Parallel Domain Decomposition Techniques Applied to Multivariate Functional Approximation of Discrete Data

1st Vijay S. Mahadevan
Mathematics and Computational Science Division
Argonne National Laboratory
Lemont, IL, 60439, USA
mahadevan@anl.gov

2nd Thomas Peterka
Mathematics and Computational Science Division
Argonne National Laboratory
Lemont, IL, 60439, USA
tpeterka@mcs.anl.gov

3rd Iulian Grindeanu
Mathematics and Computational Science Division
Argonne National Laboratory
Lemont, IL, 60439, USA
iulian@anl.gov

4th Youssef Nashed
Stats Perform
Chicago, IL, 60601, USA
youssef.nashed@statsperform.com

***Abstract*—Compactly expressing large-scale datasets through Multivariate Functional Approximations (MFA) can be critically important for analysis and visualization to drive scientific discovery. This paper presents scalable domain partitioning approach to compute MFA representation, by reducing the total work per task in combination with a nonlinear Schwarz-type, inner-outer iterative scheme for converging the interface data. For the underlying MFA, we utilize a tensor expansion of non-uniform B-spline (NURBS) basis in multi-dimensions to adaptively reduce the functional approximation error in the input data. While previous work on adaptive NURBS-based MFA has proven successful, the computational complexity of encoding large datasets on a single process can be severely prohibitive. Parallel algorithms for encoding domain-decomposed subdomains have had to rely on post-processing techniques to blend discontinuities across subdomains boundaries. In contrast, this paper presents a robust domain constrained parallel solution infrastructure to impose higher-order continuity directly on the MFA representation computed. We demonstrate effectiveness of the presented approach with an overlapping Restricted-type Additive Schwarz method (RASM) based domain decomposition solver, with a nonlinear accelerator such as ℓ-BFGS or Krylov (CGS, L-GMRes) to minimize the subdomain error residuals of the decoded MFA, and more specifically to recover continuity across non-matching boundaries. The analysis of the presented scheme for analytical and scientific datasets in 1-D and 2-D are also presented. Additionally, scalability studies are also shown for scientific 2-D datasets from a Climate problem to evaluate the parallel speedup of the algorithm on large computing clusters.**

***Index Terms*—functional approximation, domain decomposition, scalable methods, RASM**

I. Introduction

Large-scale discrete data analysis of various scientific computational simulations often require high-order continuous functional representations that have to be evaluated anywhere in the domain. Such expansions described as Multivariate Functional Approximations (MFA) [2] in arbitrary dimensions allow the original discrete data to be compressed, and expressed in a compact closed form, in addition to supporting higher-order derivative queries. One particular option is to use NURBS bases [3] for the MFA *encoding* of scientific data. Due to the potentially large datasets that need to be encoded into MFA, the need for computationally efficient algorithms (in both time and memory) to parallelize the work is critically important. It is also essential to guarantee that the solution smoothness in the reconstructed (or *decoded*) dataset is consistently preserved when transitioning from a single MFA block to multiple blocks during parallelization.

With these motivations, in the current paper, we utilize domain decomposition (DD) techniques [4] with data partitioning strategies to produce scalable algorithms to adaptively compute the MFA to reproduce a given dataset within user-specified tolerances. In such partitions, it is imperative to ensure that the continuity of the data across subdomain interfaces is maintained and is consistent with the degree of the underlying bases used in the NURBS-based MFA [5]. We present an iterative DD scheme with an outer Schwarz-type iterative scheme in order to ensure that continuity is enforced, and the overall error stays bounded when number of subdomains is increased (subdomain size decreases).

The paper is organized as follows. Section 2 summarizes the related work in using variations of the Schwarz scheme for scalable interpolation of data, and using constraints for recovering continuity along discontinuous patches. Section 3 provides details about the constrained optimization problem to resolve subdomain boundary discontinuities, along with the outer-inner DD-based solver setup to compute the continuous MFA. Next, the DD solver is applied to 1-D and 2-D analytical

Early Career Research Program, Department of Energy, US

problems to verify error convergence, and scalability of the hierarchical scheme in terms of outer iterations for decreasing subdomain sizes. The parallel scalability of the scheme is presented for scientific use-cases to compute the MFA within user-specified tolerances.

II. Related Work

DD techniques for parallel approximation of scattered data have been explored previously with Radial Basis Functions (RBF) [6], yielding good scalability and closely recovering the underlying profile. Overlapping multiplicative and additive Schwarz [7] iterative techniques for RBF [8] 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, has been shown to be scalable [9] with $O(N)$ computational complexity, as opposed to the typical $O(Nlog(N))$ complexity with traditional RBF reconstructions [10].

Application of these DD schemes and NURBS bases with isogeometric analysis (IGA) to high-fidelity modeling of non-linear Partial Differential Equations (PDEs) have enjoyed recent success [11], [12] at scale, but many 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 [13] and can be difficult to parallelize [14].

To our knowledge, using NURBS bases to compute the MFA in parallel, while maintaining higher-order continuity across subdomains has not been explored previously. To overcome some of these issues with discontinuities along NURBS patches, Zhang et al. [15] proposed to use a gradient projection scheme to constrain the value (G_0), the gradient (G_1), and the Hessian (G_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 blending techniques [1], 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. Moreover, selecting the amount of overlaps and resulting width of the blending region relies strongly on a heuristic, which can be problematic.

In contrast, we propose extensions to the constrained solvers used by Zhang et al. [15] and Xu et al. [16], and introducing a two-level, DD-based, parallel outer-inner iterative scheme to enforce the degree of continuity prescribed by the user. The outer iteration utilizes RAS [17] method, with an efficient inner subdomain solver using ℓ-BFGS as used in [18], or Krylov-type schemes with CGS, L-GMRes solvers, to minimize the decoded residual within acceptable error tolerances. Such a DD 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.

III. Approach

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) [4], [19] have been successfully employed to efficiently compute the solution of large, discretized PDEs in a scalable manner. In the current work, we utilize a data decomposition approach, with extensions to overlap subdomain data to create shared layers in order to ensure that higher-order continuity across domain boundaries are preserved. This is essential to generate consistent and accurate MFA representations in parallel. Extending the overlapping Schwarz solvers for PDE applications to MFA computation in data analysis, the amount of overlap in the data for MFA can directly affect the global convergence speed of the iterative scheme, and hence the scalability of the overall algorithm [20].

Note that the parallel constrained solver infrastructure should not amplify the approximation error, since the local decoupled subdomain solution is encoded accurately (within user tolerance) into piecewise continuous MFA representations. As the number of subdomains grows, the interface errors grow proportionally, thereby requiring larger number of outer iterations to converge. But at convergence, the overall global error in the decoded solution is still expected to be within user tolerance.

In this section, we first provide an illustrative example by formulating the constrained minimization problem to be solved and explain the hierarchical iterative methodology used in the current work. We also describe the inner subdomain solvers that are used to compute optimal NURBS-based MFA representations in parallel in order to maintain higher-order continuity across subdomain boundaries.

A. Solver Methodology

For the purpose of illustration and to explain the proposed solver methodology, consider a 1-D domain (Ω) with two subdomains ($n_s = 2$) as shown in Fig. (1), where Ω_1 and Ω_2 represent the subdomains that share an interface $\partial\Omega_{1,2}$. For generality, we also introduce an overlap layer Δ_1 and Δ_2 on each subdomain that share the decomposed data with its adjacent subdomain.

A p-th degree NURBS curve [3] is defined using the Cox-deBoor functions for each subdomain i as

$$\vec{C}(u) = \sum_{i=0}^{n} R_{i,p}(u)\vec{P}(i), \quad \forall u \in \Omega \tag{1}$$

$$R_{i,p}(u) = \frac{N_{i,p}(u)W_i}{\sum_{i=0}^{n} N_{i,p}(u)W_i} \tag{2}$$

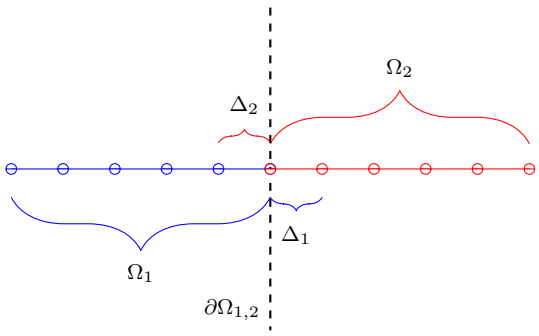


Fig. 1. 1-D parallel partitioned domain

where $R_{i,p}(u)$ are the piecewise rational functions with $\vec{P}$ control points, W_i are the control point weights, and p-th degree B-spline bases $N_{i,p}(u)$ defined on a knot-vector U. Exact high-order derivatives of these NURBS basis defined in Equation (2) can also be evaluated without any approximation errors at the control point locations using the Cox-deBoor recurrence relations [2].

Given a set of input points Q that need to be encoded into a MFA, with the weights $W = 1$ for simplicity, the 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. The LSQ solver computes the optimal control point solution $\vec{P}$ that minimizes the objective $\left\|Q - R\vec{P}\right\|$ [3], which is the local subdomain residual error between the given input data and decoded MFA representation.

In the current work, we use the unconstrained LSQ solver using Cholesky decomposition as the method of choice to compute the control point DoFs, when adaptively resolving the features in the input data through knot insertion and removal [21]. Once the local subdomain resolution is sufficiently within user-specified tolerance levels, the resulting global MFA representation is piecewise discontinuous at subdomain boundaries. The next step is then to apply global constrained solvers to minimize the continuity error in order to recover higher derivatives iteratively as needed.

The global constrained minimization problem for the two subdomain case shown in Fig. (1) can be written as

$$\left[\begin{array}{c|c} R_1 & \lambda_{1,2} \\ \hline \lambda_{2,1} & R_2 \end{array}\right] \left[\begin{array}{c} \vec{P}_1 \\ \vec{P}_2 \end{array}\right] = \left[\begin{array}{c} Q_1 \\ Q_2 \end{array}\right] \tag{3}$$

The diagonal operators R_1 and R_2 are the piecewise rational functions that minimize the local subdomain residuals, while the off-diagonal blocks $\lambda_{1,2}$ and $\lambda_{2,1}$ represent the coupling terms between the subdomains at $\partial\Omega_{1,2}$. 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.

The coupling blocks $\lambda_{i,j}$ can be viewed as Lagrange multipliers that explicitly couple the control point DoFs across a subdomain interface such that continuity is preserved in a weak sense [3]. We apply the RAS scheme to tackle this system of global equations shown in Equation (3), where the coupled terms $\lambda_{i,j}$ utilize *lagged* control point data from adjacent subdomains. It is easy to see that through a process of Gaussian elimination that the coupled operator dependency in each subdomain can be removed, resulting in a set of constrained nonlinear equations to be solved iteratively. In a continuous sense, this can also be viewed as a block-Jacobi solver applied to Equation (3), and hence popularly called the Jacobi-Schwarz DD method [17] that is equivalent to the RAS method.

In this scheme, the coupled data $\vec{P}_2$ and $\vec{P}_1$ for subdomains Ω_1 and Ω_2 respectively at $\partial\Omega_{1,2}$ are exchanged simultaneously before the local domain solves are computed. Since the exchanged constraint data is lagged at the previous iterate, the convergence rate in comparison to the more expensive multiplicative Schwarz variants [4] is slower. However, the key advantage to using RAS in the current methodology is that it only requires nearest neighbor exchange of data, which keeps communication costs bounded as number of subdomains increases [7], [17], while interlacing recomputation of the constrained control point solution. Note that in a RAS outer iterative scheme, nearest neighbor exchanges can be performed compactly per dimension and direction, thereby minimizing communication costs and eliminating global collectives. The volume of messages exchanged however depends on several factors.

1) **Continuity**: The degree of continuity (up to $p - 1$) determines the stencil needed to enforce the constraints on either side of the interface
2) **Overlap**: The amount of overlap (Δ) determines the number of coupled data layers to be communicated between neighboring domains

At convergence, the interface data at $\partial\Omega_{1,2}$ will satisfy the higher-order continuity prescriptions specified by the user thereby guaranteeing smoothness in the span of G_0 to G_{p-1}. The illustration in Fig. (1), and the methodology description in this section can be generalized and extended to arbitrary dimensions, and will serve as the basis to describe the local subdomain solvers in the following subsections.

We next present two variations of the subdomain solvers that are used in this study.

B. Constrained Nonlinear Solver

In order to ensure continuity across NURBS patches, Zhang et al. [15] evaluated the constraint matrix at test points along the interface curve. These computations require a Singular Value Decomposition (SVD) solve at every interface and can become prohibitively expensive as dimensionality increases. Alternatively, we can use a nonlinear minimizer, directly applied to the system in Equation (3), and adding a penalty term to the constrained DoFs represented by the coupling blocks $\lambda_{i,j}$. Such a system can be expressed as a minimization

problem in each subdomain with the following objective functional.

$$\begin{aligned} \vec{E}_i(\Omega_i) &= Q_i - R_i\vec{P}_i, \quad \forall i \in [1, n_s] \\ \vec{E}_c(\Delta_i) &= \sum_{\partial\Omega_{i,j}} \left[\vec{P}_i(\Delta_i \cap \Omega_j) - \vec{P}_j(\Delta_i \cap \Omega_j)\right] \\ \vec{E}_i(\Omega_i \bigcup \Delta_i) &= \vec{E}_i(\Omega_i) + \epsilon\vec{E}_c(\Delta_i), \end{aligned} \tag{4}$$

where ϵ is the boundary penalty term. In our experiments, a large value of $\epsilon = 10^7$ has yielded very good performance with favorably monotonic error reduction in the constraint residuals $r_c(\Delta_i)$. Note that $\vec{E}_i(\Omega_i)$ is in the decoded space, while the constraint residual $\vec{E}_c(\Delta_i)$ is in the control point space. Hence in this formulation, we apply the constraints directly on control points, while trying to minimize the overall decoding error for the MFA.

The augmented solution for each local subdomain after nearest neighbor exchange contains control points and corresponding derivative data to constrain DoFs on both the left and right interfaces in 1-D. This definition extends naturally to higher dimensions, with the penalized constraints on the boundary terms evaluated as a loop over all shared interfaces to compute the net residual.

To compute the solution to the minimization problem in Equation (4), we can make use of Sequential Least SQuares Programming (SLSQP) solver with equality constraints, or apply the limited memory version of Broyden-Fletcher-Goldfarb-Shannon (ℓ-BFGS) algorithm when the dimensionality increases, and memory requirements become large. Such solvers have already been proven effective for converging such constrained continuity problems with B-Splines [18]. An even more efficient method is to reformulate the objective function as a root finding problem, and applying Krylov accelerators or Anderson mixing to compute the optimally constrained solutions, with much lower computational complexity and superior convergence properties. The minimized control point solution is achieved when the interface solutions match on all $\partial\Omega_{i,j} \in \Omega$ to drive the nonlinear residuals in Equation (4) to within solver relative tolerance of 10^{-12}.

Constraints on Decoded Data: Instead of imposing the constraints in the control point space, alternatively, we can utilize the expansion of the MFA in input point space directly to minimize the decoded residual $E = Q - R\vec{P}$. The constraints in each subdomain then are essentially the jump terms between the decoded data on Ω_i and Ω_j shared at the interface $\partial\Omega_{i,j}$.

There are some advantages to this approach compared with imposing the interface constraints in the control point space, even though the potential volume of data to be communicated between subdomains is generally much larger since it scales with size of Q. They are listed below.

1) Subdomain residuals and boundary constraints are both in the decoded space, and hence no explicit need for a residual projection (encoding) or penalty term in Equation (4),
2) No explicit need for imposition of higher order derivative constraints, especially in the context of non-conforming adaptivity across subdomain interfaces, since the grid of input points is fixed,
3) Natural extensions within the ASM iterative scheme to generate overlapping variants.

When the solver is setup to use the error residuals in the actual decoded space, the natural data decomposition with overlap Δ can yield good improvements in both time and accuracy [20]. This follows the effectiveness of the overlapping additive-Schwarz preconditioning schemes in the context of linear algebra problems for PDEs [4] [17]. A demonstration of the improved efficiency and scalability in terms of iteration convergence is shown for a 1-D problem in the Section (V). The growing message size for use in overlapping RAS method when solving higher dimensional problems is a concern, and it is a topic of ongoing research to find ways to minimize the communication overheads.

IV. Implementation

The presented methods in this manuscript have been primarily implemented in Python using bindings for the DIY C++ library. DIY [22] is a programming model and runtime for block-parallel analytics on distributed-memory machines, built on MPI-3 [23]. Rather than programming for process parallelism directly in MPI, the programming model in DIY is based on block parallelism: data are decomposed into subdomains called blocks; blocks are assigned to processing elements (processes or threads); computation is described over these blocks, and communication between blocks is defined by reusable patterns. The same DIY program consisting of a block-parallel decomposition can be run on different numbers of MPI processes: it is the job of the DIY runtime to map between blocks and processes. The Python bindings to DIY, pyDIY [24], are a recent development that utilize PyBind11 [25] and MPI4Py [26] to expose the interfaces in the C++ library. In our implementation, PyDIY and DIY exclusively manage the data decomposition, including specifications to share an interface $\partial\Omega_{i,j}$ and ghost layers that represent overlaps Δ.

The overall approach is sketched in Algorithm (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. We also include adaptive knot placements to approximate the input data by introducing control point DoFs where needed to capture solution data variations.

The control point solution from this adaptive, 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 in a 2-level nested loop: the outer loop is driven by RAS iterations described in Section (III-A), while the inner loop executes the individual subdomain MFA solves simultaneously to compute the control point solution to the nonlinear optimization problem in Equation (4) with ℓ-BFGS or Krylov methods. At the start of each outer ASM iteration, the control point constraints are exchanged between

neighboring blocks in a regular nearest-neighbor communication pattern. This is sufficient to update the constraints $\vec{P}(\Delta)$ for the inner subdomain solves. The DIY send and receive data exchange API enqueues and dequeues the constraint data to neighboring blocks based on the parallel data decomposition. Depending on whether the MFA residual or the decoded residual norms are to be minimized, a subdomain solver is chosen to drive the nonlinear residual for the local block within user-specified tolerances.

At the end of the outer iterative loop, we check for convergence across all subdomains, by evaluating whether the maxima of the L_∞ error across all subdomains (n_s) of the ASM update vector is within user-specified tolerance. If this condition is satisfied, the MFA computation is stopped. On convergence, the result 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.

Algorithm 1: Hierarchical DD MFA Solver

```
decompose domain into blocks with DIY
solve local MFA without constraints
// Outer ASM loop;
iASM ← 0                                   // ASM iteration counter
do
  P⃗(Ω ∩ Δ) → enqueue outgoing constraints
  exchange constraints with neighbor blocks
  P⃗(Δ) ← dequeue incoming constraints
  // Inner ℓ-BFGS/Krylov minimization solver
  P⃗_i ← solve adaptive local MFA with constraints
  E_i ← compute decoded local error
  δP⃗(Ω_i) ← P⃗_iASM(Ω_i) − P⃗_iASM−1(Ω_i)
  dPMax ← ‖δP⃗(Ω_i)‖_∞
  iASM++
while dPMax > 10^−10 and iASM < nMaxASM
// Store subdomain solution data;
Write MFA to disk for analysis and visualization
```

V. Results

To demonstrate the effectiveness of the nested iterative algorithm, we utilize both analytical closed form functionals and scientific datasets in both 1-D and 2-D obtained from high-fidelity simulations. A brief description of the problem cases that are used as test datasets is given below.

1) Synthetic data: sinc(x)=$\frac{sin(x)}{x}$ functional on $\Omega \in [-4, 4]$
   - 1-D: Q(x) = sinc(x) + sinc(2x-1) + sinc(3x+1.5),
   - 2-D: Q(x,y) = sinc($(x+1)^2 + (y-1)^2$) + sinc($(x-1)^2 + (y+1)^2$)
2) S3D: A 3-D turbulent combustion dataset generated by an S3D simulation [27] of fuel jet combustion in the presence of an external cross-flow.
3) CESM: A 2-D Community Atmosphere Model climate model dataset on a sphere with 3600x1800 resolution

A. 1-D Results

1) Adaptive Error Convergence and Verification: The 1-D implementation of the ASM iterative solver was applied to the analytical sinc problem. An initial input problem size of 500 points was evaluated as a test case, and MFA representation for the resulting profile was computed on 4 subdomains. The solution profile, along with the control point locations from the adaptive computation are shown in Fig. (2a). We utilized the residual definition shown in Equation (4) with a G_1 constraint to be satisfied along all internal subdomain interfaces. The Fig. (2b) and Fig. (2c) present the zoomed in reconstruction of the approximation near the interface between Ω_2 and Ω_3 before, and after the ASM iterations are converged. The initially discontinuous solutions from independent, adaptive, unconstrained subdomain solves are converged to satisfy the constraints after only 3 RAS iterations, with nearest neighbor exchange of control point data at the interfaces to recover G_1 continuity.

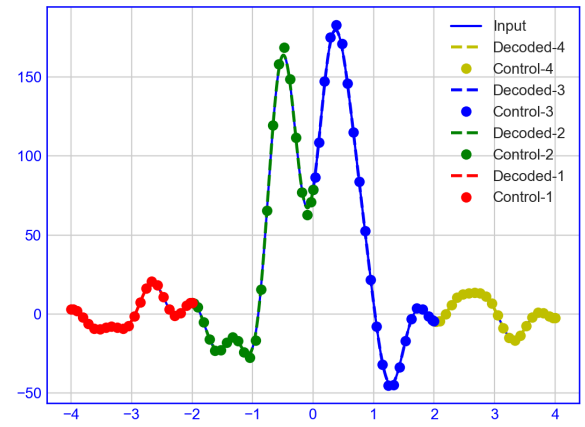
2) Overlap Experiments in 1-D: In this 1-D experiment on the closed form sinc function shown in Section (V), we utilize the decoded residual minimization with the RAS-Krylov solver combination, and increase the amount of overlap in the input points to look at the convergence speed of the RAS method. We measured the global L_2 error convergence along with the total cost in terms of outer iteration to compute the constrained control point locations. In this analysis, adaptivity in the individual subdomains was explicitly turned off so as to maintain a constant ratio of input points to control points per subdomain, as overlap region is extended in the subdomains on both the left and right sides.

Fig. (3) shows the global L_2 norm of the net error in the decoded residual, as the number of subdomains are increased. With no overlap regions, where only the interface data is shared, the total ASM iterations needed to satisfy convergence, and the net error increases linearly with n_s. As the overlap size becomes larger, for $\Delta = 32$ and $\Delta = 64$, the error growth and the number of iterations to achieve convergence remain much more bounded in comparison with the non-overlapping case. The implications of this particular result closely matches the ASM preconditioning theory often applied in PDE solvers [4], [17], [19]. It also shows that the RAS iterative scheme can be accelerated more efficiently, and scalably as a solver without a prohibitive linear complexity of O(n_s).

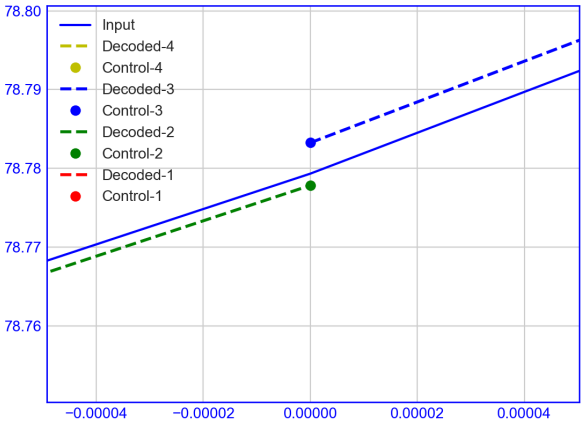
B. 2-D Results

Using the 2-D sinc analytical function, the error convergence for different NURBS bases degree p was measured. As the number of control point DoFs are increased, the global norm of the errors are reduced consistently in the current implementation. Next we tested the spatial adaptivity in 2-D on a scientific dataset to ensure that the hierarchical iterative scheme with RAS is able to capture strong solution variations, and still ensure continuity across subdomain interfaces.

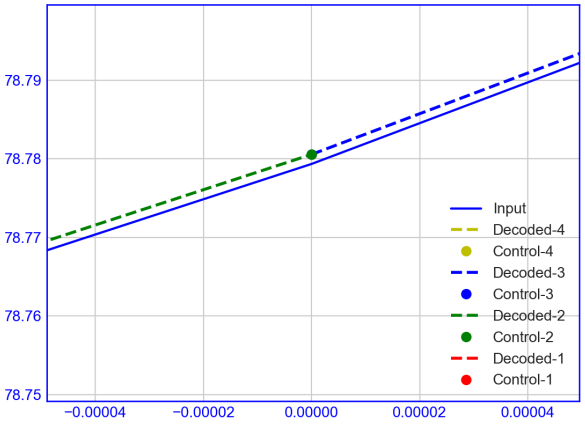
1) S3D Dataset: Spatial Adaptivity: Additionally, using adaptive resolution based on a-posteriori gradient estimates, numerical errors in strongly varying solution profiles were



(a) sinc function profile in 1-D



(b) Zoomed in image of $\Omega_{2,3}$ at (a) iASM=0



(c) Zoomed in images of $\Omega_{2,3}$ at iASM=3

Fig. 2. 1-D analytical sinc dataset with 501 input points: adaptive MFA on 4 subdomains with 10^{-4} tolerance and $p = 2$

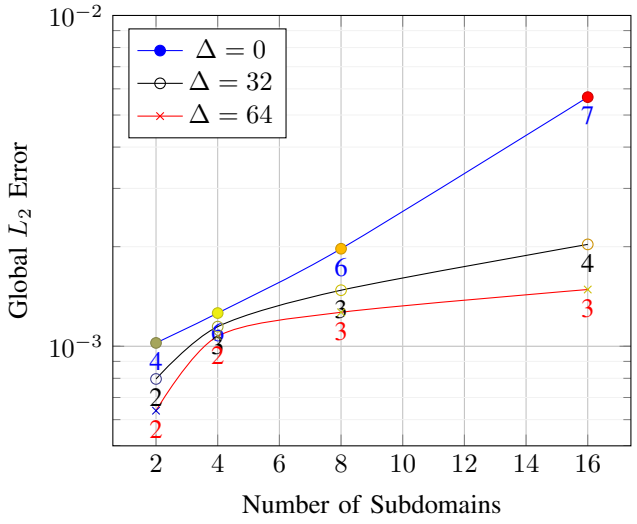


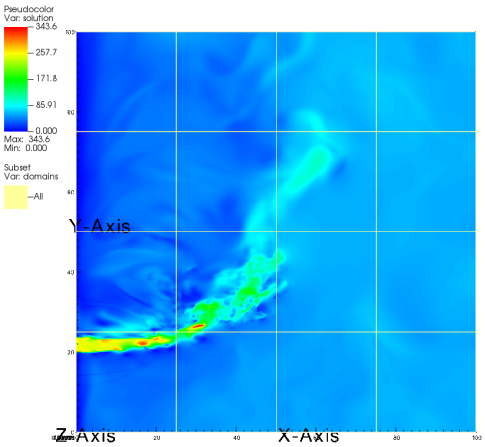
Fig. 3. 1-D sinc problem with 1025 points: Accuracy convergence and number of iterations of ASM solver with overlap variations, as n_s increases

reduced to within user-specified tolerances. A sample solution for the 2-D slice of the S3D dataset on 9 subdomains is shown in Fig. (4), along with the error profiles at the first iteration in Fig. (4b) and on convergence of the RAS iterations in fig:s3d-adaptive-2d-c.

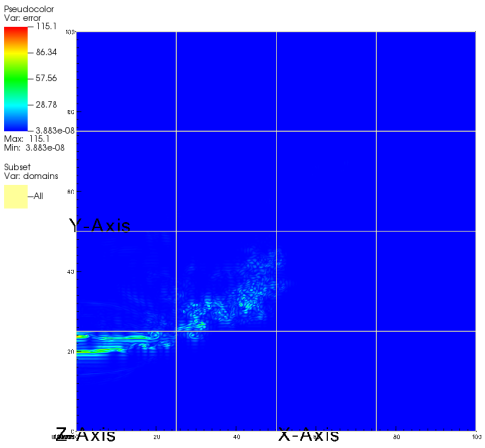
The errors in the decoupled LSQ solve shown in Fig. (4b) has relatively larger errors in the subdomains with strong gradient changes. When including knot adaptivity combined with RAS iterative scheme, the errors in the reconstructed MFA data are reduced to within user-specified relative tolerances of 10^{-2}. This result demonstrates the convergence of the proposed scheme to adaptively resolve a complex scientific solution profile, even on a small number of subdomains ($n_s = 16$).

2) Parallel Scalability: The parallel scalability of the implemented RAS iterative solver for ensuring continuity across block boundaries was measured on the 2-D CESM dataset, whose solution profile in a uniform latitude-longitude grid is shown in Fig. (5). Our experiments were executed on the Bebop cluster at Argonne National Laboratory LCRC (Laboratory Computing Resource Center). Bebop has 1024 public nodes, with Intel Broadwell or Intel Knights Landing processors, with 128 GB memory per Broadwell node, and 104 GB per KNL node. It has an Omni-Path Fabric Interconnect. Broadwell processors have 36 cores, while KNL processors have 64 cores. We primarily used the KNL processors for the scalability tests presented below.

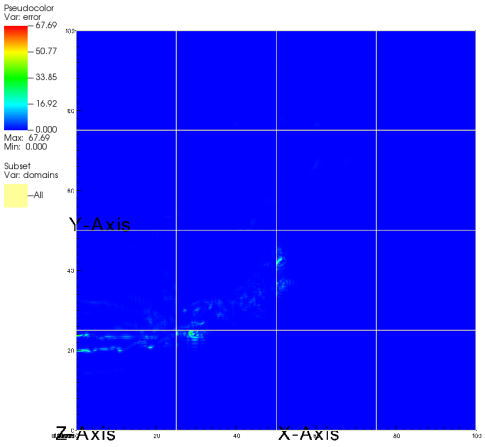
A strong scalability test was performed on the Bebop cluster with the CESM dataset, using 32 subdomains in the X-direction and 128 subdomains in the Y-direction. With this fixed DD and a uniformly placed, 100 control points per subdomain (10x10) resolution, parallel jobs were executed to compute the MFA. The Python driver utilized DIY to handle block decompositions and rank assignment, as the total number of processes used in the parallel run was varied from



(a) S3D dataset profile



(b) Error profile before RAS iteration or adaptivity



(c) Error profile after RAS iteration converges

Fig. 4. 2-D slice of the S3D dataset: profile and adaptive error resolution with tolerance=10^{-2}

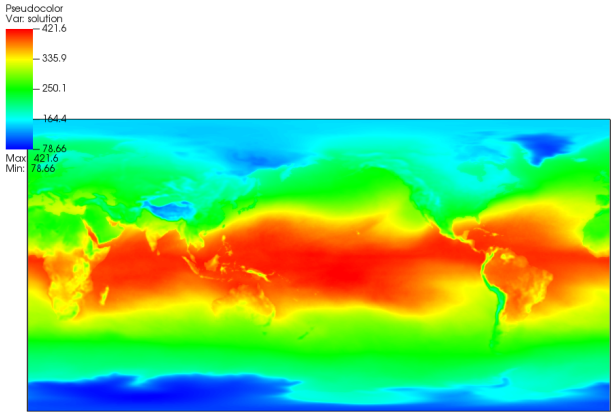


Fig. 5. 2-D CESM climate dataset

[1,2,. . .2048]. We measured the overall time for the initial subdomain solves, and the consequent RAS iteration cycle for $nMaxASM = 5$. The time to compute the MFA in parallel is shown for this strong scalability experiment in Fig. (6).

As expected, the hierarchical iterative scheme with RAS-Krylov combination shows excellent scalability for the chosen dataset, and the overall time to compute the MFA was reduced from nearly 7000s on a single process, to 9.5s on 2,048 processes, while ensuring both G_0, and G_1 continuity in the domain interfaces. The ratio of computational work in local subdomain solves in comparison with communication time in nearest-neighbor exchanges gradually increases as subdomain size shrinks. This is due to the fact that global information on smaller subdomains take multiple iterations to propagate, and hence using overlaps for such large problems would be a recommended extension in the future to improve overall scalability of the algorithm. For this intermediate size problem, the strong scaling efficiency of the presented scheme is around 35% on 2,048 processes, which dropped off from 65% parallel efficiency at 256 processes.

Another key verification performed during this strong scaling test is to ensure that the local subdomain errors computed in serial and on different process counts remain the same at convergence. This verification is important to reiterate the fact that the approximation error due to the constrained solves to recover higher-order continuity does not significantly affect the error metrics for the MFA.

VI. Conclusion

We have presented a scalable DD approach to tackle the issue of discontinuous MFA representations when performing the computations in parallel. The Restricted Additive Schwarz (RAS) method is a natural algorithmic fit for data analysis problems to create efficient MFA solutions in parallel. Through the use of Schwarz-based iterative schemes, combined with ℓ-BFGS or Krylov solvers for local subdomain solves, the hierarchical iterative technique was shown to be robust in converging to the compressed functional representation of the given data, without sacrificing the approximation accuracy. Combining

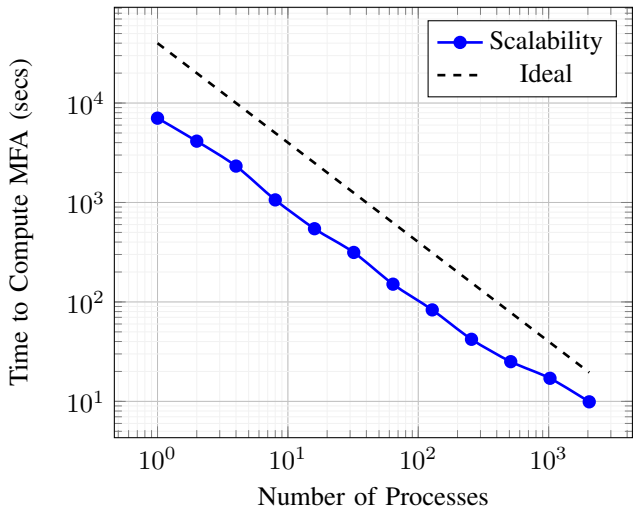


Fig. 6. Strong scalability of the 2-D CESM Problem with 4,096 subdomains and 100 control points/subdomain

NURBS-based adaptivity with a-posteriori error measures, and ensuring higher-order continuity across block boundaries, a scalable infrastructure has been presented. The PyDIY based Python implementations for 1-D and 2-D problems have been shown here to resolve complex solution profiles and gradient variations, even under decreasing subdomain sizes. The use of overlap layers can definitely improve the overall MFA accuracy and convergence speed of the RAS algorithm as demonstrated for some 1-D analytical problems, but at a slightly higher cost per iteration. Further analysis on the effect of overlap regions for RAS in higher-dimensional settings needs to be explored.

The strong scalability of the algorithm was also demonstrated for a reasonably large 2-D climate dataset with 6.5M data points. The outer-inner iterative scheme provides very good strong scalability up to 1,024 processes after which the performance degrades slightly due to the lack of local computation work in comparison with the latency and communication costs associated with the nearest-neighbor exchanges of constraint data.

A more natural way to ensure continuity across NURBS patches would be to use T-splines [28], which are specifically designed for merging higher-dimensional surfaces with non-matching knot locations. The implementation of T-splines for adaptivity in the context of MFA is currently being explored, and the presented ASM-based solver approach can still be used to impose constraints across subdomain patch boundaries, while local constraints within each block can be imposed with appropriate T-spline basis modifications.

Another extension within this infrastructure would be to utilize a multilevel MFA representation that hierarchically refines the NURBS approximation at each level by decreasing the number of subdomains used. Such computations involving multilevel MFA has not been attempted previously. With appropriate choices of prolongation and restriction operators, the RAS iterative scheme can be used with a multilevel subdomain solver to efficiently produce accurate and compact functional approximation of given data, in higher dimensions.

The methodology and experiments presented in the paper are implemented primarily in Python. These algorithms will be ported to the more production-ready C++ MFA codebase [29] in the future. The new implementations will directly make use to the DIY C++ library, and all its features for data decomposition along with fast local subdomain solvers without overheads typically associated with Python codes. We expect that the RAS hierarchical solver implementation in C++ will allow for scalable and efficient application of NURBS-based MFA to much larger problems, without losing solution continuity or approximation accuracy.

Acknowledgment

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 Bebop, a high-performance computing cluster operated by the Laboratory Computing Resource Center (LCRC) at Argonne National Laboratory.

References

[1] I. Grindeanu, T. Peterka, V. S. Mahadevan, and Y. S. Nashed, "Scalable, high-order continuity across block boundaries of functional approximations computed in parallel," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2019, pp. 1–9.

[2] C. de Boor and R. DeVore, "Approximation by smooth multivariate splines," *Transactions of the American Mathematical Society*, vol. 276, no. 2, pp. 775–788, 1983.

[3] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 2012.

[4] B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition: parallel multilevel methods for elliptic partial differential equations*. Cambridge university press, 2004.

[5] T. Peterka, S. Youssef, I. Grindeanu, V. S. Mahadevan, R. Yeh, X. Tricoche *et al.*, "Foundations of multivariate functional approximation for scientific data," in *2018 IEEE 8th Symposium on Large Data Analysis and Visualization (LDAV)*, 2018, pp. 61–71.

[6] N. Mai-Duy and T. Tran-Cong, "Approximation of function and its derivatives using radial basis function networks," *Applied Mathematical Modelling*, vol. 27, no. 3, pp. 197–220, 2003.

[7] A. St-Cyr, M. J. Gander, and S. J. Thomas, "Optimized multiplicative, additive, and restricted additive schwarz preconditioning," *SIAM Journal on Scientific Computing*, vol. 29, no. 6, pp. 2402–2425, 2007.

[8] J. Li and Y. Hon, "Domain decomposition for radial basis meshless methods," *Numerical Methods for Partial Differential Equations: An International Journal*, vol. 20, no. 3, pp. 450–462, 2004.

[9] R. Yokota, L. A. Barba, and M. G. Knepley, "Petrbfa parallel o (n) algorithm for radial basis function interpolation with gaussians," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 25-28, pp. 1793–1804, 2010.

[10] 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*, vol. 22, no. 5, pp. 1717–1740, 2001.

[11] F. Marini, "Parallel additive schwarz preconditioning for isogeometric analysis," 2015.

[12] 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*, vol. 308, pp. 151–181, 2016.

[13] M. Kapl, G. Sangalli, and T. Takacs, "Construction of analysis-suitable g1 planar multi-patch parameterizations," *Computer-Aided Design*, vol. 97, pp. 41–55, 2018.

[14] C. Hofer and U. Langer, "Fast multipatch isogeo-metric analysis solvers," 2018.
[15] X. Zhang, Y. Wang, M. Gugala, and J.-D. Müller, "Geometric continuity constraints for adjacent nurbs patches in shape optimisation," in *ECCOMAS Congress*, vol. 2, 2016, p. 9316.
[16] S. Xu, W. Jahn, and J.-D. Müller, "Cad-based shape optimisation with cfd using a discrete adjoint," *International Journal for Numerical Methods in Fluids*, vol. 74, no. 3, pp. 153–168, 2014.
[17] E. Efstathiou and M. J. Gander, "Why restricted additive schwarz converges faster than additive schwarz," *BIT Numerical Mathematics*, vol. 43, no. 5, pp. 945–959, 2003.
[18] W. Zheng, P. Bo, Y. Liu, and W. Wang, "Fast b-spline curve fitting by l-bfgs," *Computer Aided Geometric Design*, vol. 29, no. 7, pp. 448–462, 2012.
[19] P.-L. Lions, "On the schwarz alternating method. i," in *First international symposium on domain decomposition methods for partial differential equations*, vol. 1. Paris, France, 1988, p. 42.
[20] 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*, vol. 10, no. 5, pp. 1053–1061, 1989.
[21] W. Li, S. Xu, G. Zhao, and L. P. Goh, "Adaptive knot placement in b-spline curve approximation," *Computer-Aided Design*, vol. 37, no. 8, pp. 791–797, 2005.
[22] D. Morozov and T. Peterka, "Block-Parallel Data Analysis with DIY2," in *Proceedings of the 2016 IEEE Large Data Analysis and Visualization Symposium LDAV'16*, Baltimore, MD, USA, 2016.
[23] J. Dongarra *et al.*, "MPI: A Message-Passing Interface Standard Version 3.0," *High Performance Computing Center Stuttgart (HLRS)*, 2013.
[24] D. Morozov, "pyDIY: Python Bindings for the DIY Library for Block-Parallel Data Analysis," https://gitlab.kitware.com/mrzv/pydiy.
[25] W. Jakob, J. Rhinelander, and D. Moldovan, "pybind11–Seamless Operability Between C++ 11 and Python," 2017.
[26] L. D. Dalcin, R. R. Paz, P. A. Kler, and A. Cosimo, "Parallel Distributed Computing using Python," *Advances in Water Resources*, vol. 34, no. 9, pp. 1124–1139, 2011.
[27] 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*, vol. 2, no. 1, p. 015001, 2009.
[28] T. W. Sederberg, D. L. Cardon, G. T. Finnigan, N. S. North, J. Zheng, and T. Lyche, "T-spline simplification and local refinement," *ACM transactions on graphics (TOG)*, vol. 23, no. 3, pp. 276–283, 2004.
[29] T. Peterka. [Online]. Available: https://bitbucket.org/tpeterka1/mfa.git