RANDOMIZED GAUSS-SEIDEL AND COLUMN-SLICE-ACTION METHODS FOR TENSOR PROBLEMS

JAMIE HADDOCK, PAULINA HOYOS, ALONA KRYSHCHENKO, KAMILA R. LARRIPA, SHAMBHAVI SURYANARAYANAN, KARAMATOU YACOUBOU-DJIMA

1. History of Collaboration

We first note that this group of mathematicians is part of a larger group that was formed at the Women in Data Science and Mathematics (WiSDM) program at IPAM in Summer 2023. The WiSDM workshop held this past summer was a great beginning for this collaboration, where we made significant advancements in the development of iterative methods for solving tensor linear systems across various data scenarios. Subsequently, our team members have been geographically dispersed, as we live in different cities and time zones. To address this logistical challenge, we have been conducting regular virtual meetings. This week at AIM, where we could be together and concentrate on propelling forward our work on some of our exciting new questions, has been invaluable.

2. Problem Context

Solving large-scale systems of linear equations or linear regressions is one of the most common problems across the data-rich sciences. This problem arises in machine learning as subroutines of several optimization methods [1], in medical imaging [2, 3], in sensor networks [4], and in statistical analysis, to name only a few. In the matrix-vector and matrix-matrix regime, this problem is well-understood with many highly efficient methods with provable guarantees in the literature. For example, the Kaczmarz, Gauss-Seidel, and Jacobi methods are highly popular families of simple iterative methods for solving large-scale linear regressions.

The *Gauss-Seidel (GS)* and *Jacobi* methods are a related family of *column-action* iterative methods which focus on single coordinate (or group coordinate) updates to the iterates; see e.g., [5]. These methods iterate by minimizing a subset of the residual error with respect to a single coordinate; the jth iterate is

$$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} - \frac{A_{i_j}^T(A\boldsymbol{x}_{j-1} - b)}{\|A_{i_j}\|^2}\boldsymbol{e}_{i_j}, \tag{1}$$

where A_{i_j} is the i_jth column of A. These methods, often taught in numerical analysis and numerical linear algebra courses, have found success in subroutines for multigrid methods [6, 7], high performance computing [8], and PDEs [9, 10]. These methods and variants have been analyzed in [11, 12, 13, 14].

Modern data analysis is often challenging not only due to the size of the data, but due to the inherent complexity of the data. Often this modern data is *multi-modal*, with modes representing measurements along different dimensions. These include spatial and temporal dimensions of video data or word and document dimensions of text corpora data. Data of this type is often naturally represented as a higher-order generalization of a matrix, also known as a *tensor*. Development of data analytic methods for higher-order tensor data is far behind that for matrices, creating a setting in which practitioners must first transform their higher-order tensor data into matrices and then apply inadequate matrix-based methods. This approach ignores the natural structure of the data. Furthermore, computations with tensor data often remain challenging even when their matrix counterparts can be taught in introductory linear algebra courses.

Recently, Kaczmarz-type iterative methods have been proposed for a variety of tensor linear systems and regression problems [15, 16, 17]. Additionally, Kaczmarz-type methods have been proposed for a variety of deblurring [18], denoising [19], and dictionary representation-learning [20] imaging applications; each of these can be formulated as a possibly regularized tensor regression problem

$$\min_{\boldsymbol{\mathcal{X}} \in \mathfrak{X}} \|\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}\|_F^2 + \Phi(\boldsymbol{\mathcal{X}}) \tag{2}$$

where $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{m\times n\times p}$ is the measurement operator or dictionary, $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{m\times l\times p}$ represents the measurements or data, $\boldsymbol{\mathcal{X}} \in \mathfrak{X} \subset \mathbb{R}^{n\times l\times p}$ is the signal of interest, and $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}$ is the t-product between $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{X}}$ [21]. The tensor *t-product*, proposed by Kilmer and Martin [18], is a bilinear operation between tensors that allows for the generalization of many matrix algebra definitions and properties to the tensor setting.

The following fact has been useful in our analysis; the tensor t-product can be defined equivalently using a matrix-matrix product and a folding and unfolding operation on the tensor. We define

$$\text{unfold}(\boldsymbol{\mathcal{X}}) = \begin{bmatrix} \boldsymbol{\mathcal{X}}_1 \\ \boldsymbol{\mathcal{X}}_2 \\ \vdots \\ \boldsymbol{\mathcal{X}}_p \end{bmatrix}$$

where $\boldsymbol{\mathcal{X}}_k$ is the kth *frontal slice* of $\boldsymbol{\mathcal{X}}$; that is, using Matlab notation, $\boldsymbol{\mathcal{X}}_k = \boldsymbol{\mathcal{X}}(:,:,k)$. Similarly, fold is the inverse operation to unfold, so

$$\text{fold}\left(\begin{bmatrix} \boldsymbol{\mathcal{X}}_1 \\ \boldsymbol{\mathcal{X}}_2 \\ \vdots \\ \boldsymbol{\mathcal{X}}_p \end{bmatrix}\right) = \boldsymbol{\mathcal{X}}.$$

Finally, the block-circulant matrix of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{m\times n\times p}$ is defined as

$$\text{bcirc}(\boldsymbol{\mathcal{A}}) = \begin{bmatrix} \boldsymbol{\mathcal{A}}_1 & \boldsymbol{\mathcal{A}}_n & \boldsymbol{\mathcal{A}}_{n-1} & \cdots & \boldsymbol{\mathcal{A}}_2 \\ \boldsymbol{\mathcal{A}}_2 & \boldsymbol{\mathcal{A}}_1 & \boldsymbol{\mathcal{A}}_n & \cdots & \boldsymbol{\mathcal{A}}_{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \boldsymbol{\mathcal{A}}_n & \boldsymbol{\mathcal{A}}_{n-1} & \boldsymbol{\mathcal{A}}_{n-2} & \cdots & \boldsymbol{\mathcal{A}}_1 \end{bmatrix}$$

where $\boldsymbol{\mathcal{A}}_k$ is the kth frontal slice of $\boldsymbol{\mathcal{A}}$. With these definitions, we can describe the tensor t-product of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{X}}$ as a folding of the product of the block-circulant matrix of $\boldsymbol{\mathcal{A}}$ and the unfolding of $\boldsymbol{\mathcal{X}}$.

Fact 1. *Given $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{m\times l\times n}$ and $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{l\times p\times n}$, the t-product between these tensors may be equivalently defined as*

$$\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} = \textit{fold}(\textit{bcirc}(\boldsymbol{\mathcal{A}})\textit{unfold}(\boldsymbol{\mathcal{X}})).$$

3. Setting and Method Derivation

While Kaczmarz-type methods are being actively explored in the tensor regression setting, coordinate-wise or column-action methods have been considered far less in the literature. *Our central goals over the three years of the SQuaRE are to propose column-slice-action iterative methods for linear tensor problems with rigorous theoretical guarantees in a variety of problem settings.* This goal is especially important in the tensor system setting when row slices of the measurement tensor are extremely large and cannot be stored in active memory, or the tensor data is naturally stored in column-slice components (e.g., distributed across computational servers or priority indexed by column). In this setting, accessing column-slices of the tensor may be the only reliable form of data access available.

Consider the consistent tensor linear system $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{B}}$ where $\boldsymbol{\mathcal{B}} \in \mathbb{C}^{m\times p\times n}$, $\boldsymbol{\mathcal{A}} \in \mathbb{C}^{m\times l\times n}$, and $\boldsymbol{\mathcal{X}} \in \mathbb{C}^{l\times p\times n}$. We have formulated the tensor version of the randomized Gauss-Seidel for this setting,

$$\boldsymbol{\mathcal{X}}^{(t)} = \boldsymbol{\mathcal{X}}^{(t-1)} + \boldsymbol{\mathcal{E}}_j(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:}(\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}_t) \tag{3}$$

with $\boldsymbol{\mathcal{X}}^{(t)}$ and $\boldsymbol{\mathcal{X}}^{(t)}$ in $\mathbb{C}^{l\times p\times n}$, $\boldsymbol{\mathcal{E}}_j$ in $\mathbb{C}^{l\times 1\times n}$, $\boldsymbol{\mathcal{A}}^*_{:j:}$ in $\mathbb{C}^{1\times m\times n}$, $\boldsymbol{\mathcal{A}}_{:j:}$ in $\mathbb{C}^{m\times 1\times n}$, $\boldsymbol{\mathcal{B}}$ in $\mathbb{C}^{m\times p\times n}$, $\boldsymbol{\mathcal{A}}$ in $\mathbb{C}^{m\times l\times n}$.

3.1. Connection to Classical Methods.

The classical Jacobi and Gauss-Seidel methods are iterative methods used to solve a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where the square matrix $\boldsymbol{A}$ and the vector $\boldsymbol{b}$ are known and the goal is to approximate $\boldsymbol{x}$. The Gauss-Seidel method was developed in the 1800s and is considered one of the first iterative methods developed [22]. It is taught in undergraduate numerical methods courses and is similar to the Jacobi method, with the main difference being when updates are applied.

When considering $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, the matrix $\boldsymbol{A}$ is decomposed into the sum of a strictly lower triangular matrix $\boldsymbol{L}$, a diagonal matrix $\boldsymbol{D}$, and a strictly upper triangular matrix $\boldsymbol{U}$, $\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U}$. This allows the system of linear equations to be rewritten as $\boldsymbol{A}\boldsymbol{x} + \boldsymbol{L}\boldsymbol{x} + \boldsymbol{U}\boldsymbol{x} = \boldsymbol{b}$. The Jacobi method exploits this rewritten system and produces a fixed-point iterative method on the fixed-point equation $\boldsymbol{D}\boldsymbol{x} = -(\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x} + \boldsymbol{b}$ of the form

$$\boldsymbol{x}^{(k)} = -\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})\boldsymbol{x}^{(k-1)} + \boldsymbol{D}^{-1}\boldsymbol{b}.$$

Entry-wise, this takes the form

$$\boldsymbol{x}_i^{(k)} = -\frac{1}{A_{ii}} \sum_{j \neq i} A_{ij} \boldsymbol{x}_j^{(k-1)} + \frac{1}{A_{ii}} \boldsymbol{b}_i.$$

The Gauss-Seidel method, meanwhile, uses the fixed-point equation $(\boldsymbol{D} + \boldsymbol{L})\boldsymbol{x} = -\boldsymbol{U}\boldsymbol{x} + \boldsymbol{b}$ to construct the fixed-point iterative method

$$\boldsymbol{x}^{(\boldsymbol{k})} = -(\boldsymbol{D} + \boldsymbol{L})^{-1} \boldsymbol{U} \boldsymbol{x}^{(k-1)} + (\boldsymbol{D} + \boldsymbol{L})^{-1} \boldsymbol{b}.$$

This is equivalent to $\boldsymbol{x}^{(k)} = -\boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{x}^{(k)} - \boldsymbol{D}^{-1}\boldsymbol{U}\boldsymbol{x}^{(k-1)} + \boldsymbol{D}^{-1}\boldsymbol{b}$, which takes the following form entry-wise:

$$\boldsymbol{x}_i^{(k)} = -\frac{1}{A_{ii}} \sum_{j=1}^{i-1} A_{ij} \boldsymbol{x}_j^{(k)} - \frac{1}{A_{ii}} \sum_{j=i+1}^{n} A_{ij} \boldsymbol{x}_j^{(k-1)} + \frac{1}{A_{ii}} \boldsymbol{b}_i.$$

The convergence properties of the Jacobi and Gauss-Seidel method are dependent on the properties of the matrix $\boldsymbol{A}$, specifically upon the spectral radius of the matrices involved in the Jacobi and Gauss-Seidel updates, $-\boldsymbol{D}^{-1}(\boldsymbol{L} + \boldsymbol{U})$ and $-(\boldsymbol{D} + \boldsymbol{L})^{-1}\boldsymbol{U}$, respectively.

In the recent literature, the method referred to as *randomized Gauss-Seidel* is, in fact, a variant of randomized coordinate descent applied to the least-squares objective. We note below that this can be viewed as a variant of either Jacobi's method or Gauss-Seidel applied to the normal equations $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$ in which only a single coordinate is updated.

In addition to the questions previously identified in our proposal (and reiterated) below, this lead us to a new question:

Question 1. *Can we derive the classical Jacobi and Gauss-Seidel methods for the tensor system $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{B}}$ where $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{m \times m \times p}$? Does it make sense to take the "upper-" and "lower-triangular" parts of a tensor $\boldsymbol{\mathcal{A}}$? Can we emulate the classical convergence proofs for these methods, and if so, what is the spectral radius of the tensor?*

3.2. **Connection to Coordinate Descent.** The Gauss-Seidel method can also be viewed as an instance of the coordinate descent algorithm applied to solving the least square regression problem. For a general unconstrained convex optimization problem

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}),$$

coordinate descent aims to iteratively update each coordinate by solving the smaller optimization

$$x_i^{(t)} = \arg\min_{x} \; f(x_1^t, x_2^{(t)}, \ldots x_{i-1}^{(t)}, x, x_{i+1}^{(t-1)}, \cdots, x_n^{(t-1)}). \tag{4}$$

While the coordinates are updated cyclically in the formula above, other methods of sampling the coordinates, including at random, can be used here.

When $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$, (4) simplifies to solving for $x_i^{(t)}$ in $\nabla_i f(\boldsymbol{x}^{(t)}) = \boldsymbol{0}$, which, for this function f, yields

$$\left[\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x}^{(t)} - \boldsymbol{A}^T \boldsymbol{b}\right]_i = 0.$$

This simplifies to

$$x_i^{(t)} = \frac{-\sum_{j=1}^{k-1} (\boldsymbol{A}^\top \boldsymbol{A})_{ki} \; x_j^{(t)} - \sum_{j=k+1}^{n} (\boldsymbol{A}^\top \boldsymbol{A})_{ki} \; x_j^{(t-1)} + \boldsymbol{A}_{:k}^\top \boldsymbol{b}}{\|\boldsymbol{A}_{:k}\|^2},$$

which matches the iterates of Gauss-Seidel when applied to solve the normal equation $\boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^\top \boldsymbol{b}$; that is, *a cyclically-updating coordinate descent method on the least-squares objective coincides with the classical Gauss-Seidel method applied to the normal equations.*

We now turn our attention to the tensor regression case. We note that *one can derive the update* (3) *as the coordinate descent update on the least-squares objective for the t-product regression problem.* Let $L(\boldsymbol{\mathcal{X}}) = \frac{1}{2}\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}\|_F^2$ and note that we can rewrite this objective, using Fact 1, as

$$L(\boldsymbol{\mathcal{X}}) = \frac{1}{2} \sum_{i'=1}^{mp} \sum_{j=1}^{l} (\text{bcirc}(\boldsymbol{\mathcal{A}})_{i':} \text{unfold}(\boldsymbol{\mathcal{X}})_{:j} - \text{unfold}(B)_{i'j})^2.$$

We then derive the partial derivative

$$\begin{aligned}\frac{\partial L}{\partial X_{sjt}} &= \sum_{i'=1}^{mp} (\text{bcirc}(\boldsymbol{\mathcal{A}})_{i':}\text{unfold}(\boldsymbol{\mathcal{X}})_{:j} - \text{unfold}(\boldsymbol{\mathcal{B}})_{i',j})\text{bcirc}(A)_{i',(t-1)n+s} \\ &= \sum_{i'=1}^{mp} \text{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}})_{i',j}\text{bcirc}(\boldsymbol{\mathcal{A}})_{i',(t-1)n+s} \\ &= \sum_{i'=1}^{mp} \text{bcirc}(\boldsymbol{\mathcal{A}}^\top)_{(t-1)n+s,i'}\text{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}})_{i',j} \\ &= (\text{bcirc}(\boldsymbol{\mathcal{A}}^\top)\text{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}))_{(t-1)n+s,j} \\ &= \text{unfold}(\boldsymbol{\mathcal{A}}^\top(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}))_{(t-1)n+s,j} \\ &= (\boldsymbol{\mathcal{A}}^\top(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}}))_{sjt}.\end{aligned}$$

Thus, $\frac{\partial L}{\partial X_{s::}} = \boldsymbol{\mathcal{A}}_{:s:}^\top(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{B}})$. Now, if we suppose that $\boldsymbol{\mathcal{X}}^{(t)} = \boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{E}}_s\boldsymbol{\mathcal{Z}}$, we may solve for the tensor $\boldsymbol{\mathcal{Z}}$ producing the update which will satisfy the stationary equation $\frac{\partial L}{\partial X_{s::}} = 0$, that is we wish to find $\boldsymbol{\mathcal{Z}}$ so that

$$\begin{aligned}\boldsymbol{\mathcal{A}}_{:s:}^\top(\boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{E}}_s\boldsymbol{\mathcal{Z}}) - \boldsymbol{\mathcal{B}}) = 0 \\ -\boldsymbol{\mathcal{A}}_{:s:}^\top\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{E}}_s\boldsymbol{\mathcal{Z}} = -\boldsymbol{\mathcal{A}}_{:s:}^\top(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{B}}) \\ \boldsymbol{\mathcal{Z}} = (\boldsymbol{\mathcal{A}}_{:s:}^\top\boldsymbol{\mathcal{A}}_{:s:})^{-1}\boldsymbol{\mathcal{A}}_{:s:}^\top(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{B}}).\end{aligned}$$

Thus, we see that *the tRGS update 3 may be derived as the "coordinate" update which produces a stationary point of the objective* $L(\boldsymbol{\mathcal{X}})$. We will be interested in understanding the coordinate descent derivation of *block methods*, which use blocks of columns in each update, in the future:

Question 2. *Are randomized* block *Gauss-Seidel methods derived in the same way? Do these occur as block coordinate descent updates applied to the least-squares objective, L?*

3.3. **Gauss-Seidel Duality with Kaczmarz Methods.** We begin with the primal problem

$$\min_{\boldsymbol{x}} \frac{1}{2}\|\boldsymbol{x}\|^2 \text{ subject to } \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}. \tag{5}$$

We define the Lagrangian

$$L(\boldsymbol{x}, \boldsymbol{\nu}) = \frac{1}{2}||\boldsymbol{x}||^2 + \sum_{i=1}^{m} \nu_i(\boldsymbol{b_i} - \boldsymbol{a}_i^T\boldsymbol{x})$$

where $\boldsymbol{x}$ is the primal variable and $\boldsymbol{\nu}$ is the dual variable. Each ν_i is a Lagrange multiplier associated with the ith equality constraint. We can simplify this equation to

$$L(\boldsymbol{x}, \boldsymbol{\nu}) = \frac{1}{2}\boldsymbol{x}^T\boldsymbol{x} + \boldsymbol{v}^T(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}).$$

To determine the primal to dual variable transformation, we look at the optimum with respect to the primal, that is the $\boldsymbol{x}$ that satisfies the stationary equation

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\nu}) = \boldsymbol{x} - \boldsymbol{A}^\top\boldsymbol{\nu} = 0.$$

This forces $\boldsymbol{x} = \boldsymbol{A}^\top\boldsymbol{\nu}$. This equation relates the primal and dual variables at their respective optimums. To create the dual problem, we use this variable transformation, which yields

$$g(\boldsymbol{\nu}) = \frac{1}{2}\boldsymbol{\nu}^T\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{\nu} + \boldsymbol{\nu}^T\boldsymbol{b} - \boldsymbol{\nu}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\nu}.$$

Simplifying yields the dual problem in the form $g(\boldsymbol{\nu}) = \boldsymbol{\nu}^T\boldsymbol{b} - \frac{1}{2}\boldsymbol{\nu}^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{\nu}$.

We now apply coordinate descent to this dual objective. Recall that coordinate descent algorithms solve optimization problems by minimizing along each coordinate (or coordinate hyperplane). This essentially breaks a complex problem into a number of more tractable subproblems [23]. Assuming that

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\nu}^{(t-1)} - c\boldsymbol{e}_i$$

and that $\boldsymbol{\nu}^{(t)}$ satisfies the stationary equation

$$\nabla_i g(\boldsymbol{\nu}^{(t)}) = 0$$

yields $c = \frac{(\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{\nu}^{(t)} - \boldsymbol{b})_i}{\|\boldsymbol{A}_{i:}\|^2}$. We see that

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\nu}^{(t-1)} - \frac{(\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{\nu}^{(t)} - \boldsymbol{b})_i}{\|\boldsymbol{A}_{i:}\|^2} \boldsymbol{e}_i.$$

Because $\|\boldsymbol{A}_{i:}\|^2 = (\boldsymbol{A}\boldsymbol{A}^\top)_{ii}$ we have the update

$$\boldsymbol{\nu}^{(t)} = \boldsymbol{\nu}^{(t-1)} - \frac{1}{(\boldsymbol{A}\boldsymbol{A}^\top)_{ii}} (\boldsymbol{A}\boldsymbol{A}^\top \boldsymbol{\nu}^{(t)} - \boldsymbol{b})_i \boldsymbol{e_i}.$$

If i is chosen randomly, this is RGS applied to the dual problem. We note that $(\boldsymbol{A}\boldsymbol{A}^\top)_{ii}$ is the Lipschitz constant of the ith component function of

$$g(\boldsymbol{\nu}) = \sum_{i=1}^{m} \nu_i b_i - \frac{1}{2}\nu_i^2 \|\boldsymbol{A}_{i:}\|^2.$$

This leads us to ask whether tRGS and tRK, i.e., the tensor variants of RGS and RK, are related in the same way via a primal and dual variables transformation.

Question 3. *Are tRK and tRGS methods applied to a primal and dual problem and related via a primal variable and dual variable relationship like in the matrix-vector case? If so, is the transformation given by $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{Y}}$ where $\boldsymbol{\mathcal{X}}$ is an iterate of tRK, and $\boldsymbol{\mathcal{Y}}$ is an iterate of tRGS?*

4. Conjecture and Problem 1

Building off the work in [24], which proves convergence in expectation of the block randomized Gauss-Seidel for usual matrix-vector linear systems, our original conjecture for the method defined by update (3) was the following.

Conjecture 1. *Let $\boldsymbol{\mathcal{X}}^\ddagger$ be the tensor of minimal Frobenius norm such that $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\ddagger = \boldsymbol{\mathcal{Y}}$ and $\boldsymbol{\mathcal{X}}^{(t)}$ the t-th approximation of $\boldsymbol{\mathcal{X}}^\ddagger$ given by the update (3) with initial iterate $\boldsymbol{\mathcal{X}}^0$. Under some mild assumptions on $\boldsymbol{\mathcal{X}}^0$ and $\boldsymbol{\mathcal{A}}$, the expected error at the t-th iteration satisfies*

$$\mathbb{E}\left[\|\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{X}}^\ddagger\|_F^2 \mid \boldsymbol{\mathcal{X}}^{(0)}\right] \leq (1-r)^t \|\boldsymbol{\mathcal{X}}^{(0)} - \boldsymbol{\mathcal{X}}^\ddagger\|_F^2$$

where $0 < r < 1$ depends upon the conditioning of the tensor $\boldsymbol{\mathcal{A}}$.

Moreover, one of our main goals for this week at AIM was to work on this conjecture, which we described in the following proposed problem.

Proposed Problem 1. *We will prove Conjecture 1 and identify regimes in which assumptions can be relaxed. We will also extend this method and convergence results to other problem domains, such as regression problems and corrupted systems of linear equations.*

As a result of our discussions and work during this week at AIM, we have refined our original conjecture and proved the fundamental lemmas needed for the corresponding convergence result. This is detailed in the next two subsections.

4.1. **Updates to Conjecture.** Currently, we do not think that the Randomized Gauss-Seidel iterations, whose updates are given by (3), converge to the tensor of minimal Frobenius norm $\boldsymbol{\mathcal{X}}^\ddagger$ unless it is the only solution $\boldsymbol{\mathcal{X}}^*$, which occurs in the overdetermined consistent case. For overdetermined inconsistent tensor linear systems, the algorithm converges to the least-squares solution $\boldsymbol{\mathcal{X}}_{LS}$. Finally, in the underdetermined consistent case, we have convergence of the residual $\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^*$ to zero, but it is unclear to which solution the tRGS iterates converge. Our results are summarized in Table 1.

	overdetermined	underdetermined
consistent	$\boldsymbol{\mathcal{X}}^{(t)}$ converges to $\boldsymbol{\mathcal{X}}^*$	$\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^*\|$ converges to 0
inconsistent	$\boldsymbol{\mathcal{X}}^{(t)}$ converges to $\boldsymbol{\mathcal{X}}_{LS}$	?

Table 1. Convergence of RGS algorithm depending on the conditioning of the tensor $\boldsymbol{\mathcal{A}}$.

The updates to our conjecture and our understanding of the differences amongst the underdetermined and overdetermined cases has led us to the following question:

Question 4. *Can we classify the form of the solution to which the tRGS iterates converge in the consistent, underdetermined case? What happens in the inconsistent, underdetermined case?*

4.2. **Fundamental Lemmas.** We have also proved the following fundamental lemmas which will allow us to prove our refined conjecture. Our first line of investigation is into the form of the operator which defines the tRGS update. We show that under a natural transformation, this is a projection operator, and investigate the norm and structure of the inverse matrix defining this operator; see Section 4.2.1. We next prove orthogonality of the transformation of the sequential iterates under $\boldsymbol{\mathcal{A}}$ and the residual of $\boldsymbol{\mathcal{X}}^{(t)}$; see Section 4.2.2. This provides a Pythagorean theorem-like result for us to appeal to in decomposing the error. We finally prove that the expectation of the residual decreases exponentially with a rate depending upon the conditioning of the projection operators; see Section 4.2.3.

4.2.1. *Structure and Norm of Inverse in Projector.* The tRGS update (3) applies the tensor $\boldsymbol{\mathcal{E}}_j(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:}$ to the residual in each iteration. This tensor, after product with $\boldsymbol{\mathcal{A}}$, yields a projector defined as

$$\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}} = \boldsymbol{\mathcal{A}}_{:j:}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:},$$

where $\boldsymbol{\mathcal{A}}_{:j:}$ is a $m \times 1 \times n$ (tube) tensor.

We are interested in studying the term $(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}$, which appears in the tRGS update. First, from [15], we know that, under the t-product,

$$(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1} = \text{fold}\left(\frac{1}{\sqrt{n}}\mathbf{F}^*\text{diag}(\mathbf{D}^{-1})\right),$$

where $\mathbf{F}$ is the Discrete Fourier Transform (DFT) matrix, and $\mathbf{D}$ is a diagonal matrix such that $\text{bcirc}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:}) = \mathbf{F}^*\mathbf{D}\mathbf{F}$.

We are able to obtain a bound on the norm of $(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}$ by writing $\mathbf{D}$ with more specificity. First, note that each row, say, $\mathbf{a}_i$, $i = 1, \ldots, n$, of $\text{bcirc}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})$ contains permutations of the entries of the same vector, that we denote as $\tilde{\mathbf{a}}$. In [25], the authors prove that each diagonal element i of $\mathbf{D}$ can be written as a DFT of the vectors $\mathbf{a}_i$. We use this fact and the property that the DFT is unitary up to a normalization factor to obtain an upper bound on the spectral norm of $\text{bcirc}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}$.

4.2.2. *Show the orthogonality of* $(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)$ *and* $(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)})$. The essential tool to show this is the interaction of the projection operator $\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}} = \boldsymbol{\mathcal{A}}_{:j:}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:}$. Observe that $\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}$ is a projection operator since $\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}} = \mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}$. Now,

$$\begin{aligned}
\langle \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star, \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)}\rangle =& \langle \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}_{:j:}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star) - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star,\\
& \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}_{:j:}(\boldsymbol{\mathcal{A}}^*_{:j:}\boldsymbol{\mathcal{A}}_{:j:})^{-1}\boldsymbol{\mathcal{A}}^*_{:j:}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star) - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)}\rangle\\
=& \langle \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star) - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star, -\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)\rangle\\
=& \langle (\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star) - \mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star), -\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)\rangle\\
=& \langle (\mathcal{I} - \mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}})(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star), -\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)\rangle\\
=& 0
\end{aligned}$$

4.2.3. *Expectation of residuals.* The orthogonality result established in the previous step, in turn, gives us the following result -

$$\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2 = \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2 - \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)}\|_2^2 \tag{6}$$

Sketch of Convergence of tensor Randomized Gauss-Seidel algorithm:

Taking expectation (conditioned till time $t-1$) on both sides of the Eqn. 6 gives us

$$\begin{aligned}
\mathbb{E}^{(t-1)}\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2 &= \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2 - \mathbb{E}^{(t-1)}\left[\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t)}\|_2^2\right]\\
&= \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2 - \mathbb{E}^{(t-1)}\left[\|\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)\|_2^2\right]\\
&= \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2\left(1 - \frac{\mathbb{E}^{(t-1)}\left[\|\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star)\|_2^2\right]}{\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^\star\|_2^2}\right)
\end{aligned}$$

We can simplify the following term as -

$$\begin{aligned}
&\left[\mathbb{E}^{(t-1)}\|\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_{(t-1)}:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\|_2^2\right] \\
&= \mathbb{E}^{(t-1)}\langle \mathrm{bcirc}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}]\mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}), \mathrm{bcirc}(\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}})\mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\rangle \\
&\underset{(1)}{=} \mathbb{E}^{(t-1)}\langle \mathrm{bcirc}(\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}})\mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}), \mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\rangle \\
&\underset{(2)}{=} \langle \mathrm{bcirc}(\mathbb{E}^{(t-1)}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}])\mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}), \mathrm{unfold}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\rangle \\
&\geq \sigma_{\min}(\mathrm{bcirc}(\mathbb{E}^{(t-1)}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}]))\|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}\|^2
\end{aligned}$$

Step (1) uses the fact that $\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}$ is a projection operator, while step (2) follows from linearity of bcirc. Combining these results together yields the following convergence rate for the tRGS method -

$$\mathbb{E}^{(t-1)}\left[\|\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_{(t-1)}:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\|_2^2\right] \leq \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}\|_2^2\left(1 - \sigma_{\min}(\mathrm{bcirc}(\mathbb{E}^{(t-1)}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_t:}}]))\right)$$

By utilizing the tower property of conditional expectation, we can further claim that -

$$\mathbb{E}\left[\|\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j_{(t-1)}:}}(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(t-1)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star})\|_2^2|\boldsymbol{\mathcal{X}}^{(0)}\right] \leq \|\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{(0)} - \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{X}}^{\star}\|_2^t\left(1 - \sigma_{\min}(\mathrm{bcirc}(\mathbb{E}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}]))\right)$$

Motivated by simplifying this bound and desribing quantities in terms of the original tensor, we come to the following new question:

Question 5. *Can $\sigma_{\min}(bcirc(\mathbb{E}[\mathcal{P}_{\boldsymbol{\mathcal{A}}_{:j:}}]))$ be further simplified under some distribution over the columns? Is this quantity defined already in the tensor literature?*

5. Looking Ahead

While we are not including them here, we completed numerical experiments for the proposed problems listed below.

Proposed Problem 2. *We will complete a thorough suite of numerical experiments on synthetic and real data. In particular, we plan to understand the effect of different conditionings of $\boldsymbol{\mathcal{A}}$ on the empirical and theoretical rates of convergence of Algorithm 3, and to understand any discrepancy between these rates.*

Proposed Problem 3. *We will additionally develop and test an implementation of Algorithm 3 for deblurring problems. This setting's unique challenges and operator structures will motivate new variants of the method and theoretical results.*

We plan to build off of this progress and will complete a suite of numerical experiments which will inform and support our hypotheses, and which will form the foundation of numerical experiments for our manuscript in preparation.

In addition, we would like to investigate the following proposed problems in our future work. These problems came from our original proposal to AIM.

Proposed Problem 4. *We also plan to extend the randomized Gauss-Seidel method described in the algorithm defined by update (3) to the factorized operator setting. We will prove rigorous guarantees for these methods in this setting, which will be novel even for matrix operators.*

Proposed Problem 5. *As previously noted, the aforementioned techniques for solving tensor linear systems utilize the tensor t-product framework. An overarching goal of this project is to generalize these iterative methods to other tensor products (like the CP product* [26]*).*

In addition to these problems we identified before visiting AIM, we have identified future questions in our work during our first week. We will consider Questions 1, 2, 3, 4, and 5 in addition to our original proposed problems.

References

[1] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[2] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography. *J. Theoret. Biol.*, 29:471–481, 1970.

[3] G.T. Herman and L.B. Meyer. Algebraic reconstruction techniques can be made computationally efficient. *IEEE T. Med. Imaging*, 12(3):600–609, 1993.

[4] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, 2001.

[5] G. H. Golub and C. F. Van Loan. *Matrix computations*. JHU Press, 2013.

[6] U. Rüde. *Mathematical and computational techniques for multilevel adaptive methods*. SIAM, 1993.

[7] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Elsevier, 2000.

[8] J. Wolfson-Pou and E. Chow. Distributed Southwell: an iterative method with low communication costs. In *Proc. Int. Conf. High Perform. Comput. Network. Stor. Anal.*, pages 1–13, 2017.

[9] C. Glusa, E. G. Boman, E. Chow, S. Rajamanickam, and D. B. Szyld. Scalable asynchronous domain decomposition solvers. *SIAM J. Sci. Comput.*, 42(6):C384–C409, 2020.

[10] D. B. Magoules, F.and Szyld and C. Venet. Asynchronous optimized Schwarz methods with and without overlap. *Numer. Math.*, 137(1):199–227, 2017.

[11] D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Math. Oper. Res.*, 35(3):641–654, 2010.

[12] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1590–1604, 2015.

[13] P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Math. Program.*, 156:433–484, 2016.

[14] A. Frommer and D. B. Szyld. On the convergence of randomized and greedy relaxation schemes for solving nonsingular linear systems of equations. *Numer. Algorithms*, 92(1):639–664, 2023.

[15] A. Ma and D. Molitor. Randomized kaczmarz for tensor linear systems. *BIT Numerical Mathematics*, 62(1):171–194, 2022.

[16] X. Chen and J. Qin. Regularized Kaczmarz algorithms for tensor recovery. *SIAM J. Imaging Sci.*, 14(4):1439–1471, 2021.

[17] L. Tang, Y. Yu, Y. Zhang, and H. Li. Sketch-and-project methods for tensor linear systems. *Numer. Linear Algebr.*, 30(2):e2470, 2023.

[18] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. A.*, 34(1):148–172, 2013.

[19] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[20] Elizabeth Newman and Misha E Kilmer. Nonnegative tensor patch dictionary approaches for image compression and deblurring applications. *SIAM Journal on Imaging Sciences*, 13(3):1084–1112, 2020.

[21] Misha E Kilmer and Carla D Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[22] Yousef Saad and Henk A Van Der Vorst. Iterative solution of linear systems in the 20th century. *Journal of Computational and Applied Mathematics*, 123(1-2):1–33, 2000.

[23] Hao-Jun Michael Shi, Shenyinying Tu, Yangyang Xu, and Wotao Yin. A primer on coordinate descent algorithms. *arXiv preprint arXiv:1610.00040*, 2016.

[24] W Wu and D Needell. Convergence of the randomized block gauss-seidel method. *SIAM Undergraduate Research Online*, 11:369–382, 2018.

[25] T. De Mazancourt and D. Gerlic. The inverse of a block-circulant matrix. *IEEE Transactions on Antennas and Propagation*, 31(5):808–810, 1983.

[26] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.