\delta| = p$ are shown in Fig. (9). The Python driver
 460 utilized DIY to handle block decompositions and rank assignments, as the total number of tasks
 461 used in the parallel run was increased. We measured the overall computational time for setting up
 462 the problem, the initial subdomain solves, and the consequent RAS iteration cycle to convergence,
 463 which includes the nearest neighbor communication at each iteration. We also show the time for
 464 decoding the MFA that is used to measure the errors in each subdomain, and the overall total that
 465 includes the effort spent on each of these various tasks. This task-wise breakdown helps us clearly
 466 visualize the steps that scale linearly and the ones that do not.

467 As expected, the RAS iterative scheme shows excellent scalability for the chosen datasets, and

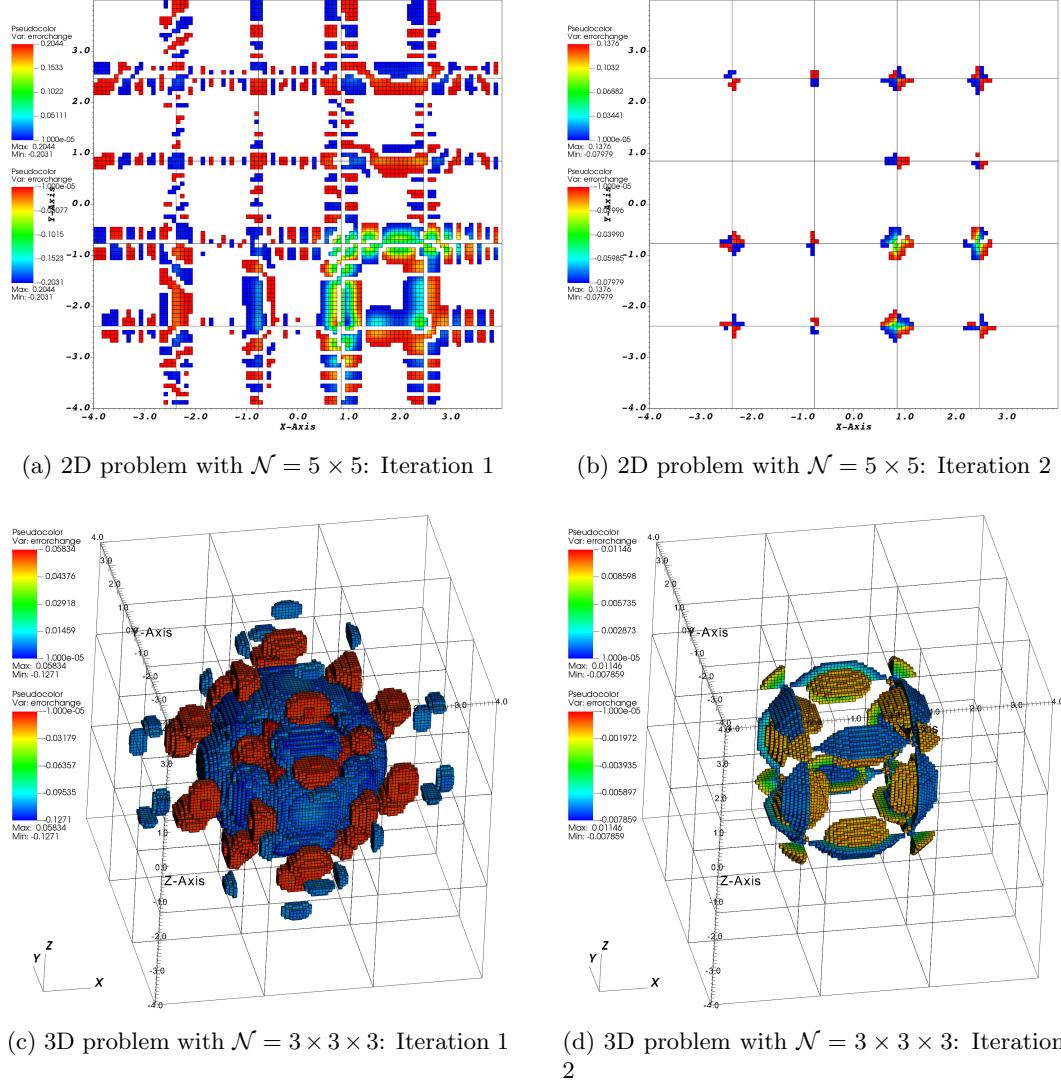
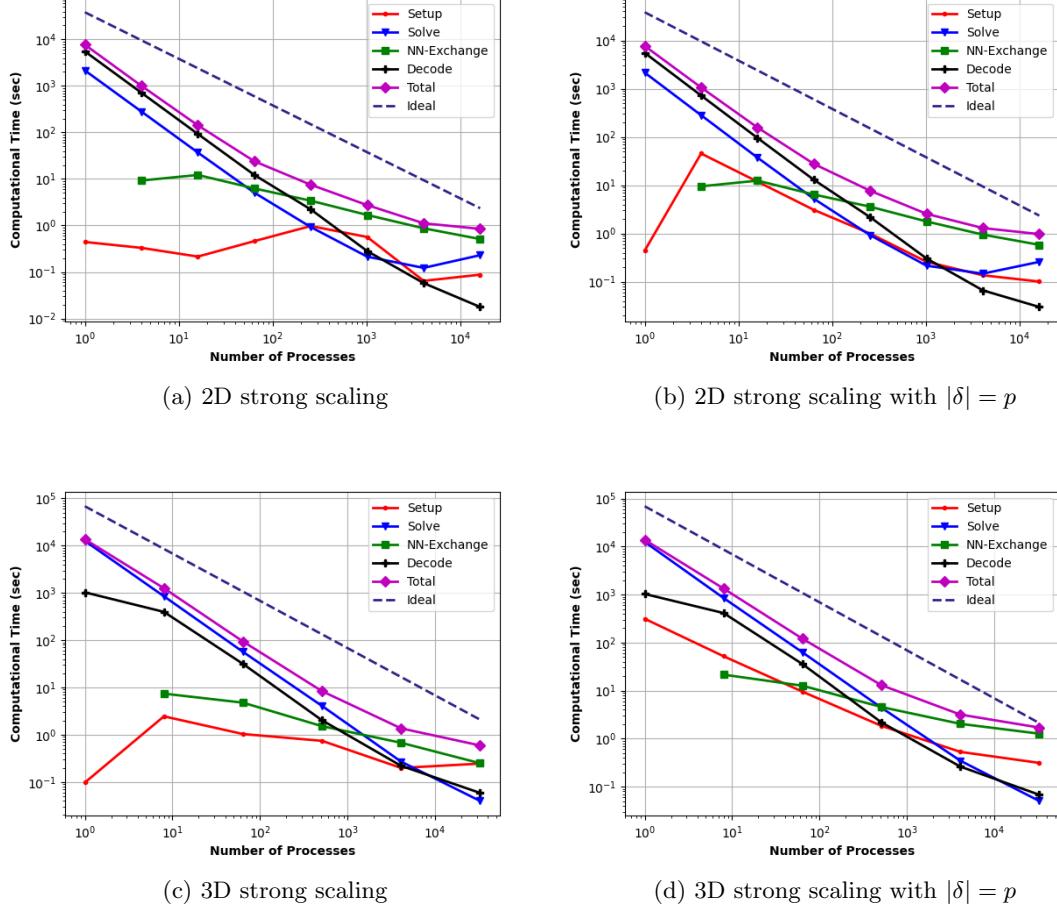


Fig. 8: Error convergence for 2D and 3D problems on the first and second iterate

468 the overall time to compute the MFA in parallel was reduced at a nearly ideal rate up to 10^3 MPI
 469 tasks as \mathcal{N} increases, while ensuring C^{p-1} continuity in the subdomain interfaces. It is important to
 470 note that the dominant computational time is driven by the decoupled LSQ solution computation
 471 and decoding operations, which are embarrassingly parallel as the size of the subdomains (determined
 472 by number of knot spans) decreases in direct proportion to the tasks. Given that the scalability
 473 of the linear algebraic LSQ solvers [2, 10] and Sparse Matrix Vector (SpMV) products used in the
 474 decode tasks are well understood, the bottlenecks potentially occur only from the nearest neighbor

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

475 communication for constraint data exchanges, which remain relatively small in magnitude up to
 476 32K processes tested in the 3D experiments. The overall strong scaling efficiency remains around
 477 50% for both overlapping and non-overlapping 2D problem cases at 16,384 tasks. However, the
 478 added setup cost and less than ideally scaling nearest neighbor communications reduce the 3D
 479 problem efficiency for the overlapping subdomain cases to 25% at 32,768 tasks from 70% in the
 480 nonoverlapping cases.

481 Additionally, the parallel efficiency degradation behavior in augmented runs due to the high
 482 setup cost occurs due to the current choice of implementations to determine intervals in both the
 483 input space data (\vec{Q}) and extra knot spans (δ) that need to be exchanged with neighboring subdo-
 484 mains. This setup cost is purely local in the case when $|\delta| = 0$, in contrast to the communication
 485 dominated setup when $|\delta| > 0$. It is evident that the cost of nearest neighbor exchanges at scale

486 is driven by the size of the messages being transferred between subdomains, which is a function of
 487 $|\delta|$. The increased amount of data shows smaller degradation in communication for 2D problems
 488 as compared to 3D, and hence the strong scaling behavior is dependent on the dimension of the
 489 problem being solved that is representative of performance in all structured grid solvers for PDEs.

490 **Weak Scaling.** Given that the performance of the overlapping and augmented MFA scheme
 491 was comparable to non-overlapping cases ($|\delta| = 0$), and since the error reduction from having extra
 492 overlaps always results in better solution recovery, we strictly focus on overlapping cases alone for
 493 the weak scaling study. Here, the overall work per subdomain is maintained constant, and the
 494 number of tasks are increased from 1 to 16,384 in 2D, and from 1 to 32,768 in 3D, similar to the
 495 strong scaling study. The weak scaling results maintaining an overall MFA compression rate of
 496 $\eta = 2^d, \forall d \in [2, 3]$ is shown in Fig. (10).

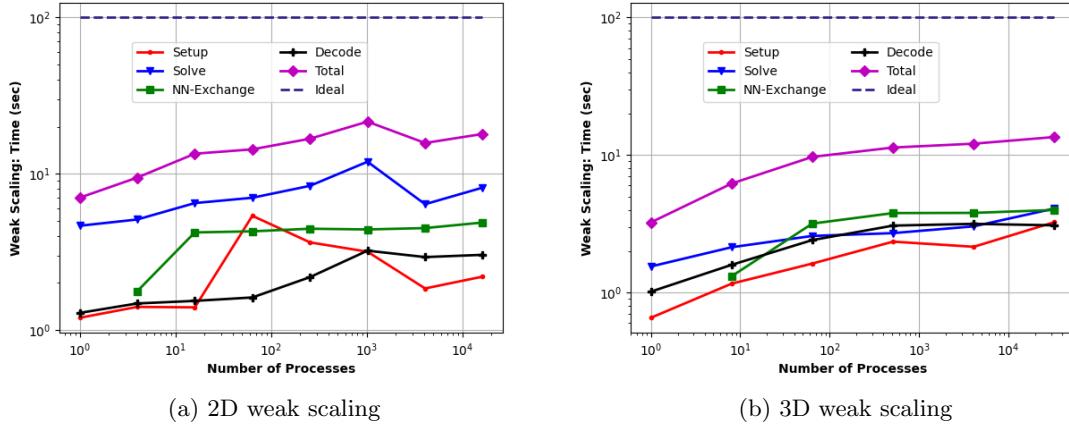


Fig. 10: Weak scaling performance of RAS solver with $p = 3$ and $|\delta| = p$

497 The weak scaling study demonstrates that the overall performance of the RAS iterative scheme
 498 for large number of subdomains does not significantly affect the parallel efficiency, which are around
 499 40% in 2D and 24% in 3D at the fine limit tested. The subdomain solve and nearest neighbor data
 500 exchange dominate the overall time to solution. However, it is important to note that the actual
 501 runtime for the MFA computation only grows by a factor of 2, even on 16K processes or more.

502 **Performance Study on S3D Dataset.** Finally, we consider the case of the S3D combustion
 503 dataset shown in Fig. (7) and measure the strong scaling performance on up to 4,096 processes.
 504 Using parallel MPI-IO implemented with DIY, and exposed through the Python interface, a strong
 505 scaling performance study was measured on this realistic dataset and shown in Fig. (11). Note that
 506 the IO cost for reading the chunk of data required on each task is included in the setup time shown
 507 in the figure.

508 The performance and error analysis indicate good speedup to reduce overall time for MFA
 509 computation, until nearest neighbor communication and data exchanges start dominating the overall
 510 workflow. These results show similar behavior to the strong scaling studies performed on synthetic

511 datasets and provide confirmation on the feasibility of the presented approach for tackling real-world
 512 large datasets.

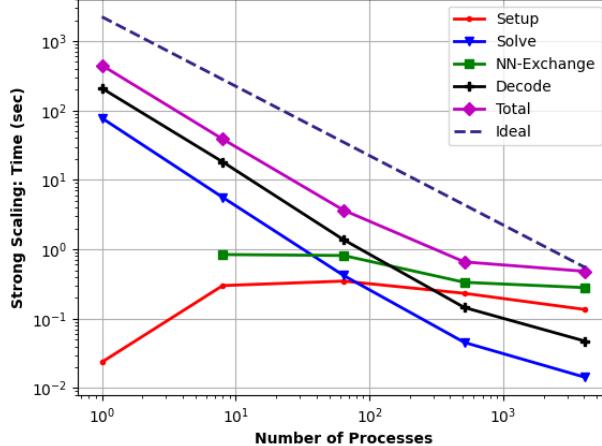


Fig. 11: Strong scaling for the S3D dataset with $p = 5$

513 **4. Summary.** We have presented a scalable DD approach to tackle the issue of discontinuous
 514 B-spline based MFA representations when performing the computations in parallel. The Restricted
 515 Additive Schwarz (RAS) method is a natural algorithmic fit for data analysis problems to create
 516 efficient MFA solutions in parallel. Through the use of Schwarz-based iterative schemes, combined
 517 with constrained local subdomain solvers, the two-level iterative technique has been shown to
 518 be robust in converging to the compressed functional representation of the given data, without
 519 sacrificing the approximation accuracy measured on a single subdomain of equivalent control point
 520 resolution. Replacing B-spline bases with NURBS bases ($W \neq 1$) only requires imposing the
 521 constraints on the $\vec{P}_i W_i$ data instead of \vec{P}_i alone, which is naturally accomplished with minor
 522 modifications in Algorithm (2.1). This can also be combined with a posteriori error measures [29]
 523 to adaptively resolve solution variations while ensuring higher-order continuity across subdomain
 524 boundaries with appropriate knot insertion, removal and communication of shared DoFs in $\Delta \cup$
 525 δ regions. Another natural way to ensure continuity across adaptively resolved NURBS or B-
 526 spline patches would be to use T-splines [34], which are specifically designed for merging higher-
 527 dimensional surfaces with non-matching knot locations. All presented ideas should extend for
 528 T-splines instead of B-splines as well with modifications to \mathcal{C} in Equation (2.4).

529 We have demonstrated that the use of overlap layers δ can certainly improve the overall MFA
 530 accuracy, with a slightly larger one-time setup cost that gets amortized in the overall computation
 531 time. We determined that for all the problems tested, including real datasets, $|\delta| = p$ to $|\delta| = 2p$ is
 532 optimal in terms of error recovery and computational cost even for 3D problems up to 32,768 tasks.
 533 The iterative scheme shows good parallel performance for both 2D and 3D problems tested, and
 534 the parallel efficiency degrades only when the cost of nearest neighbor subdomain data exchanges
 535 start to creep up beyond the cost of the local constrained subdomain solve. Given that scaling

characteristics of these processes are well understood in the literature, the parallel speedups behave predictably well at scale on large computing machines tested.

The PyDIY based Python implementations for 1-, 2- and 3-dimensional problems have been shown to resolve large, complex solution profiles with strong gradient variations, even under decreasing subdomain sizes. Depending on the needs for visualization or in-situ analysis, choices on clamped or floating knots can be made with no modifications to the implementation. This scheme can also be used to achieve scalable high-order solution field transfers between component models, a process more generally referred to as *remapping* [16], by imposing constraints on the subdomain solvers to satisfy various metrics of interest [24] such as global conservation and monotonicity. The exploration of parallel MFA for such applications will be pursued in the future.

Acknowledgments. This work is supported by Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357, program manager Laura Biven. We gratefully acknowledge the computing resources provided on the Theta supercomputer operated by Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357, which was used to generate all the strong and weak scaling results. This research also used resources in the Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory to test and optimize performance of the drivers at scale.

554

REFERENCES

- 555 [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.
- 556 [2] M. BENZI, *Preconditioning techniques for large linear systems: a survey*, Journal of computational Physics, 182 (2002), pp. 418–477.
- 557 [3] Å. BJØRCK, *Numerical methods for least squares problems*, SIAM, 1996.
- 558 [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.
- 559 [5] 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.
- 560 [6] J. A. COTTRELL, T. J. HUGHES, AND Y. BAZILEVS, *Isogeometric analysis: toward integration of CAD and FEA*, John Wiley & Sons, 2009.
- 561 [7] L. B. DA VEIGA, D. CHO, L. F. PAVARINO, AND S. SCACCHI, *Overlapping schwarz methods for isogeometric analysis*, SIAM Journal on Numerical Analysis, 50 (2012), pp. 1394–1416.
- 562 [8] L. DALCIN, N. COLLIER, P. VIGNAL, A. CÔRTES, AND V. M. CALO, *Petiga: A framework for high-performance isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 308 (2016), pp. 151–181.
- 563 [9] 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.
- 564 [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.
- 565 [11] C. DE BOOR AND R. DEVORE, *Approximation by smooth multivariate splines*, Transactions of the American Mathematical Society, 276 (1983), pp. 775–788.
- 566 [12] 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.
- 567 [13] M. O. DEVILLE, P. F. FISCHER, AND E. H. MUND, *High-order methods for incompressible fluid flow*, vol. 9, Cambridge university press, 2002.
- 568 [14] J. DONGARRA ET AL., *MPI: A Message-Passing Interface Standard Version 3.0*, High Performance Computing Center Stuttgart (HLRS), (2013).
- 569 [15] W. DORNISCH AND S. KLINKEL, *Boundary conditions and multi-patch connections in isogeometric analysis*, PAMM, 11 (2011), pp. 207–208.
- 570 [16] J. K. DUKOWICZ AND J. W. KODIS, *Accurate conservative remapping (re zoning) for arbitrary lagrangian-*

- 586 *eulerian computations*, SIAM Journal on Scientific and Statistical Computing, 8 (1987), pp. 305–321.
 587 [17] E. EFSTATHIOU AND M. J. GANDER, *Why restricted additive schwarz converges faster than additive schwarz*,
 588 BIT Numerical Mathematics, 43 (2003), pp. 945–959.
 589 [18] I. GRINDEANU, T. PETERKA, V. S. MAHADEVAN, AND Y. S. NASHED, *Scalable, high-order continuity across block*
 590 *boundaries of functional approximations computed in parallel*, in 2019 IEEE International Conference on
 591 Cluster Computing (CLUSTER), IEEE, 2019, pp. 1–9.
 592 [19] C. HOFER AND U. LANGER, *Fast Multipatch Isogeometric Analysis Solvers*, PhD thesis, Johannes Kepler Uni-
 593 versity Linz, 2018.
 594 [20] W. JAKOB, J. RHINELANDER, AND D. MOLDOVAN, *pybind11—Seamless Operability Between C++ 11 and Python*,
 595 2017.
 596 [21] M. KAPL, G. SANGALLI, AND T. TAKACS, *Construction of analysis-suitable g1 planar multi-patch parameteri-
 597 zations*, Computer-Aided Design, 97 (2018), pp. 41–55.
 598 [22] J. LI AND Y. HON, *Domain decomposition for radial basis meshless methods*, Numerical Methods for Partial
 599 Differential Equations: An International Journal, 20 (2004), pp. 450–462.
 600 [23] P.-L. LIONS, *On the schwarz alternating method. i*, in First international symposium on domain decompositon
 601 methods for partial differential equations, vol. 1, Paris, France, 1988, p. 42.
 602 [24] V. S. MAHADEVAN, J. E. GUERRA, X. JIAO, P. KUBERRY, Y. LI, P. ULLRICH, D. MARSICO, R. JACOB,
 603 P. BOCHEV, AND P. JONES, *Metrics for intercomparison of remapping algorithms (mira) protocol applied
 604 to earth system models*, Geoscientific Model Development, 15 (2022), pp. 6601–6635.
 605 [25] N. MAI-DUY AND T. TRAN-CONG, *Approximation of function and its derivatives using radial basis function
 606 networks*, Applied Mathematical Modelling, 27 (2003), pp. 197–220.
 607 [26] F. MARINI, *Parallel Additive Schwarz Preconditioning for Isogeometric Analysis*, PhD thesis, Università degli
 608 Studi di Milano, 2015.
 609 [27] E. MERZARI, W. POINTER, A. OBABKO, AND P. FISCHER, *On the numerical simulation of thermal striping in
 610 the upper plenum of a fast reactor*, in International Congress on Advances in Nuclear Power Plants 2010,
 611 ICAPP 2010, 2010, pp. 595–603.
 612 [28] D. MOROZOV AND T. PETERKA, *Block-Parallel Data Analysis with DIY2*, in Proceedings of the 2016 IEEE
 613 Large Data Analysis and Visualization Symposium LDAV’16, Baltimore, MD, USA, 2016.
 614 [29] Y. S. NASHED, T. PETERKA, V. MAHADEVAN, AND I. GRINDEANU, *Rational approximation of scientific data*,
 615 in International Conference on Computational Science, Springer, 2019, pp. 18–31.
 616 [30] K. PAUL, C. ZIMMERMANN, T. X. DUONG, AND R. A. SAUER, *Isogeometric continuity constraints for multi-
 617 patch shells governed by fourth-order deformation and phase field models*, Computer Methods in Applied
 618 Mechanics and Engineering, 370 (2020), p. 113219.
 619 [31] T. PETERKA, S. YOUSSEF, I. GRINDEANU, V. S. MAHADEVAN, R. YEH, X. TRICOCHE, ET AL., *Foundations of
 620 multivariate functional approximation for scientific data*, in 2018 IEEE 8th Symposium on Large Data
 621 Analysis and Visualization (LDAV), 2018, pp. 61–71.
 622 [32] L. PIEGL AND W. TILLER, *The NURBS book*, Springer Science & Business Media, 2012.
 623 [33] J. O. RAMSAY, G. HOOKER, AND S. GRAVES, *Functional Data Analysis with R and MATLAB*, Springer Pub-
 624 lishing Company, Incorporated, 1st ed., 2009.
 625 [34] T. W. SEDERBERG, D. L. CARDON, G. T. FINNIGAN, N. S. NORTH, J. ZHENG, AND T. LYCHE, *T-spline simpli-
 626 fication and local refinement*, ACM transactions on graphics (TOG), 23 (2004), pp. 276–283.
 627 [35] B. SMITH, P. BJORSTAD, AND W. GROPP, *Domain decomposition: parallel multilevel methods for elliptic partial
 628 differential equations*, Cambridge university press, 2004.
 629 [36] A. ST-CYR, M. J. GANDER, AND S. J. THOMAS, *Optimized multiplicative, additive, and restricted additive
 630 schwarz preconditioning*, SIAM Journal on Scientific Computing, 29 (2007), pp. 2402–2425.
 631 [37] J. SUN, D. LENZ, H. YU, AND T. PETERKA, *Mfa-dvr: Direct volume rendering of mfa models*, 2022, <https://doi.org/10.48550/ARXIV.2204.11762>, <https://arxiv.org/abs/2204.11762>.
 632 [38] S. XU, W. JAHN, AND J.-D. MÜLLER, *Cad-based shape optimisation with cfd using a discrete adjoint*, Interna-
 633 tional Journal for Numerical Methods in Fluids, 74 (2014), pp. 153–168.
 634 [39] R. YOKOTA, L. A. BARBA, AND M. G. KNEPLEY, *Petrbf—a parallel o(n) algorithm for radial basis func-
 635 tion interpolation with gaussians*, Computer Methods in Applied Mechanics and Engineering, 199 (2010),
 636 pp. 1793–1804.
 637 [40] X. ZHANG, Y. WANG, M. GUGALA, AND J.-D. MÜLLER, *Geometric continuity constraints for adjacent NURBS
 638 patches in shape optimisation*, in ECCOMAS Congress, vol. 2, 2016, p. 9316.
 639 [41] W. ZHENG, P. BO, Y. LIU, AND W. WANG, *Fast B-spline curve fitting by L-BFGS*, Computer Aided Geometric
 640 Design, 29 (2012), pp. 448–462.

642 **Government License.** The submitted manuscript has been created by UChicago Argonne,
643 LLC, Operator of Argonne National Laboratory (“Argonne”). Argonne, a U.S. Department of
644 Energy Office of Science laboratory, is operated under Contract No. DE-AC02-06CH11357. The
645 U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevo-
646 cable worldwide license in said article to reproduce, prepare derivative works, distribute copies to
647 the public, and perform publicly and display publicly, by or on behalf of the Government. The
648 Department of Energy will provide public access to these results of federally sponsored research in
649 accordance with the DOE Public Access Plan.