Highlights

Compression of Tokamak Boundary Plasma Simulation Data Using a Maximum Volume Algorithm for Matrix Skeleton Decomposition

Sebastian De Pascuale, Kenneth Allen, David L. Green, Jeremy D. Lore

- CUR matrix decomposition shows improved asymptotic computational complexity over SVD.
- SOLPS-ITER data is compressed at orders of magnitude faster compute time than SVD.
- CUR technique provides an interpretable framework for archiving simulation data.

Compression of Tokamak Boundary Plasma Simulation Data Using a Maximum Volume Algorithm for Matrix Skeleton Decomposition

Sebastian De Pascuale[a], Kenneth Allen[b], David L. Green[a,c], Jeremy D. Lore[a]

[a] *Oak Ridge National Laboratory, 1 Bethel Valley Rd, Oak Ridge, 37830, Tennessee, United States*
[b] *University of Georgia, 210 S Jackson St, Athens, 30602, Georgia, United States*
[c] *Commonwealth Scientific and Industrial Research Organization, Sydney, New South Wales, Australia*

Abstract

This report demonstrates satisfactory data compression of SOLPS-ITER simulation output ranging from 2D fields, 1D profiles, and 0D scalar variables with a novel matrix decomposition approach. The singular value decomposition (SVD) scales poorly for large matrix sizes and is unsuited to the application on high dimensional data common to fusion plasma physics simulation. We employ the columns-submatrix-rows (CUR) matrix factorization technique in order to compute a low-rank approximation up to two orders of magnitude faster than the SVD, but within a nominal L2-norm relative error of $\epsilon = 10^{-2}$. In addition, the CUR approach maintains the original format of the data, in its extracted columns and rows, allowing for interpretable data storage at the original resolution of the simulation. We utilize an iterative algorithm to compute the CUR decomposition of simulation output by maximizing the volume, or linearly independent information content, of a low-rank submatrix contained within the data. Experiments over $n \times n$

randomized test matrices with embedded rank-deficient features show that this maximum volume implementation of CUR matrix approximation has reduced asymptotic computational complexity on the order of n compared to the SVD, which scales approximately as n^3. These results show that the CUR technique can be used to effectively select time step snapshots (columns) of over 140 SOLPS-ITER output variables and the associated discretized coordinate timeseries (rows) allowing for reconstruction of the complete simulation dynamics.

Keywords: SOLPS-ITER, scrape-off-layer, CUR matrix decomposition, low-rank matrix approximation, dimensionality reduction, data compression

1. Introduction

The singular value decomposition (SVD) has found wide application to several important problems in fusion plasma physics. It has been used in stellarator design to extract physically relevant dynamics from experiment (1), analyze turbulent transport in tokamaks through a separation of scales (2), (3), filter noise from particle-in-cell kinetic simulations (4), and identify damped eigenmodes in gyrokinetic simulations (5), (6). The utility of the SVD is derived from its theoretical determination of matrix rank, the maximum number of linearly independent column or row vectors in the assembled data. With the SVD, features in a data set are identified with respect to a least-squares factorization of an $m \times n$ matrix into rank r components ordered by the coverage of variance in the data (7), (8). This “best-fit” interpretation of the SVD product components comes with a challenging cost, the algorithms employed typically have a computational complexity that scales as $\mathcal{O}(\min(m^2n, nm^2))$ (9), (10). Several investigations have circumvented this restriction in the attempt to effectively compress high degrees-of-freedom plasma physics simulation data into a storable format for recording MHD magnetic fields (11) and the evolution of the gyrokinetic distribution function (12), (13) by opting to apply the SVD only over subsets of a matrix dimension to lower the scope of the problem.

This paper pursues further development of data compression using matrix decomposition in circumstances where fusion plasma physics simulation have high dimensionality that is unfavorable to the application of the SVD. The primary aim of an alternative approach is to efficiently balance the fidelity of matrix reconstruction error with the degree of data compression.

We attempt to overcome the approximately cubic scaling of the SVD with a technique that extracts actual columns and rows within a matrix, which are found to be quasi-linearly independent, to perform faster factorization of simulation-based data. The entries at the intersection of these components form a submatrix that represents important features extracted from within the data. This columns-submatrix-rows (CUR) approach was first adapted from a modified quasi-Gram-Schmidt procedure to obtain compressed factorizations of rank-deficient matrices (14), (15). Other schemes randomly sample over a probability distribution of the normalized column space (16), (17), leading to preferential selection of components. The works of (18), (19), & (20) have shown additive error bounds for hybridized CUR when using a statistical leverage score based on the span of top right singular vectors from the SVD. We prioritize computational efficiency and follow the "pseudoskeleton" approximation developed by (21), (22), & (23). The prescribed CUR decomposition calculates a "maximum volume" submatrix by rapidly searching for close to uncorrelated columns and rows of matrix data up to a target number (pseudo-rank) (24).

The data-driven aspect of this work also offers the distinct advantage of preserving exact quantities in the original format of the simulation data to construct a low-rank matrix approximation. Compared to the SVD, which reduces dimensionality by projecting matrix data onto a new orthonormal basis, the CUR decomposition avoids attaching physical meaning to the product components beyond what the data already contain (9), (20). For example, application of matrix factorization to a tensor requires transformation of a discretized scalar field produced by simulation into an "unfolded," or stacked dimension, matrix of the data, often arranged in spatial by temporal entries (25), (26). In this reorganized representation, the columns and rows extracted by CUR provide a “maximal” representative set of the coordinates and time step states of the solution at the original resolution of the simulation. Here the components of CUR can be used to identify coordinate timeseries (as synthetic diagnostics) and output time step (as snapshot cadence) that together yield complete state information via matrix reconstruction. We extend this concept to the application of CUR decomposition as a data compression method for archival preservation of tokamak plasma boundary simulations with SOLPS-ITER (Scrape Off Layer Plasma Simulator), a state-of-the-art multi-fluid plasma and kinetic neutrals transport solver that has been benchmarked against decades of ITER development research (27), (28), (29).

SOLPS-ITER is computationally expensive to run, placing high value on the many possible output quantities of simulation. There can be up to 140 output variables, depending on the number of plasma species, that span thousands of time steps and hundreds of million floating point numbers. Instead of pruning the simulation calculation, the CUR may be imposed to “truncate” all or part of the simulation data suite as an interpretable record. In this paper we demonstrate the favorable characteristics of a Maximum Volume Skeletal Decomposition (MVSD) algorithm for CUR when compared against the SVD. Section 2 follows with a description of the procedure for an alternating maximum volume algorithm for matrix skeleton decomposition and demonstrates its computational scaling against the SVD. In Sec. 3.1 we present SOLPS-ITER simulations of the tokamak plasma boundary carried

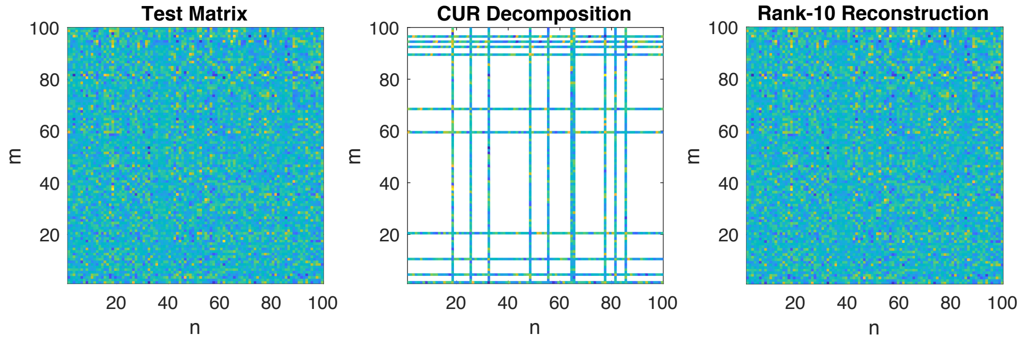

Figure 1: Example decomposition of a randomized 100×100 test matrix into its Columns-sUbmatrix-Rows (CUR) components. Left panel shows an *ad hoc* randomized matrix with embedded low-rank ($r = 10$) features. Center panel shows the corresponding $r = 10$ CUR components identified by the Maximum Volume Skeletal Decomposition (MVSD) algorithm, with the submatrix defined by the entries at the intersections of the columns and rows. Right panel shows the low-rank approximation of the original data using this matrix factorization scheme. Each panel is presented on the same color scale range.

out for this work. The subsequent Secs. 3.2 – 3.4 discuss the advantages of our approach on the full computational domain of SOLPS-ITER as well as on profiles and scalars of the system state. We conclude with a summary on the limitations of this data compression approach and suggest opportunities available for future work including, integration with computational and experimental workflows for the efficient scientific and operational analysis of fusion plasma devices.

2. Low-Rank Matrix Approximation via a Maximum Volume Algorithm

The aim of low-rank matrix approximation is to reduce a collection of data arranged in columns and rows by the product of components with lower dimensionality. This problem can be described in terms of a minimization between the fit of the approximation and the number of linearly independent column vectors, or rank, extracted from the data. A reduction in rank serves to promote both the modeling of a matrix and the compression of its data assuming that low-rank information is contained to sufficient degree. The focus of this paper is on the latter category of approaches and is discussed in this part by way of example. Figure 1 presents in the right panel a randomized test matrix of size 100×100 with a rank-10 feature explicitly embedded in its construction, determined by a break in the spectrum of the singular value decomposition (SVD). The center panel shows the application of the columns-submatrix-rows (CUR) decomposition carried out by the algorithm used in this work. In effect, each of the selected columns and rows in the CUR decomposition is the result of an optimization that we will show can be carried out faster than the standard SVD for high dimensional data. These components highlight the only entries of the original data set necessary to reproduce the matrix, and in a descriptive sense form the supporting "skeleton" of its structure. In this case the reconstruction to good fidelity is shown in the right panel of Fig. 1.

In comparison to the SVD, the CUR decomposition trades a guaranteed minimum least-squares error and rank ordered variance for an efficient and interpretable low-rank matrix approximation (7), (8). These procedures are well-defined on experimental or simulation timeseries of dynamics when the data is arranged into a matrix of spatial by temporal measurements. With respect to the CUR, the extracted columns represent full time step states and the rows represent independent coordinates or variables. The submatrix of this decomposition is defined by the block of entries at the intersection of these columns and rows. Given an adequate submatrix U_{rr} of rank-r, data from the original timeseries matrix is preserved along the subset of r columns C_{ir} and rows R_{rj} identifying the coordinates and time steps most significant to the complete reconstruction. In that sense, CUR decomposition determines the underlying spatio-temporal structure that supports the rest of the data in a matrix and is appropriately also known as the pseudoskeleton approximation (21).

We now define these terms in detail for our application of the CUR approach with a maximum volume skeletal decomposition (MVSD) algorithm. Suppose we select from a matrix $\mathbf{M}$ an invertible $r \times r$ submatrix, composed of entries at the intersection of r columns and rows. Without loss of generality permute the indices to obtain the block structure

$$\mathbf{M} = \begin{bmatrix} U & X \\ Y & Z \end{bmatrix}. \tag{1}$$

such that an approximation of the information contained in Z can be recovered from only a limited set of entries by

$$\mathbf{M}_r = \begin{bmatrix} U \\ Y \end{bmatrix} U^{-1} \begin{bmatrix} U & X \end{bmatrix} = \begin{bmatrix} U & X \\ Y & YU^{-1}X \end{bmatrix} = CU^{-1}R. \tag{2}$$

At a given rank specified as the retained number of columns and rows in the CUR factorization or singular values in the SVD, the CUR requires $r(1+r)$ fewer entries than the SVD to be stored. For low-rank matrices the differences in memory requirements are negligible. We define a relative compression ratio, R, between the original $m \times n$ matrix size and the number of entries extracted by a rank-r decomposition with either the SVD or MVSD as:

$$R_{SVD} = \frac{r(m+n+1)}{mn}, \quad R_{MVSD} = \frac{r(m+n-r)}{mn} \tag{3}$$

Note that in the unintended case of a full rank "decomposition," the SVD relative compression can exceed $R_{SVD} > 1$ whereas the MVSD yields $R_{MVSD} = 1$.

In order to ensure a well-conditioned submatrix U with invertability, we devise a measure of the volume as the modulus of the determinant

$$\mathrm{vol}(U) = |\mathrm{det(U)}| \tag{4}$$

following (22). Instead of an expensive deterministic search for the global submatrix that satisfies $\mathrm{vol}(U_{\max}) > \mathrm{vol}(U_{r\times r}) > 0$, we pursue an iterative algorithm that avoids any submatrix being too close to singular by increasing the volume towards a local, or dominant, maximum. Reference (23) derives an inequality for the bounded error of a CUR approximation taken with respect to a globally maximum volume submatrix

$$||\mathbf{M} - \mathbf{M_r}||_\infty \leq (r+1)\sigma_{r+1}(\mathbf{M}) \tag{5}$$

where $\sigma_{r+1}(\mathbf{M})$ is the $r+1$ singular value of the SVD of $\mathbf{M}$. As it would be exceedingly costly to find $U_{\max}$, we proceed with dominant submatrices satisfying $\mathrm{vol}(U_{\max}) \leq r^{r/2}\mathrm{vol}(U_{r\times r})$. The standard scheme for finding a locally dominant submatrix in particular can be expressed in pseudocode as shown in Algorithm 1.

This algorithm utilizes the products $\mathbf{M}U_I^{-1}$ and $U_I \backslash \mathbf{M}$ to define a dominant submatrix for rectangular matrices $m \times n$ instead of the square matrix determinant, requiring that the absolute value of all entries in each are no

Algorithm 1 Alternating Column/Row Maximum Volume Submatrix

```
Require: $\det(U_0) \neq 0$
Ensure: $k < 1000$
  k = 1
  $c_{ij} = 1 + 2\tau$
  $r_{ij} = 1 + 2\tau$
  while $(c_{ij}|r_{ij}) > 1 + \tau$ do
      if $k$ is even then
          $V_k = U_k \backslash \mathbf{M}_{i:}$
          $c_{ij} = \max(|V_k|, \mathrm{all})$
          $U_k = U_{:i} \leftarrow j^{th}$ column of $\mathbf{M}$
      else if $k$ is odd then
          $V_k = \mathbf{M}_{:j} U_k^{-1}$
          $r_{ij} = \max(|V_k|, \mathrm{all})$
          $U_k = U_{j:} \leftarrow i^{th}$ row of $\mathbf{M}$
      end if
      $k = k + 1$
  end while
```

larger than a tolerance of $1+\tau$. Alternating rows and columns are exchanged between the partitioned submatrix U_{ij} and the data matrix $\mathbf{M}$ until the information content changes by at most a factor of $1+\tau$. We note that the ratio between the volume of any sequential interchange of row or column in two invertible submatrices is preserved, such that $\mathrm{vol}(U_1) = |v_{ij}|\mathrm{vol}(U_0)$. This implies that the series $\mathrm{vol}(U_k)$ increases until we obtain a dominant submatrix. For this study we allow $\tau = 10^{-10}$ with $\max(k) = 1000$ and initialize U_0 as a $r \times r$ submatrix consisting of column and row entries taken from the r randomly selected indices of the $m \times n$ matrix $\mathbf{M}$. Reference (24) advances this methodology through a *greedy* algorithm for image processing where up to r rows and columns are swapped at a time on each iteration, enabling faster determination of the maximum volume submatrix. We showcase here the efficacy of the original algorithm and leave further developments to future work.

2.1. Empirical Scaling with Data Size

As an illustration of the advantages of this approach, we first demonstrate asymptotic behavior for the extraction of low-rank data embedded in high matrix degrees-of-freedom. Let r be the intended rank of the feature set and consider a $n \times n$ square matrix $\mathbf{M}$. We construct a randomized low-rank matrix of real entries $C = AB^T$, where A and B are independent normally distributed $n \times r$ matrices. To artificially create a rapid fall off of singular values distinct break in magnitude at rank r characteristic of low-rank information, we embed the matrix C in a randomized $n \times n$ Matlab test matrix with preassigned logarithmic decay in singular values, σ. Figure 2 shows the characteristic spectrum of the singular value decomposition of $\mathbf{M}$

for $n = 100$ (the matrix corresponding to these values can be seen in the left panel of Fig. 2). There are $r = 10$ significant singular values with energies, σ^2, at least 4 orders of magnitude higher than the tail of the distribution. This *ad hoc* formulation allows for the evaluation of separate matrix decomposition schemes for compressing the test data down to the low-rank M_r information it contains. We use the total compute time on a 2.4 GHz Intel Core i7 processor from algorithm initialization to reconstruction of $\mathbf{M_r}$ under the MATLAB svd(M, r), svds(M, r), and our implementation of maxvol(M, r). The relative error, ϵ, achieved by each respective decomposition scheme for a rank-r approximation is defined in terms of the spectral norm (natural matrix norm induced by the L2-norm) with respect to the normalization

$$\epsilon = \frac{\|M_r - M\|_2}{\|M\|_2} = \frac{\sigma_{max}\left(M_r - M\right)}{\sigma_{max}\left(M\right)} \tag{6}$$

and is used as a measure of algorithm performance.

The singular value decomposition (SVD) command utilizes LAPACK to compute all n (or $\min\left(m, n\right)$) singular values of a matrix and therefore has a compute time independent of rank selected after the distribution of singular values is obtained. As an alternative approach to this baseline we also consider the subset of singular value decomposition (SVDS) routine, which returns the r largest singular values and associated components of the matrix $\mathbf{M}$ using Lanczos Bidiagonalization (30)(31) and is especially effective at handling large sparse matrices faster than the SVD. Finally, the maximum volume pseudoskeleton decomposition (MVSD) follows Alg. 1 in MATLAB and approximates the original data up to rank r in the form of Eq. 2. The right panel of Fig. 2 shows the performance of repeated application of these methods at $r = 10$ for embedding matrix sizes of $n : 100, \cdots, 3000$. A mono-

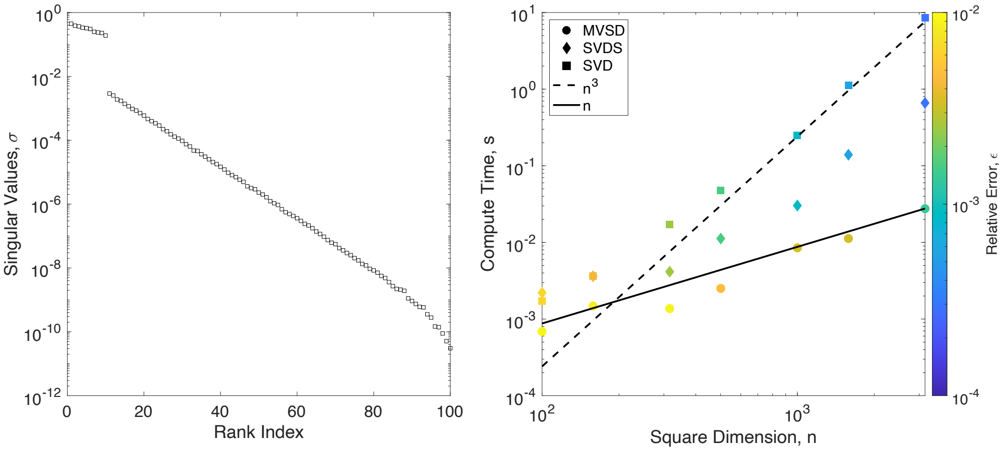

Figure 2: Projected scaling of the MVSD, SVDS, and SVD matrix factorization methods with system size for embedded low-rank features. Left panel shows the singular value energy spectrum for a 100×100 random square matrix constructed with a rank 10 substructure. Right hand panel shows the compute time in seconds for a rank $r = 10$ reconstruction using the aforementioned methods versus the number of entries, n, along one dimension of the generated square test matrices. Each point is represented in color against the total relative error in terms of the L2-norm of the decomposition. Unweighted monomial fits of the asymptotic behavior of the MVSD and SVD are shown by the dashed, n, and solid, n^3, lines.

mial fit, linear and cubic in n, is used as a reference for extrapolating the prospective compute time, in seconds, out to larger matrices. From this test we find that the accuracy of extracting fixed low-rank information improves in all three methods with matrix size, but presents less than an order of magnitude L2-norm relative error difference between them. In general, the SVD approaches n^3 scaling while the SVDS nears a n^2 asymptotic limit. Though the MVSD has higher relative error to within a small (< 5) factor, the CUR algorithm implemented in this work maintains very close to n scaling across the entire range of sampled matrices. These results are at least dependent on the presence of a distinct singular value spectrum, but for the purposes of data compression presuming the possibility of a reduced rank approximation we take them as sufficient for highlighting the expected behavior.

3. Data Compression of Fusion Plasma Physics Coupled Simulations

3.1. Modeling the Tokamak Plasma Boundary with SOLPS-ITER

SOLPS-ITER is the state-of-the-art release of the coupled 2D multi-fluid plasma solver (B2.5) and 3D kinetic neutrals Monte-Carlo code (EIRENE) (27) (28). The simulation suite is widely used, spanning several decades, for ITER divertor design and studying plasma physics at the tokamak boundary spanning from the outer plasma core, across the scrape-off-layer to the vacuum vessel and divertor. Transport dynamics are determined from evolution of coupled fluid continuity, energy, and momentum equations, coupled to parallelized computation of neutral trajectories with the associated plasma-neutral and plasma-surface interactions. Each fluid equation is advanced implicitly, with the coupled nonlinear system solved through Picard iteration. The plasma equations are solved on a field-aligned grid to handle the large anisotropy, with classical (Braginskii-like) transport in the parallel direction and *ad hoc* form of Fick's Law for convective and cross-field diffusion. Computational resources are available to EIRENE in terms of the total time per MC iteration and the number of CPU cores. Fundamentally, SOLPS-ITER is limited to restrictive time step size (typically 10^{-5} to 10^{-7} s) due to the strong nonlinearities, particularly in the plasma-neutral and radiative rate coefficients. At the same time, the timescales for particle balance (puff source and pumping terms are a small fraction of the recycled flux) can be long (seconds) and the simulations can take on the order of days-weeks-months to converge towards the steady-state for experimental and reactor class devices, such as DIII-D and ITER, respectively (32).

The primary time-dependent output of the software suite is the b2time.nc netcdf file, which comprises the time series of numerous state variables and post-processed quantities of interest. For any given run this can account for upwards of 50 million floating point numbers of varying degrees of orders of magnitude, which for extensive simulations can occupy a gigabyte of memory when multiple ion species are present and 10s of thousands of time steps are recorded. Additionally, SOLPS-ITER is often utilized to perform many runs over several input parameter scans that can increase these storage requirements by a multiplicative factor. Since the kinetic neutral dynamics are taken to be on a timescale much faster than the fluid plasma transport, time-dependent information from EIRENE is not typically preserved and time-independent information is kept in a separate file format due to the code coupling requirements with B2.5. Recent validation efforts with SOLPS-ITER have tested the significance of different contributions to the error in predictions of the final converged steady state. It was found that using a fast fluid neutrals model in place of expensive EIRENE calculations can introduce up to 50% error in edge electron densities, while a hybrid-kinetic approach offers a speed up factor of 6 with maximum error constrained to 20% (33). A scoping study separating out the B2.5 finite volume and EIRENE Monte Carlo sources of numerical error found that the choice of discretization in the plasma grid contributes up to 18% discrepancies in the density and temperature profiles. Though statistical errors introduced by the Monte Carlo coupling can be brought down to a level less than 0.5% through an iteration averaging procedure the finite particle sampling still limits the maximum bias error to be on the order of 1% (34). A key challenge evident in the implementation of SOLPS-ITER simulations is the balance of reasonable numerical accuracy with computational cost. For the purposes of data compression of the simulation output presented here, we target a numerical fidelity on the order of 1%, or an absolute relative error of 10^{-2}.

A typical SOLPS-ITER simulation up to steady-state conditions for an experimental configuration, such as DIII-D shot equilibria, can produce as much as 140 output dynamic variables on the order of 1000 - 10000 time steps. The scope of these physically relevant quantities encompasses up to 3–6 2D matrices per plasma species (n_s, T_s, p_s) of size 98×38, around 60 1D profiles (including line integrated measurements and fluxes) of size 38×1, and 80 0D scalar timeseries (including total power and currents) of size 1×1. In total,

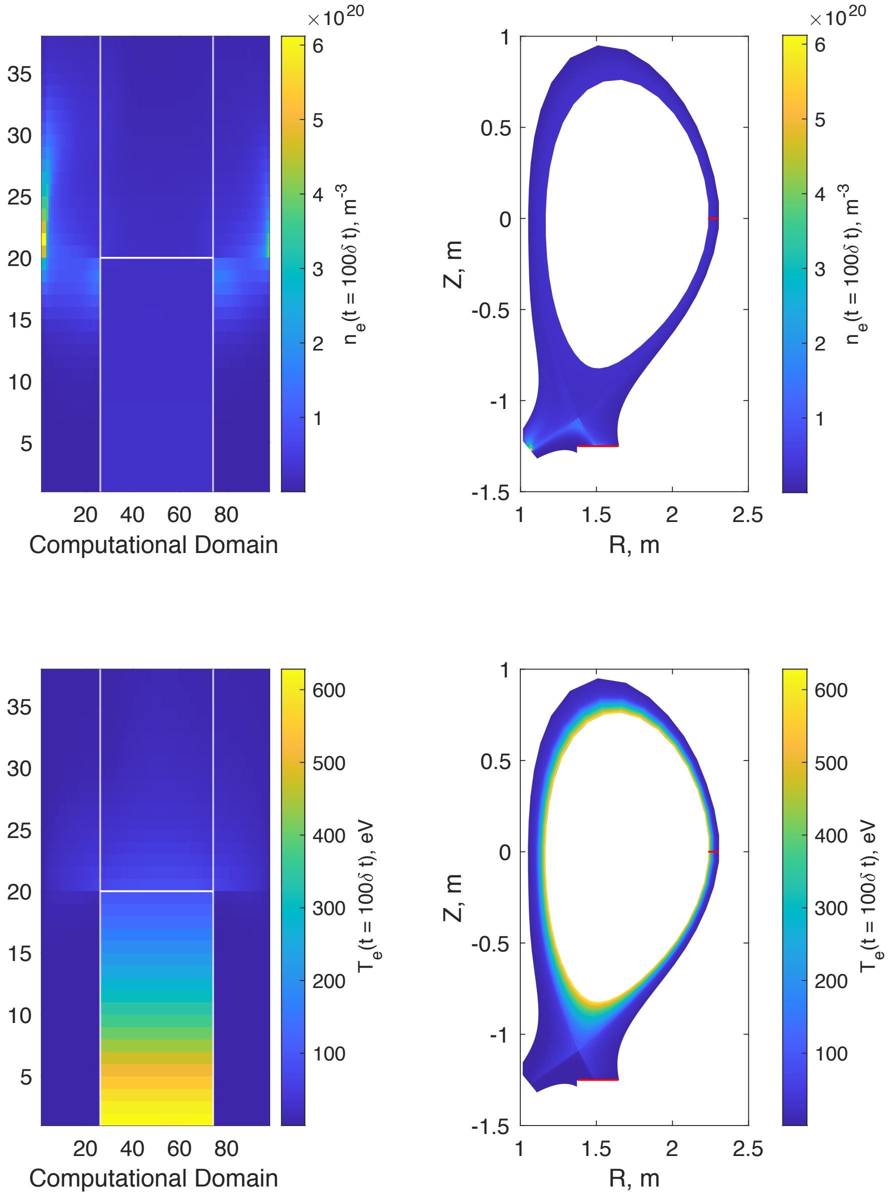

Figure 3: Snapshots of the full finite volume SOLPS simulation at $100\delta t$. Left column shows the computational domain matrix of the calculation and right column shows the magnetic equilibrium configuration for DIII-D shot 174310 at 3500 ms corresponding to the same variables and time steps. Top panels show the 2D electron density and bottom panels show the 2D electron temperature.

nearly 25000 entries are calculated and advanced in time with respect to the plasma transport fluid variables. Though SOLPS-ITER simulation is largely encumbered by the Monte Carlo treatment of the kinetic neutral trajectories, the resulting interactions are preserved only in the plasma response unless EIRENE data is specifically recorded in a separate output file. It is assumed that the kinetic neutral dynamics are much faster than the fluid plasma timescales and reach convergence at each integration time step such that it can be feasible to restart SOLPS-ITER intermittently over a simulation run, provided the appropriate plasma state information is retained. SOLPS-ITER is often utilized in a time-independent manner to predict steady-state conditions for operational scenarios and experimental design, implying that the simulation data can be compressed through extraction of the latent solutions as low-rank features within matrix data.

3.2. 2D Field Data

Figure 3 shows two snapshots of the plasma state at 100 time steps, $\delta t = 10^{-5}$ s, into a SOLPS simulation using a magnetic equilibrium corresponding to DIII-D shot 174310 at 3500 ms with only deuterium plasma species for the fluid ions. Both the 2D electron density (SOLPS label, *ne3da*) and electron temperature (*te3da*), as fundamental variables governing the dynamics of the scrape-off-layer, are shown in the top and bottom panels respectively. The rightmost panels illustrate the finite-volume geometry of the experimental configuration. In the leftmost panels of Fig. 3, the computational domain associated with this mapping is presented. The x-axis corresponds to the parallel to magnetic field direction and the y-axis corresponds to the perpendicular to magnetic field direction. Subdivisions with respect to the the following plasma regions can be identified: along the y-axis the inboard divertor volume is recorded from 1 to 26 on the x-axis, the core and scrape-off layer share the domain between $x = 26$–74 and are split across the separatrix between $y = 19$ & 20, lastly the outboard divertor occupies $x = 76$–98. In terms of synthetic diagnostic quantities typically utilized in the analysis of SOLPS simulations, we also consider 1D profiles of the following target quantities. The outboard divertor target (ODT) electron density (*nesepa*) is obtained at $Z = -1.25$ m from $R = 1.378$ m – 1.645 m and electron temperature (*tesepa*) over the same range as illustrated by the red line highlight. These profiles are asymmetric across the separatrix in the nonuniform sampling across R, and are located in the computational domain at $(x = 98, y = 1$–$38)$. For this baseline case a steady-state solution was perturbed by doubling the input power and running to approximate convergence. As expected, on the outboard divertor target the electron density falls to a level of 1.5×10^{19} m^{-3} just inside the separatrix while the electron temperature gradually rises past 9 eV over 1600 time steps, $\delta t = 1 \times 10^{-5} s$.

We apply matrix decomposition schemes for data compression to the output of SOLPS simulations directly in the discretized coordinate-time step domain without accounting for the measured spatio-temporal sampling. To enable the required matrix operations over the tensor output, Figure 4 takes the preceding simulation in the 2D computational domain and reshapes the spatial dimensions into a single column major ordered list of coordinates versus time steps. This procedure follows the spatio-temporal separation of matrix "unfolding," or stacking, that allows a discretized scalar field tensor to be represented by a tensor product of *Topos* and *Chronos* elements com-

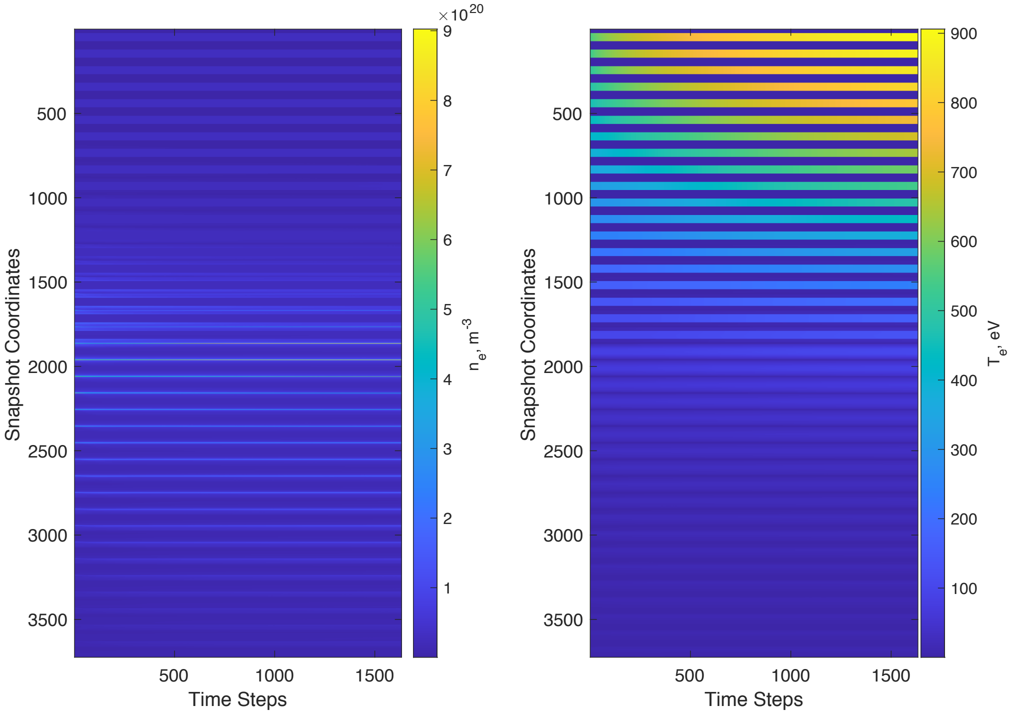

Figure 4: SOLPS 2D simulations of the electron density on the left and electron temperature on the right reshaped into an expanded coordinates-time steps matrix. Each column vector represents a snapshot of the full computational domain ($98 \times 38 = 3724$).

monly utilized in principal component analysis (PCA) (25), (26). The left hand panel shows the evolution of electron density and the right hand panel shows the corresponding dynamics of the electron temperature for the same plasma regions. This transformation augments the system size of the simulation data matrix from 38 × 98 × 1600 to 3724 × 1600 requiring additional processing to achieve a successful decomposition.

In Figures 5 & 6 the performance of the three data compression methods is compared, highlighting the prospective gains due to the MVSD algorithm. For the electron density simulation, which features highly localized peaks in values at the divertor target separatrix, the low-rank reconstruction at $r = 4$ for the MVSD results in a total relative error of 5×10^{-2} with a compute time of 10^{-2} s, greater than 2 orders of magnitude faster than the SVD and nearly 1 order of magnitude faster than the SVDS. The MVSD plateaus at an accuracy of 10^{-3} around a rank reconstruction of $r = 25$, and intersects the SVDS routine at $r = 12$ for 10^{-1} s compute time.

Both the SVDS and SVD reach 5×10^{-5} relative error at high rank reconstructions of the original data matrix, here truncated to $r = 98$. Due to the large size the overall difference in relative compression with respect to the ratio of number of terms retained is negligible. On the right hand panel of Fig. 5 the MVSD and SVD algorithm provide similar accuracy at low rank for compression down to a relative ratio of 0.01. The methods are further separated by one order of magnitude in relative error at the MVSD threshold level of 10^{-3} for a relative compression of 0.04. Therefore, for an acceptable relative error tolerance of 10^{-2} the MVSD algorithm achieves one (two) order(s) of magnitude faster data compression down to 0.01 than the

SVDS (SVD) method.

The simulation of electron temperature presents the same results, albeit with higher accuracy in all three methods due to the global gradients in values shown by the right hand panel of Fig. 4. In Figure 6 the MVSD relative error begins at 5×10^{-3} for a $r = 4$ low-rank reconstruction and falls to a threshold level of 10^{-5} around a high-rank reconstruction of $r = 40$, intersecting the SVDS at an $r = 14$ and compute time of 2×10^{-1} s. Again, the MVSD algorithm is two orders of magnitude faster than the standard SVD and demonstrates high data compression to a ratio of 0.01 at a relative error of 10^{-4} for $r = 10$.

In practice, the *a priori* selection of appropriate rank is challenging. When the SVD is computed, the full range of singular values give a measure of the least-squares error captured by each component. A requested fidelity could be maintained across the range of simulation outputs by setting the rank separately for each dataset according to the singular value truncation *a posteriori*. However, computation of the SVD is prohibitive in matrices with high degrees-of-freedom as identified in the scaling argument of Fig. 2. The SVDS, which requires a rank assignment for reasonable compute times, would rather need to be pursued. In Figs. 5 & 6 the SVDS method demonstrates an asymptotic limit in accuracy gain that approaches the compute time of the SVD for increasing rank. Though the MVSD algorithm instead shows a relative error threshold about 1 order of magnitude higher than this range, overall it performs the fastest of the three methods for low-rank reconstructions. This suggests that it might be adequate to choose a desired relative compression, at 0.01 or so for these 2D datasets, and base the rank

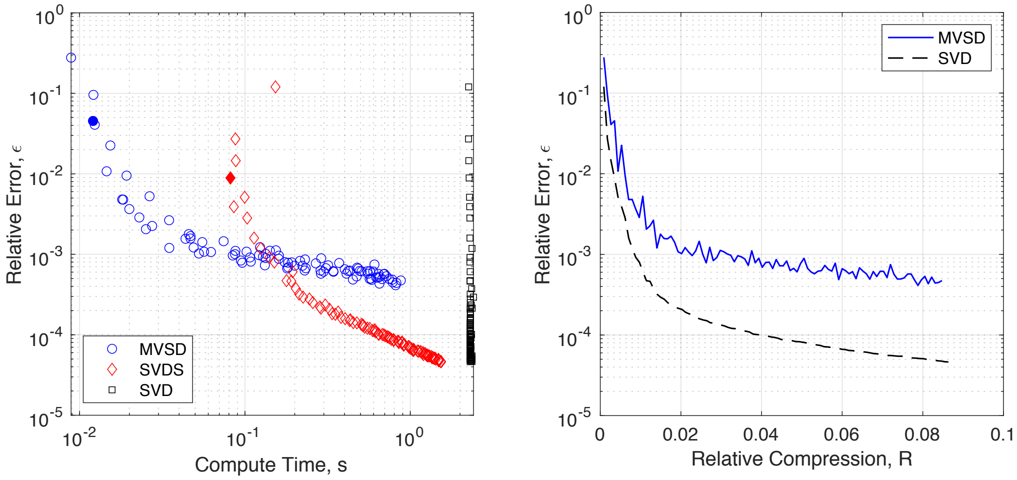

Figure 5: Performance of the MVSD, SVDS, and SVD on the SOLPS 2D simulation of electron density reshaped as a matrix (3724 × 1600). Left hand panel shows the total relative error for each rank reconstruction up to a partial rank of $r = 98$ versus the compute time. The low-rank $r = 4$ reconstructions are indicated by the solid blue and red points for the MVSD and SVDS methods, respectively. The right hand panel shows the relative error versus relative compression as the rank of reconstruction is increased for the MVSD and SVD methods.

selection off the associated number of components in the CUR decomposition. In this case, the relative error between matrix approximation schemes is more reasonable and closer to a factor of 2 difference.

3.3. 1D Profile Evolution

Figure 7 takes the 1D profiles of electron density and temperature on the outboard divertor target more commonly recorded than the full 2D simulation and shows the respective dynamics in the left and right hand panels. For the case considered here, with the modification to the input power, the density at the separatrix drops gradually over 1000 time steps whereas the

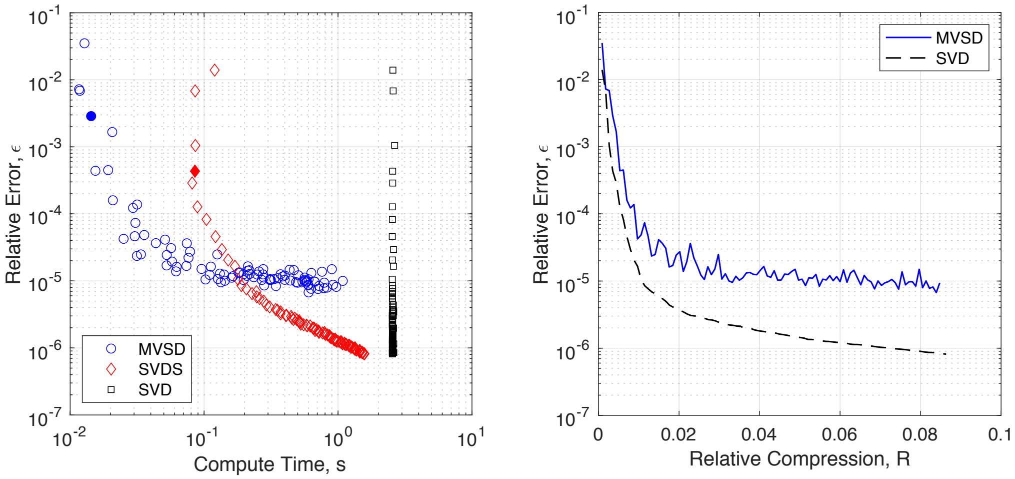

Figure 6: Performance of the MVSD, SVDS, and SVD on the reshaped matrix (3724×1600) of SOLPS 2D simulation of electron temperature. Left hand panel shows the total relative error for each rank reconstruction up to a partial rank of $r = 98$ versus the compute time. The low-rank $r = 4$ reconstructions are indicated by the solid blue and red points for the MVSD and SVDS methods, respectively. The right hand panel shows the relative error versus compression factor as the rank of reconstruction is increased for the MVSD and SVD methods.

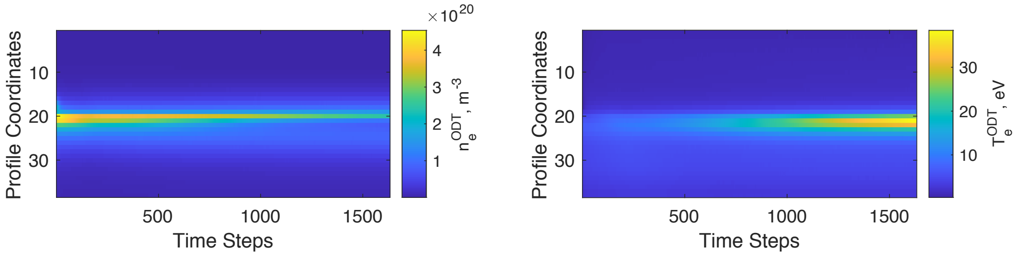

Figure 7: SOLPS 1D simulations of the outboard divertor target (ODT) electron density on the left and electron temperature on the right taken from the full computational domain. Each column vector represents a profile of coordinates at a single time step on the outboard divertor target.

temperature around the separatrix increases quickly in the last 500 time steps of the simulation. In contrast to the preceding reshaped calculations, all 38 coordinates are stored in adjacent ordering as output from the simulation to preserve the spatial correlation of the solution.

Figures 8 & 9 highlight the performance of the MVSD on datasets of rectangular matrix size. In the left hand panel of Fig. 8 for the electron density simulation, the MVSD maintains at least a factor of 2 faster computation than the SVDS method up to a low rank reconstruction of $r = 4$. The two algorithms reproduce the same total relative error of 10^{-3}–10^{-4} and similar compute times in the range of $r = 5$–10. At higher ranks, the SVDS is up to one order of magnitude more efficient than the MVSD. This abrupt change in compute time can be attributed to the implementation of the MATLAB algorithm for the SVDS, which in the documentation is described as capable of improved performance at increased rank when repeated singular values are present (as can be the case after a break in singular spectrum) (30)(31). In all cases the SVD performs slowest at a compute time of 10^{-1} s. The right hand panel of Fig. 8 further shows that the difference in relative error between the decomposition schemes is negligible and that the same data compression relative ratio can be achieved by the MVSD as the SVD.

Fig. 9 presents confirmation of the results for the simulation of electron temperature. In that case there is minimal gain in accuracy when considering higher rank reconstructions past $r = 10$. It is evident that a low-rank MVSD of at least $r = 4$ is sufficient to represent the SOLPS simulation data to within a total relative error of 10^{-2} at the fastest speed. We remark that at full rank, the discrepancy between the highest fidelity reconstructions is due to the default limited tolerance on the Matlab iterative implementation of the SVD and SVDS schemes. In this fringe case, the inverse operation utilized by the MVSD algorithm for the full matrix and successive matrix multiplications actually achieves the lowest error.

3.4. 0D Scalar Timeseries

Figure 10 shows the concatenated timeseries of 77 0D scalar quantities obtained from the SOLPS-ITER simulation and typically used as metrics for plasma state characterization. These heterogenous variables span several orders of magnitude and disparate physical units. To allow satisfactory decomposition of the augmented matrix data, each scalar is normalized by the absolute value of the mean from the respective timeseries. This procedure attempts to allow meaningful comparison of the total relative error when the data is reconstructed. Fig. 10 also indicates the variable names selected by the rank $r = 4$ MVSD along with the identified full-fidelity snapshot states at time steps marked by the vertical black lines at $N\delta t = 1$, 49, 847, and 2994.

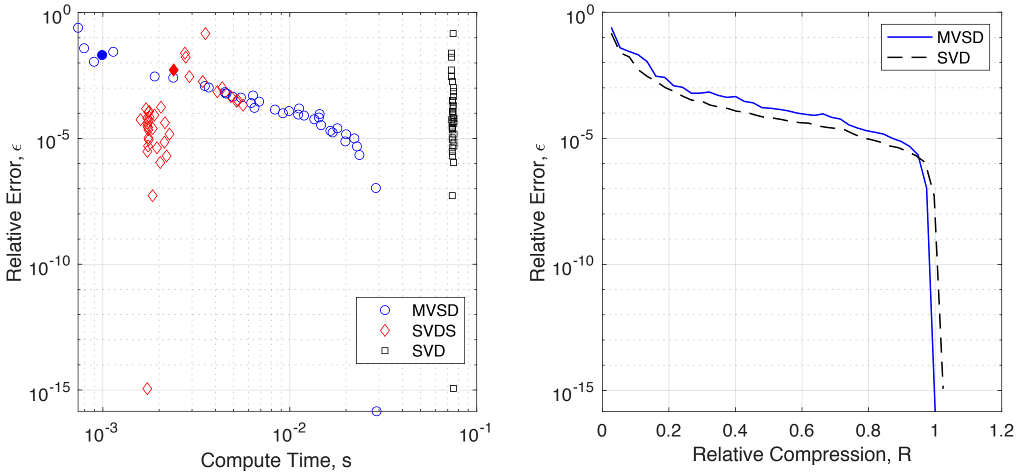

Figure 8: Performance of the MVSD, SVDS, and SVD on the SOLPS 1D simulation of outboard divertor target electron density (38 × 1600). Left hand panel shows the total relative error for each rank reconstruction up to full rank $r = 38$ versus the compute time. The low-rank $r = 4$ reconstructions are indicated by the solid blue and red points for the MVSD and SVDS methods, respectively. The right hand panel shows the relative error versus relative compression as the rank of reconstruction is increased for the MVSD and SVD methods.

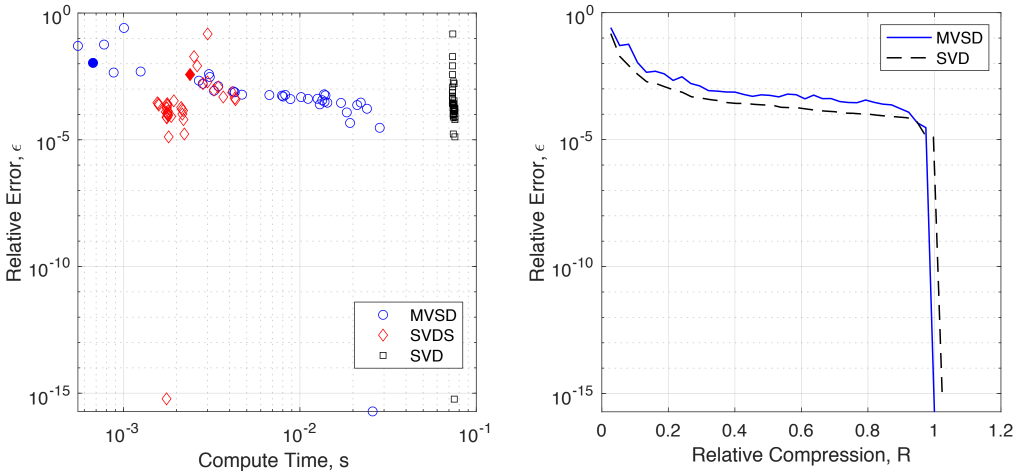

Figure 9: Performance of the MVSD, SVDS, and SVD on the SOLPS 1D simulation of outboard divertor target electron temperature (38 × 1600). Left hand panel shows the total relative error for each rank reconstruction up to full rank $r = 38$ versus the compute time. The low-rank $r = 4$ reconstructions are indicated by the solid blue and red points for the MVSD and SVDS methods, respectively. The right hand panel shows the relative error versus relative compression as the rank of reconstruction is increased for the MVSD and SVD methods.

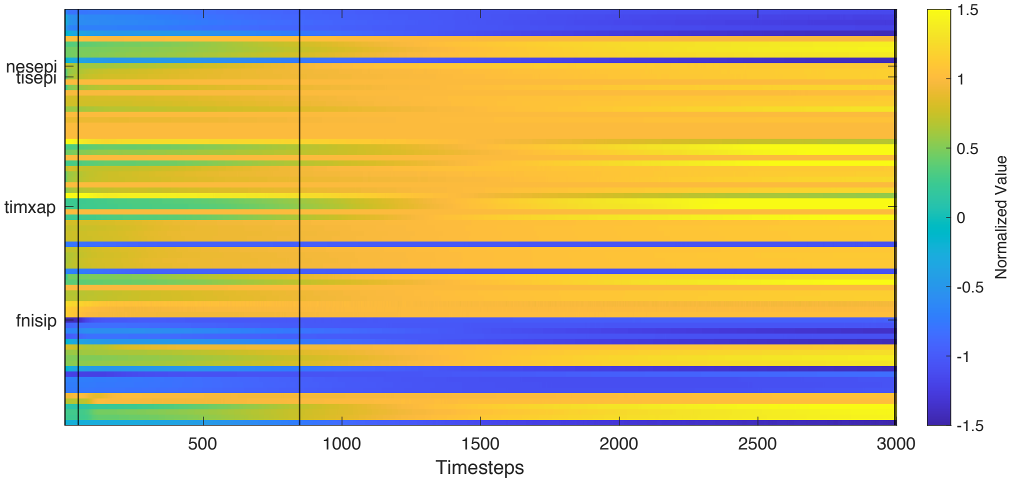

Figure 10: SOLPS 0D simulations of timeseries integrated or calculated from the full solution. Each column vector represents a snapshot of the full plasma state with respect to the scalar quantities at a single time step. The variables extracted by the MVSD low-rank $r = 4$ reconstruction are labeled along the y-axis and the corresponding time step components are denoted by vertical black lines.

In this case, the dynamics are reproduced from the Western (inboard divertor) edge separatrix electron density, *nesepi* (m^{-3}), and ion temperature, *tisepi* (eV), as well as the Western (inboard divertor) separatrix throat poloidal particle flux, *fnisip* (s^{-1}), and Eastern (outboard divertor) edge maximum ion temperature, *timxamp* (eV). The MVSD is predominantly distributed along the Western separatrix and coupled to the Eastern edge by the ion temperature over the first third of the simulation. Though it is clear that many of these scalar quantities exhibit similar temporal evolution and that the MVSD is not guaranteed to find the global submatrix of maximal information content, this example displays the capability of the CUR approach to extract prioritized spatio-temporal measurements from integrated simulation data that could be used to indicate governing dynamics and restart intervals for SOLPS-ITER.

Figure 11 clarifies these remarks in the comparison of performance by the MVSD, SVDS, and SVD on the SOLPS scalar timeseries. The left panel of shows that MVSD still achieves the fastest compute time, two orders of magnitude less than the SVD, at a level of 10^{-2} total relative error for the rank $r = 4$ reconstruction. The SVDS method intersects the MVSD at $r = 7$ and exceeds the efficiency of the algorithm at $r = 14$ until reaching the lowest reconstruction error of 10^{-4} for $r = 24$. On the right hand panel of Fig. 11 the data compression achieved by the MVSD is separated from the SVD by a factor of 3 total relative error. For relative compression ratios below 0.1 there is about a $R = 0.025$ positive offset i.e., requiring higher rank, for the MVSD to reproduce equivalent reconstructions. Below a total relative error of 2×10^{-3} this offset grows to $R = 0.05$, eventually nearing a threshold

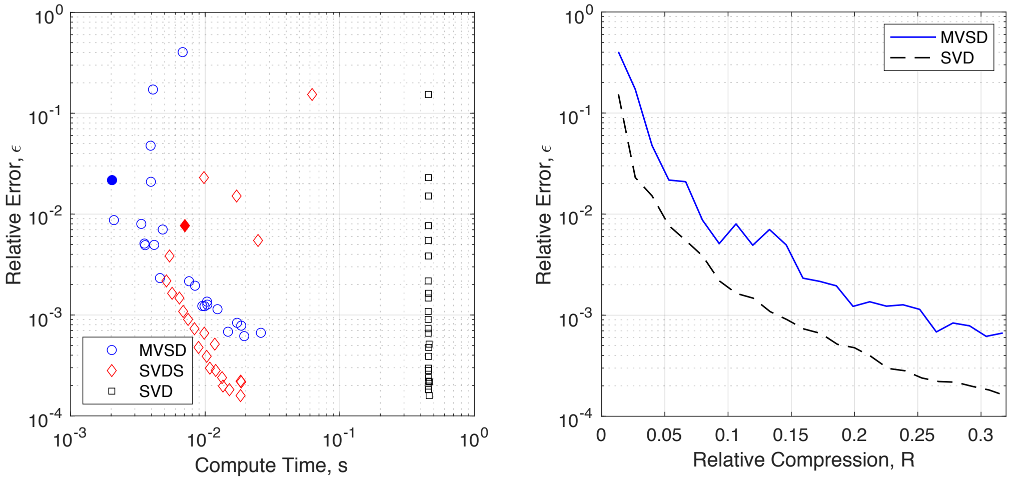

Figure 11: Performance of the MVSD, SVDS, and SVD on the SOLPS 0D scalar timeseries calculations (77×3000). Left hand panel shows the total relative error for each rank reconstruction up to a partial rank of $r = 24$ versus the compute time. The low-rank $r = 4$ reconstructions are indicated by the solid blue and red points for the MVSD and SVDS methods, respectively. The right hand panel shows the relative error versus relative compression as the rank of reconstruction is increased for the MVSD and SVD methods.

accuracy of 7×10^{-4}.

The advantage of the MVSD algorithm, as demonstrated in this work, is two-fold: (1) large matrices of simulation data can be efficiently compressed to acceptable error tolerances, and (2) the extracted decomposition is performed on exact spatial coordinates and temporal snapshots instead of a transformed basis. These benefits motivate inclusion of the MVSD in a SOLPS simulation workflow whereby more output variables could be reasonably stored than is done in practice, all linked to an interpretable record of which data corresponds to the full fidelity solution (the components of the CUR factorization). Of course, the MVSD need not be applied to all of the simulation output but could instead be utilized in a manner to enhance the availability of additional sources of information describing the plasma state at minimal extra cost.

4. Conclusions

An algorithm for the compression of matrices was applied to tokamak boundary simulation data from the SOLPS-ITER fluid plasma and kinetic neutrals interactions transport code. This paper applies a maximum volume approach to matrix pseudoskeleton decomposition (MVSD), which extracts a number of columns and rows corresponding to a submatrix of “locally” maximal information content at the specified rank. In contrast to the singular value decomposition (SVD), computation of the MVSD scales nearly linearly with square matrix size when low-rank features are present. The CUR approach as a data compression method is suggested to be tractable on high-dimensional plasma physics simulation data, at least where an acceptable error in the reconstruction can be tolerated. Its performance is characterized here on the SOLPS-ITER code, where it is common to curtail output variables to a reasonable number when repeated runs are carried out and storage is limited.

SOLPS-ITER provides state-of-the-art simulation of the scrape-off-layer, the boundary region of plasma magnetic confinement in fusion experiments. Typically, the code is utilized to obtain a solution of the steady-state conditions for determination of component viability facing the plasma, such as the divertor, or interpretation of observations from experimental shots. These tasks can require a degree of tuning various parameters and result in an array of simulation runs with disjoint characteristics. Increasing the quantity of output variables could be useful for allowing a comprehensive analysis of the dynamics between runs or negate the need to repeat a simulation in order to obtain previously uncalculated data. For the scientist user of SOLPS-ITER, traceability of the MVSD is straightforward. The algorithm preserves the exact spatial coordinate timeseries and full fidelity time step states used in the CUR data compression of specific dynamic variables.

SOLPS-ITER can produce upwards of 140 output variables over thousands of time steps, stored in matrices ranging in size from the 2D computational domain (38×98) to 1D profiles (38) and 0D scalars. In total, over 120 million floating point numbers ($\sim$ 500 MB – 1 GB) can be recorded per simulation run. The results of this work show that the MVSD is at least 2 orders of magnitude faster to compute than the SVD and maintains at least 10^{-2} total relative error with rank $r = 4$–6 for all cases investigated here. The low-rank reconstruction of SOLPS-ITER data ordered in spatial coordinates versus time step snapshots yields sufficient relative compression ratios that improve with the size of the data. The MVSD algorithm also performs better than or as well as the SVDS up to a total relative error of at least 10^{-3}. We note that there is substantial research being pursued to allow the extension of the SOLPS-ITER computational grid into the vacuum where neutrals reside and to the plasma facing component wall geometry for improved physical fidelity (35). The datasets from these simulations can be larger than those presented here due to the customized finite-volume grid discretizations.

The efficiency of the MVSD facilitates simulation data mobility, allowing the full SOLPS-ITER plasma state information to be stored and transferred at reduced costs. In addition, two lines of research for future investigation are evident. The first concerns the identification of limited spatial coordinates extracted by the MVSD with optimal placement of synthetic diagnostics in SOLPS-ITER for monitoring plasma dynamics, which could be corroborated by experimental observations. The second concerns the complete time step snapshots in the MVSD, which could be chosen as intervals for recording dependencies required to restart SOLPS-ITER simulation. The utility of the MVSD algorithm for CUR matrix factorization demonstrated here on fluid plasma dynamics in the tokamak boundary motivates its extension to other high-dimensional simulations. Gyrokinetic plasma physics with much higher degrees of freedom, exhibiting data up to 5D in space and 1D in time, could be targeted as another candidate for efficient data compression where disparate scales from turbulent transport could be a challenge for this approach.

Acknowledgement

This work is supported in part by the US DOE under contract DE-AC05-00OR22725. Dr. Kenneth Allen was supported in part by the DOE Office of Science Graduate Student Research (SCGSR) program, and under thesis advisement by Dr. Ming-Jun Lai. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

[1] N. Pomphrey, L. Berry, A. Boozer, R. Hatcher, S. Hirshman, L.-P. Ku, W. Miner, H. Mynick, W. Reiersen, D. Strickler, P. Valanju, Innovations in compact stellarator coil design, Nuclear Fusion 41 (3) (2001) 339.

[2] S. Benkadda, T. D. de Wit, A. Verga, A. Sen, A. Team, X. Garbet, Characterization of coherent structures in tokamak edge turbulence, Physical Review Letters 73 (1994) 3403–3406. doi:10.1103/PhysRevLett.73.3403.

[3] S. Futatani, S. Benkadda, D. del Castillo-Negrete, Spatiotemporal multiscaling analysis of impurity transport in plasma turbuluence us-

ing proper orthogonal decomposition, Physics of Plasmas 16 (042506) (2009). doi:10.1063/1.3095865.

[4] D. del Castillo-Negrete, D. Spong, S. Hirshman, Proper orthogonal decomposition methods for noise reduction in particle-based transport calculations, Physics of Plasmas 15 (092308) (2008). doi:10.1063/1.2979680.

[5] D. Hatch, P. Terry, F. Jenko, F. Merz, W. Nevins, Saturation of gyrokinetic turbulence through damped eigenmodes, Physical Review Letters 106 (115003) (2011). doi:10.1103/PhysRevLett.106.115003.

[6] D. Hatch, P. Terry, F. Jenko, F. Merz, M. Pueschel, W. Nevins, E. Wang, Role of subdominant stable modes in plasma microturbulence, Physics of Plasmas 18 (055706) (2011). doi:10.1063/1.3563536.

[7] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218. doi:10.1007/BF02288367.

[8] L. Mirsky, Symmetric gauge functions and unitarily invariant norms, Quarterly Journal of Mathematics 11 (1) (1960) 50–59. doi:10.1093/qmath/11.1.50.

[9] G. Golub, C. van Loan, Matrix Computations, third ed., The Johns Hopkins University Press, London, 1996.

[10] P. Holmds, J. Lumley, G. Berkooz, Turbulence, Coherent Structures, Dynamical Systems and Symmetry, Cambridge University Press, New York, 1996.

[11] D. del Castillo-Negrete, S. Hirshman, D. Spong, E. D'Azevedo, Compression of magnetohydrodynamic simulation data using singular value decomposition, Journal of Computational Physics 222 (2007) 265–286. doi:10.1016/j.jcp.2006.07.022.

[12] D. Hatch, D. del Castillo-Negrete, P. Terry, Analysis and compression of six-dimensional gyrokinetic datasets using higher order singular value decomposition, Journal of Computational Physics 231 (2012) 4234–4256. doi:10.1016/j.jcp.2012.02.007.

[13] Y. Asahi, K. Fujii, D. Heim, S. Maeyama, X. Garbet, V. Grandgirard, Y. Sarazin, G. Dif-Pradalier, Y. Idomura, M. Yagi, Compressing the time series of five dimensional distribution function data from gyrokinetic simulation using principal component analysis, Physics of Plasmas 28 (1) (2021) 012304. doi:10.1063/5.0023166.

[14] G. Stewart, Four algorithms for the efficient computation of truncated qr approximations to a sparse matrix, Numerical Mathematics 83 (1999) 313–323.

[15] H. Cheng, Z. Gimbutas, P. Martinsson, V. Rokhlin, On the compression of low rank matrices, SIAM Journal of Scientific Computing 26 (2005) 1389–1404.

[16] A. Frieze, R. Kannan, S. Vempala, Fast monte-carlo algorithms for finding low-rank approximations, Journal of the ACM 51 (6) (2004) 1025–1041. doi:10.1145/1039488.1039494.

[17] K. Hamm, L. Huang, Stability of sampling for cur decompositions, Foundations of Data Science 2 (2) (2020) 1–36. doi:10.3934/fods.2020006.

[18] P. Drineas, M. W. Mahoney, S. Muthukrishnan, Subspace sampling and relative-error matrix approximation: Column-based methods, in: J. Díaz, K. Jansen, J. Rolim, U. Zwick (Eds.), Approximation, Randomization, and Combinatorial Optimization, Vol. 4110 of Lecture Notes in Computational Science, Springer, 2006, pp. 321–326.

[19] P. Drineas, M. Mahoney, S. Muthukrishnan, Relative-error cur matrix decompositions, SIAM Journal on Matrix Analysis and Applications 30 (2) (2008) 844–881. doi:10.1137/07070471X.

[20] M. Mahoney, P. Drineas, Cur matrix decompositions for improved data analysis, Proceedings of the National Academy of Sciences (055706) (2011). doi:10.1073/pnas.0803205106.

[21] S. Goreinov, E. Tyrtyshnikov, N. Zamarashkin, A theory of pseudoskeleton approximations, Linear Algebra and Its Applications 261 (1997) 1–21.

[22] S. Goreinov, E. Tyrtyshnikov, The maximum-volume concept in approximation by low-rank matrices, Contemporary Mathematics 280 (2001) 47–51.

[23] S. A. Goreinov, I. V. Oseledets, D. V. Savostyanov, E. E. Tyrtyshnikov, N. L. Zamarashkin, How to find a good submatrix, in: Matrix Methods: Theory, Algorithms, Applications, World Scientific, Hackensack, NY, 2010, pp. 247–256. doi:?

[24] K. Allen, A geometric approach to low-rank matrix and tensor completion, Ph.D. thesis, University of Georgia, `https://esploro.libs.uga.edu/esploro/outputs/doctoral/A-Geometric-Approach-to-Low-Rank-Matrix/9949421027102959` (2022).

[25] N. Aubry, R. Guyonnet, R. Lima, Spatiotemporal analysis of complex signals: Theory and applications, Journal of Statistical Physics 64 (1991) 683–739.

[26] L. de Lathauwer, B. de Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM Journal on Matrix Analysis and Applications 21 (4) (2000) 1253–1278. `doi:10.1137/S0895479896305696`.

[27] S. Wiesen, D. Reiter, V. Kotov, M. Baelmans, W. Dekeyser, A. S. Kukushkin, S. W. Lisgo, R. A. Pitts, V. Rozhansky, G. Saibene, I. Veselova, S. Voskoboynikov, The new solps-iter code package, Journal of Nuclear Materials 463 (2015) 480–484. `doi:10.1016/j.jnucmat.2014.10.012`.

[28] X. Bonnin, W. Dekeyser, R. Pitts, D. Coster, S. Voskoboynikov, S. Wiesen, Presentation of the new solps-iter code package for tokamak plasma edge modeling, Plasma and Fusion Research 11 (1403102) (2016) 1403102(1)–(6). `doi:10.1585/pfr.11.1403102`.

[29] R. Pitts, X. Bonnin, F. Escourbiac, H. Frerichs, J. Gunn, T. Hirai, A. Kukushkin, E. Kaveeva, M. Miller, D. Moulton, V. Rozhansky, I. Senichenkov, E. Sytova, O. Schmitz, P. Stangeby, G. D. Tem-

merman, I. Veselova, S. Wiesen, Physics basis for the first iter tungsten divertor, Nuclear Materials and Energy 20 (2019) 100696(1)–(25). doi:10.1016/j.nme.2019.100696.

[30] R. M. Larsen, Lanczos bidiagonalization with partial reorthogonalization, Department of Computer Science, Aarhus University DAIMI PB-357 (1998). doi:10.7146/dbp.v27i537.7070.

[31] J. Baglama, L. Reichel, Augmented implicitly restarted lanczos bidiagonalization methods, SIAM Journal on Scientific Computing 27 (1) (2005) 19–42. doi:10.1137/04060593X.

[32] E. Kaveeva, V. Rozhansky, I. Senichenkov, I. Veselova, S. Voskoboynikov, E. Sytova, X. Bonnin, D. Coster, Speed-up of solps-iter code for tokamak edge modeling, Nuclear Fusion 58 (2018) 126018(1)–(15). doi:10.1088/1741-4326/aae162.

[33] N. Horsten, M. Groth, W. Dekeyser, W. V. Uytven, S. Aleiferis, S. Carli, J. Karhunen, K. D. Lawson, B. Lomanowski, A. G. Meigs, S. Menmuir, A. Shaw, V. Solokha, B. Thomas, J. Contributors, Validation of solps-iter simulations with kinetic, fluid, and hybrid neutral models for jet-ilw low-confinement mode plasmas, Nuclear Materials and Energy 33 (101247) (2022) 1–9. doi:10.1016/j.nme.2022.101247.

[34] D. Boeyaert, S. Carli, K. Ghoos, W. Dekeyser, S. Wiesen, M. Baelmans, Numerical error analysis of solps-iter simulations of east, Nuclear Fusion 63 (016005) (2023) 1–16. doi:10.1088/1741-4326/aca0ab.

[35] W. Dekeyser, P. Boerner, S. Voskoboynikov, V. Rozhansky, I. Senichenkov, L. Kaveeva, I. Veselova, E. Vekshina, X. Bonnin, R. Pitts, M. Baelmans, Plasma edge simulations including realistic wall geometry with solps-iter, Nuclear Materials and Energy 27 (2021) 100999(1)–(11). `doi:10.1016/j.nme.2021.100999`.