Proceedings of the Canadian Society for Mechanical Engineering International Congress
32nd Annual Conference of the Computational Fluid Dynamics Society of Canada
Canadian Society of Rheology Symposium
CSME/CFDSC/CSR 2025
May 25–28, 2025, Montréal, Québec, Canada

LEAST-SQUARES RECONSTRUCTION OF A 3D POTENTIAL FROM ITS MEASURED SPATIAL VELOCITY FIELD

Matthew Harker

Department of Systems Engineering, École de technologie supérieure, Montréal, Canada
*matthew.harker@etsmtl.ca

***Abstract*—This paper describes a new method for the least-squares reconstruction of a hypersurface from measured 3D gradient data, for example, the potential function of a measured velocity field. The novel aspect of the proposed algorithm is that the solution relies on representing 3D data with hypermatrices, and obtaining the least-squares solution to the problem as the solution of a hypermatrix equation, i.e., a set of linear equations that equates hypermatrices. The new approach enables the solution of the reconstruction problem over an $l \times m \times n$ hypersurface in $\mathcal{O}(n^4)$ time (with $l \sim m \sim n$), whereas a straightforward approach of solving the problem with a standard least-squares approach (vectorization) would yield an order $\mathcal{O}(n^9)$ algorithm. The new algorithm is therefore five orders of magnitude faster than the state-of-the-art. The new algorithm is tested with synthetic data with synthetic noise. The method is, however, applicable to the problem of reconstructing a potential from velocity data measured, for example, via particle image velocimetry.

***Keywords-component*—potential function ; reconstruction from gradients ; Sylvester equation ; hypermatrix equation ; least-squares**

I. Introduction

The integration of a function of several variables from its gradient is an important inverse problem in physics, finding applications in the mechanics, imaging, microscopy, as well as in mathematics in the solution of differential equations. In engineering, many types of measurement function by measuring the gradient of a quantity of interest, for example, photometric stereo is a method of recovering surface structure by measuring the surface gradient from images of a surface under different lighting conditions [28]. Similar principles are used to measure gradients with scanning electron and scanning helium microscopes [22]. In fluid mechanics, the technique of particle image velocimetry is for estimating velocity fields in fluids which can then be integrated to obtain the local pressure potential [13]. Hence, the efficient and accurate reconstruction of a hypersurface from 3D gradient data has countless possible uses in engineering and physics. In the analytic case, a function of several variables, $f(x, y, z)$, can be obtained from its gradient by integration, provided that the gradient satisfies the integrability conditions [1],

$$\frac{\partial^2 f}{\partial x \partial y} = \frac{\partial^2 f}{\partial y \partial x} \qquad \frac{\partial^2 f}{\partial x \partial z} = \frac{\partial^2 f}{\partial z \partial x} \qquad \frac{\partial^2 f}{\partial y \partial z} = \frac{\partial^2 f}{\partial z \partial y} \ . \tag{1}$$

In the discrete case, numerical integration of the gradient field proceeds sequentially, and even with exact data, the approximate formulas will yield an increasing error in the direction of integration (e.g., [29]). However, when noise is present in the data, the integrability conditions will no longer hold, and some form of approximation is therefore essential [10].

An early solution of the gradient reconstruction problem in 2D is the method proposed in [12]. At the time, it was too computationally expensive to solve the linear system of equations, so that an iterative scheme was proposed. Unfortunately, the proposed method has no fixed point, so a heuristic stopping point must be set. A method based on the FFT was subsequently proposed [4], however, it and similar methods [15] require the unrealistic assumption that the surface to be reconstructed is periodic. Further solutions are based on the "vectorization" of the surface, i.e., the representation of an $m \times n$ array as an mn-vector [14], [18], [24], where the coefficient matrix has a sum of Kronecker products structure. While there is an effort into developing efficient numerical solutions for Kronecker product structures in least-squares problems [7], [27], and algorithms exist for large sparse least-squares problems [19], these algorithms are all fundamentally of order $\mathcal{O}(n^6)$ for the 2D $m \times n$ reconstruction problem with $m \sim n$. Recent articles indicate there is still a tendency towards these vectorized solutions with low accuracy numerical approximation formulas [20], [21],

The author acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

[30].

In [8], it was shown that the 2D reconstruction problem could be solved with higher accuracy with an order $\mathcal{O}(n^3)$ algorithm, which showed that the normal equations of the least-squares problem were a Sylvester matrix equation. Further developments showed that given the inverse nature of the problem, several types of regularization could be introduced in the same framework without compromising the three orders of magnitude improvement in computation time [10]. The premise of the present paper is that a three dimensional hypersurface or potential function can be discretized as a 3D hypermatrix, i.e., the function,

$$\Phi = f\left(x, y, z\right) \;, \tag{2}$$

can be discretized as the hypermatrix,

$$\mathsf{\Phi} = \left[f\left(x_j, y_i, z_k\right)\right]_{i,j,k=1}^{l,m,n} \;. \tag{3}$$

The least-squares cost function of the reconstruction problem is formulated in terms of a hypermatrix Frobenius norm, and the corresponding normal equations are shown to be what we term a *hypermatrix equation*. An algorithm is developed to solve this hypermatrix equation with the asymptotic complexity of $\mathcal{O}(n^4)$ for an $l \times m \times n$ discretization of the potential function where $l \sim m \sim n$. For comparison, the hypermatrix equation represents $lmn \sim n^3$ equations, which using typical matrix methods is solved in order $\mathcal{O}(n^9)$ time. The new reconstruction algorithm is therefore five orders of magnitude faster than the state-of-the-art.

The computational advantage of the proposed algorithm is rooted in the Sylvester matrix equation [26], and its efficient numerical solution [2], [6], which solves the 2D reconstruction problem. Problems in 3D (and 4D, should the case arise) can be solved by the notion of the Sylvester hypermatrix equation, first introduced and solved in [11]. The present algorithm is an extension of the method that is implemented in the Matlab toolbox [9], which has been used in many real-world applications [5]. The contributions of this paper are:

1) A new $\mathcal{O}(n^4)$ algorithm is proposed for the reconstruction of a 3D potential from its measured gradient/velocity field[1].
2) The Sylvester equation is extended to the Sylvester hypermatrix equation for solving 3D least-squares problems for the first time (Appendix A).
3) Introduction of a computational framework that opens the door to variations of the solution, such as regularization methods.

The proposed algorithm is verified using synthetic data with added synthetic noise to demonstrate that the reconstruction algorithm produces accurate reconstructions even when real-world measurement data is used. Variations of the 2D algorithm include reconstruction over conformal domain shapes [23], constrained reconstruction, spectral regularization, and Tikhonov regularization; the present algorithm can be extended to these variations in the same manner.

[1]MATLAB Code for the algorithm will made available at : mathworks.com/matlabcentral/profile/authors/4201723

II. Preliminaries

The proposed algorithm makes use of numerical differentiation matrices, which appear to have first been used in [16] specifically for problems that required boundary conditions. The general case of such matrices that did not require boundary conditions and use higher order numerical differentiation formulas were developed in [8]. The general principle is that numerical differentiation formulas (e.g., from [3]) over the abscissae $x = a, a + h_1, a + h_1 + h_2$ can be collected and put into matrix form, such as,

$$\begin{bmatrix} f'(a) \\ f'(a+h_1) \\ f'(a+h_1+h_2) \end{bmatrix} \approx \tag{4}$$
$$\begin{bmatrix} \frac{-2h_1-h_2}{h_1(h_1+h_2)} & \frac{h_1+h_2}{h_1 h_2} & -\frac{h_1}{(h_1+h_2)h_2} \\ -\frac{h_2}{h_1(h_1+h_2)} & \frac{h_2-h_1}{h_1 h_2} & \frac{h_1}{(h_1+h_2)h_2} \\ \frac{h_2}{h_1(h_1+h_2)} & \frac{-h_1-h_2}{h_1 h_2} & \frac{2h_2+h_1}{(h_1+h_2)h_2} \end{bmatrix} \begin{bmatrix} f(a) \\ f(a+h_1) \\ f(a+h_1+h_2) \end{bmatrix}$$

By using local numerical differentiation formulas for n points, we obtain a general form of the discretized derivative of a function in the form,

$$\boldsymbol{f}' \approx \mathsf{D}\boldsymbol{f} \;. \tag{5}$$

With such a definition, the proposed algorithm does not require that the data be at evenly spaced intervals. An important property of a numerical differentiation matrix is that we have,

$$\mathsf{D}\boldsymbol{e} = \boldsymbol{0} \;, \tag{6}$$

where $\boldsymbol{e}$ is a vector of ones, i.e., the property that the numerical derivative of a constant function is zero, and further that the matrix is rank one deficient.

The numerical differentiation operators can be applied to hypermatrices in analogy to partial derivatives of a function of several variables. An $l \times m \times n$ hypermatrix is defined as [17],

$$\mathsf{X} = \left[x_{ijk}\right]_{i,j,k=1}^{l,m,n} \;. \tag{7}$$

The Frobenius norm of a hypermatrix, i.e., square root of the sum of squares of all entries, is defined here as,

$$\|\mathsf{X}\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n} x_{ijk}^2} \;. \tag{8}$$

Given the $p \times l$ matrix A, the $q \times m$ matrix B, and the $r \times n$ matrix C, the multiplication along the three dimensions of the hypermatrix yields the hypermatrix X' such that,

$$x'_{\alpha\beta\gamma} = \sum_{i=1}^{l}\sum_{j=1}^{m}\sum_{k=1}^{n} a_{\alpha i} b_{\beta j} c_{\gamma k} x_{i,j,k} \;. \tag{9}$$

For conciseness, this operation will be denoted [17],

$$\mathsf{X}' = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \cdot \mathsf{X} \;, \tag{10}$$

whereby the result, X', is a $p \times q \times r$ hypermatrix. In two dimensions, this notation is equivalent to the operation,

$$(\mathsf{A}, \mathsf{B}) \cdot \mathsf{X} = \mathsf{A}\mathsf{X}\mathsf{B}^{\mathrm{T}} \;. \tag{11}$$

III. Reconstruction from Gradients

In principle, the reconstruction problem involves finding a discretized hypersurface F such that its measured gradient with hypermatrix components $\hat{\mathsf{F}}_y$, $\hat{\mathsf{F}}_x$, and $\hat{\mathsf{F}}_z$, is approximately equal its numerical gradient, i.e.,

$$\begin{aligned}\hat{\mathsf{F}}_y &\approx (\mathsf{D}_y, \mathsf{I}_m, \mathsf{I}_n) \cdot \mathsf{F}\\ \hat{\mathsf{F}}_x &\approx (\mathsf{I}_l, \mathsf{D}_x, \mathsf{I}_n) \cdot \mathsf{F}\\ \hat{\mathsf{F}}_z &\approx (\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z) \cdot \mathsf{F}\ ,\end{aligned} \tag{12}$$

where the derivatives are ordered in such a manner as to have a right handed coordinate system that aligns with the hypermatrix indexing. The solution that ensures that this is true in the least-squares sense is therefore the hypermatrix F that minimizes the corresponding cost-function,

$$\begin{aligned}\epsilon\left(\mathsf{F}\right) = &\left\|(\mathsf{D}_y, \mathsf{I}_m, \mathsf{I}_n) \cdot \mathsf{F} - \hat{\mathsf{F}}_y\right\|_{\mathrm{F}}^2 + \left\|(\mathsf{I}_l, \mathsf{D}_x, \mathsf{I}_n) \cdot \mathsf{F} - \hat{\mathsf{F}}_x\right\|_{\mathrm{F}}^2\\ &+ \left\|(\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z) \cdot \mathsf{F} - \hat{\mathsf{F}}_z\right\|_{\mathrm{F}}^2\ .\end{aligned} \tag{13}$$

In order to find the minimum, we require the gradient of the cost-function with respect to the unknown hypermatrix. This gradient is derived in Appendix A, with the formula given in Equation (48), from which we obtain the corresponding gradient hypermatrix, i.e.,

$$\begin{aligned}\frac{\partial \epsilon\left(\mathsf{F}\right)}{\partial \mathsf{F}} = &\left(\mathsf{D}_y^{\mathrm{T}}, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \Big((\mathsf{D}_y, \mathsf{I}_m, \mathsf{I}_n) \cdot \mathsf{F} - \hat{\mathsf{F}}_y\Big)\\ &+ \left(\mathsf{I}_l, \mathsf{D}_x^{\mathrm{T}}, \mathsf{I}_n\right) \cdot \Big((\mathsf{I}_l, \mathsf{D}_x, \mathsf{I}_n) \cdot \mathsf{F} - \hat{\mathsf{F}}_x\Big)\\ &+ \left(\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z^{\mathrm{T}}\right) \cdot \Big((\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z) \cdot \mathsf{F} - \hat{\mathsf{F}}_z\Big)\ .\end{aligned} \tag{14}$$

Equating the gradient to zero and expanding, we obtain,

$$\begin{aligned}&\left(\mathsf{D}_y^{\mathrm{T}}\mathsf{D}_y, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \mathsf{F} + \left(\mathsf{I}_l, \mathsf{D}_x^{\mathrm{T}}\mathsf{D}_x, \mathsf{I}_n\right) \cdot \mathsf{F} + \left(\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z^{\mathrm{T}}\mathsf{D}_z\right) \cdot \mathsf{F}\\ &= \left(\mathsf{D}_y^{\mathrm{T}}, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \hat{\mathsf{F}}_y + \left(\mathsf{I}_l, \mathsf{D}_x^{\mathrm{T}}, \mathsf{I}_n\right) \cdot \hat{\mathsf{F}}_x + \left(\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_z\ .\end{aligned} \tag{15}$$

This equation, which equates hypermatrices, can be referred to as a hypermatrix Sylvester equation, which was proposed and solved in [11]. This hypermatrix equation represents a system of lmn linear equations in lmn unknowns, however, because of its particular structure can be solved in a much more efficient manner than the equivalent linear system. In this case, it is also of note that the set of equations is rank one deficient. This is due to the fact that the coefficient matrices are numerical differentiation matrices, which as noted must satisfy the identities,

$$\mathsf{D}_y \boldsymbol{e} = \mathbf{0}\ , \qquad \mathsf{D}_x \boldsymbol{e} = \mathbf{0}\ , \qquad \text{and} \qquad \mathsf{D}_z \boldsymbol{e} = \mathbf{0}\ , \tag{16}$$

that is, that the derivatives of constant functions must be zero. Thus, a constant potential function, which we may denote as,

$$\mathsf{F}_c = (\boldsymbol{e}, \boldsymbol{e}, \boldsymbol{e}) \cdot c\ , \tag{17}$$

is a solution to the normal equations regardless of the value of c, which aligns with the well-known fact that a function can be reconstructed from its gradient only up to a constant of integration. Some care must be therefore be taken in obtaining the solution of the normal equations, which in this case effectively have a one dimensional null-space. The solution to the 2D problem obtained in [10] can be generalized to the 3D problem in order to reduce the set of equations to an $(l-1)\times(m-1)\times(n-1)$ hypermatrix equation with three auxiliary Sylvester equations (2D), and three simple linear systems of equations. To accomplish this, we define the three Householder reflections [7],

$$\mathsf{P}_y = \mathsf{I}_l - 2\frac{\boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}}{\boldsymbol{u}^{\mathrm{T}}\boldsymbol{u}} \qquad \mathsf{P}_x = \mathsf{I}_m - 2\frac{\boldsymbol{v}\boldsymbol{v}^{\mathrm{T}}}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} \qquad \mathsf{P}_z = \mathsf{I}_n - 2\frac{\boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}}{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}}\ , \tag{18}$$

with the vectors $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$, respectively of lengths l, m, and n, are given in terms of $\boldsymbol{e}$ and coordinate vectors, $\boldsymbol{e}_1$, as,

$$\boldsymbol{u} = \boldsymbol{e} + \sqrt{l}\boldsymbol{e}_1\ , \qquad \boldsymbol{v} = \boldsymbol{e} + \sqrt{m}\boldsymbol{e}_1\ , \qquad \boldsymbol{w} = \boldsymbol{e} + \sqrt{n}\boldsymbol{e}_1\ , \tag{19}$$

which will effectively transform the null-space to a more convenient basis. We thereby define the transformation of the solution of the equation as,

$$\mathsf{F} = (\mathsf{P}_y, \mathsf{P}_x, \mathsf{P}_z) \cdot \mathsf{G}\ , \tag{20}$$

in terms of an unknown hypermatrix G and substitute it into the normal equations to obtain,

$$\begin{aligned}&\left(\mathsf{D}_y^{\mathrm{T}}\mathsf{D}_y\mathsf{P}_y, \mathsf{P}_x, \mathsf{P}_z\right) \cdot \mathsf{G} + \left(\mathsf{P}_y, \mathsf{D}_x^{\mathrm{T}}\mathsf{D}_x\mathsf{P}_x, \mathsf{P}_z\right) \cdot \mathsf{G}\\ &+ \left(\mathsf{P}_y, \mathsf{P}_x, \mathsf{D}_z^{\mathrm{T}}\mathsf{D}_z\mathsf{P}_z\right) \cdot \mathsf{G} = \left(\mathsf{D}_y^{\mathrm{T}}, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \hat{\mathsf{F}}_y\\ &+ \left(\mathsf{I}_l, \mathsf{D}_x^{\mathrm{T}}, \mathsf{I}_n\right) \cdot \hat{\mathsf{F}}_x + \left(\mathsf{I}_l, \mathsf{I}_m, \mathsf{D}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_z\ .\end{aligned} \tag{21}$$

We subsequently multiply the hypermatrix equation by the matrix triple $\left(\mathsf{P}_y^{\mathrm{T}}, \mathsf{P}_x^{\mathrm{T}}, \mathsf{P}_z^{\mathrm{T}}\right)$ to obtain,

$$\begin{aligned}&\left(\mathsf{P}_y^{\mathrm{T}}\mathsf{D}_y^{\mathrm{T}}\mathsf{D}_y\mathsf{P}_y, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \mathsf{G} + \left(\mathsf{I}_l, \mathsf{P}_x^{\mathrm{T}}\mathsf{D}_x^{\mathrm{T}}\mathsf{D}_x\mathsf{P}_x, \mathsf{I}_n\right) \cdot \mathsf{G}\\ &+ \left(\mathsf{I}_l, \mathsf{I}_m, \mathsf{P}_z^{\mathrm{T}}\mathsf{D}_z^{\mathrm{T}}\mathsf{D}_z\mathsf{P}_z\right) \cdot \mathsf{G} = \left(\mathsf{P}_y^{\mathrm{T}}\mathsf{D}_y^{\mathrm{T}}, \mathsf{P}_x^{\mathrm{T}}, \mathsf{P}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_y\\ &+ \left(\mathsf{P}_y^{\mathrm{T}}, \mathsf{P}_x^{\mathrm{T}}\mathsf{D}_x^{\mathrm{T}}, \mathsf{P}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_x + \left(\mathsf{P}_y^{\mathrm{T}}, \mathsf{P}_x^{\mathrm{T}}, \mathsf{P}_z^{\mathrm{T}}\mathsf{D}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_z\ .\end{aligned} \tag{22}$$

Noting that Householder reflections are orthonormal transformations, i.e.,

$$\mathsf{P}_y^{\mathrm{T}}\mathsf{P}_y = \mathsf{I}_l\ , \qquad \mathsf{P}_x^{\mathrm{T}}\mathsf{P}_x = \mathsf{I}_m\ , \qquad \mathsf{P}_z^{\mathrm{T}}\mathsf{P}_z = \mathsf{I}_n\ , \tag{23}$$

the application of orthogonal transformations in this manner does not change the solution to the equation, whereby, the reflections have been constructed such that,

$$\begin{aligned}\hat{\mathsf{A}} &= \mathsf{D}_y\mathsf{P}_y = \begin{bmatrix}\mathbf{0} & \hat{\mathsf{D}}_y\end{bmatrix}\\ \hat{\mathsf{B}} &= \mathsf{D}_x\mathsf{P}_x = \begin{bmatrix}\mathbf{0} & \hat{\mathsf{D}}_x\end{bmatrix}\\ \hat{\mathsf{C}} &= \mathsf{D}_z\mathsf{P}_z = \begin{bmatrix}\mathbf{0} & \hat{\mathsf{D}}_z\end{bmatrix}\ ,\end{aligned} \tag{24}$$

and the null vectors of the modified differentiation matrices are the coordinate vectors, $\boldsymbol{e}_1$. Further, we have,

$$\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}} = \mathsf{P}_y^{\mathrm{T}}\mathsf{D}_y^{\mathrm{T}}\mathsf{D}_y\mathsf{P}_y = \begin{bmatrix}0 & \mathbf{0}^{\mathrm{T}}\\ \mathbf{0} & \hat{\mathsf{D}}_y^{\mathrm{T}}\hat{\mathsf{D}}_y\end{bmatrix}\ . \tag{25}$$

If we then let,

$$\begin{aligned}\tilde{\mathsf{F}}_y &= \left(\mathsf{I}_l, \mathsf{P}_x^{\mathrm{T}}, \mathsf{P}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_y\\ \tilde{\mathsf{F}}_x &= \left(\mathsf{P}_y^{\mathrm{T}}, \mathsf{I}_m, \mathsf{P}_z^{\mathrm{T}}\right) \cdot \hat{\mathsf{F}}_x\\ \tilde{\mathsf{F}}_z &= \left(\mathsf{P}_y^{\mathrm{T}}, \mathsf{P}_x^{\mathrm{T}}, \mathsf{I}_n\right) \cdot \hat{\mathsf{F}}_z\ ,\end{aligned} \tag{26}$$

then the hypermatrix equation can be written as,

$$\left(\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}}, \mathsf{I}_m, \mathsf{I}_n\right) \cdot \mathsf{G} + \left(\mathsf{I}_l, \hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}, \mathsf{I}_n\right) \cdot \mathsf{G} + \left(\mathsf{I}_l, \mathsf{I}_m, \hat{\mathsf{C}}^{\mathrm{T}}\hat{\mathsf{C}}\right) \cdot \mathsf{G}$$
$$= \left(\hat{\mathsf{A}}^{\mathrm{T}}, \mathsf{I}_m, \mathsf{I}_n^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_y + \left(\mathsf{I}_l, \hat{\mathsf{B}}^{\mathrm{T}}, \mathsf{I}_n\right) \cdot \tilde{\mathsf{F}}_x + \left(\mathsf{I}_l, \mathsf{I}_m, \hat{\mathsf{C}}^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_z , \tag{27}$$

where the new coefficient matrices have strategically placed zero structures. In order to solve this hypermatrix equation, we introduce the $2 \times 2 \times 2$ block hypermatrix partitioning of G by means of the following permutation matrices,

$$\mathsf{Q}_l = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} \\ \mathsf{I}_{l-1} \end{bmatrix}, \qquad \mathsf{Q}_m = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} \\ \mathsf{I}_{m-1} \end{bmatrix}, \qquad \mathsf{Q}_n = \begin{bmatrix} \mathbf{0}^{\mathrm{T}} \\ \mathsf{I}_{n-1} \end{bmatrix} , \tag{28}$$

such that we have the sub-hypermatrices,

$$\begin{aligned} \mathsf{G}_{000} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \mathsf{G} & \mathsf{G}_{001} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \mathsf{G} \\ \mathsf{G}_{100} &= \left(\mathsf{Q}_l^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \mathsf{G} & \mathsf{G}_{101} &= \left(\mathsf{Q}_l^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \mathsf{G} \\ \mathsf{G}_{010} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \mathsf{G} & \mathsf{G}_{011} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \mathsf{G} \\ \mathsf{G}_{110} &= \left(\mathsf{Q}_l^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \mathsf{G} & \mathsf{G}_{111} &= \left(\mathsf{Q}_l^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \mathsf{G} . \end{aligned} \tag{29}$$

which is, in fact, a scalar, three vectors (1D), three matrices (2D) and one hypermatrix (3D). Multiplying the hypermatrix equation by each of the coefficient triplets in the above, we obtain the scalar equation,

$$(0, 0, 0) \cdot \mathsf{G}_{000} = 0 , \tag{30}$$

the three vector equations,

$$\begin{aligned} \left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}}\mathsf{Q}_l, 1, 1\right) \cdot \mathsf{G}_{100} &= \left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_y \\ \left(1, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\mathsf{Q}_m, 1\right) \cdot \mathsf{G}_{010} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_x \\ \left(1, 1, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\hat{\mathsf{C}}\mathsf{Q}_n\right) \cdot \mathsf{G}_{001} &= \left(\boldsymbol{e}_1^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_z . \end{aligned} \tag{31}$$

the three Sylvester matrix equations,

$$\left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}}\mathsf{Q}_l, \mathsf{I}_{m-1}, 1\right) \cdot \mathsf{G}_{110} + \left(\mathsf{I}_{l-1}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\mathsf{Q}_l, 1\right) \cdot \mathsf{G}_{110}$$
$$= \left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_y + \left(\mathsf{Q}_l^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_x \tag{32}$$

$$\left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}}\mathsf{Q}_l, 1, \mathsf{I}_{n-1}\right) \cdot \mathsf{G}_{101} + \left(\mathsf{I}_{l-1}, 1, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\hat{\mathsf{C}}\mathsf{Q}_n\right) \cdot \mathsf{G}_{101}$$
$$= \left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_y + \left(\mathsf{Q}_l^{\mathrm{T}}, \boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_z \tag{33}$$

$$\left(\mathsf{I}_{l-1}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\mathsf{Q}_l, 1\right) \cdot \mathsf{G}_{011} + \left(\mathsf{I}_{l-1}, 1, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\hat{\mathsf{C}}\mathsf{Q}_n\right) \cdot \mathsf{G}_{011}$$
$$= \left(\boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_x + \left(\boldsymbol{e}_1^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}\mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_z , \tag{34}$$

and finally the $(l-1)\times(m-1)\times(n-1)$ hypermatrix equation,

$$\left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}\hat{\mathsf{A}}\mathsf{Q}_l, \mathsf{I}_{m-1}, \mathsf{I}_{n-1}\right) \cdot \mathsf{G}_{111}$$
$$+ \left(\mathsf{I}_{l-1}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}\hat{\mathsf{B}}\mathsf{Q}_m, \mathsf{I}_{n-1}\right) \cdot \mathsf{G}_{111}$$
$$+ \left(\mathsf{I}_{l-1}, \mathsf{I}_{m-1}, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\hat{\mathsf{C}}\mathsf{Q}_n\right) \cdot \mathsf{G}_{111}$$
$$= \left(\mathsf{Q}_l^{\mathrm{T}}\hat{\mathsf{A}}^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_y + \left(\mathsf{Q}_l^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}\hat{\mathsf{B}}^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_x$$
$$+ \left(\mathsf{Q}_l^{\mathrm{T}}, \mathsf{Q}_m^{\mathrm{T}}, \mathsf{Q}_n^{\mathrm{T}}\hat{\mathsf{C}}^{\mathrm{T}}\right) \cdot \tilde{\mathsf{F}}_z . \tag{35}$$

Clearly the element G_{000} can take on any value and represents the constant of integration. The linear equations are solved in order $\mathcal{O}(n^2)$ time by any standard solver. The Sylvester equations can be solved by means of the Bartels-Stewart algorithm [2] in order $\mathcal{O}(n^3)$ time. The numerical solution of the three term hypermatrix equation in $\mathcal{O}(n^4)$ time was developed in [11], and is effectively and extension of the Bartels-Stewart algorithm. The hypermatrix in its reduced form, i.e., $(l-1) \times (m-1) \times (n-1)$, is full rank and thereby has a unique solution. The numerical solutions to the Equations (30) through (35) therefore provide the elements of the solution G^* of the hypermatrix Equation (21). Finally, the reconstructed surface, F_{LS}, is obtained by substituting the solution into the transformation in Equation (20), i.e.,

$$\mathsf{F}_{\mathrm{LS}} = (\mathsf{P}_y, \mathsf{P}_x, \mathsf{P}_z) \cdot \mathsf{G}^* , \tag{36}$$

which is the solution of the normal equations in Equation (15), up to a constant of integration.

IV. Numerical Testing

In order to test the proposed algorithm, we have generated synthetic data from a transcendental function in three dimensions of the form,

$$F(\boldsymbol{x}) = F_0(\boldsymbol{x}) + \sum_{k=1}^{q} A_k \exp\left(-\left(\boldsymbol{x} - \boldsymbol{x}_k\right)^{\mathrm{T}} \mathsf{M}_k^{-1} \left(\boldsymbol{x} - \boldsymbol{x}_k\right)\right) , \tag{37}$$

that is, Gaussian peaks are added for local structure, and the function,

$$F_0(\boldsymbol{x}) = \frac{1}{3}\left(x - \frac{3}{2}\right)^2 + \frac{x}{4}\sin(4y) + \frac{z}{5} , \tag{38}$$

gives the surface global structure (Figure 1). This type of surface mimics variations in pressure, temperature, or density and is transcendental in nature, meaning that polynomial interpolation functions will not be able to model it exactly. Synthetic noise with a standard deviation of 5% of the amplitude of the function has been added to the exact gradient of the function to simulate a measurement situation where the gradient is not integrable (Figure 2). The reconstruction error of the algorithm is given as the difference between the reconstructed hypersurface and the original hypersurface, i.e.,

$$\mathsf{R} = \mathsf{F} - \mathsf{F}_{\mathrm{LS}} , \tag{39}$$

and is shown in Figure 3. The histogram of the residual errors is shown in Figure 4, which shows that the reconstruction errors are Gaussian when the error in the gradient field is Gaussian. This result indicates that the proposed algorithm is an unbiased least-squares estimator of the potential function when the measurement error is Gaussian. As indicated, the proposed algorithm is of $\mathcal{O}(n^4)$ asymptotic complexity, whereas it does not appear that there exists any other algorithm that is faster than order $\mathcal{O}(n^9)$. Rough computation times for the algorithm on a modern laptop are about 0.015 seconds for a $21 \times 28 \times 36$ hypersurface (roughly $20K$ grid points), about 1.4 seconds for a $130 \times 110 \times 140$ hypersurface (roughly $2M$

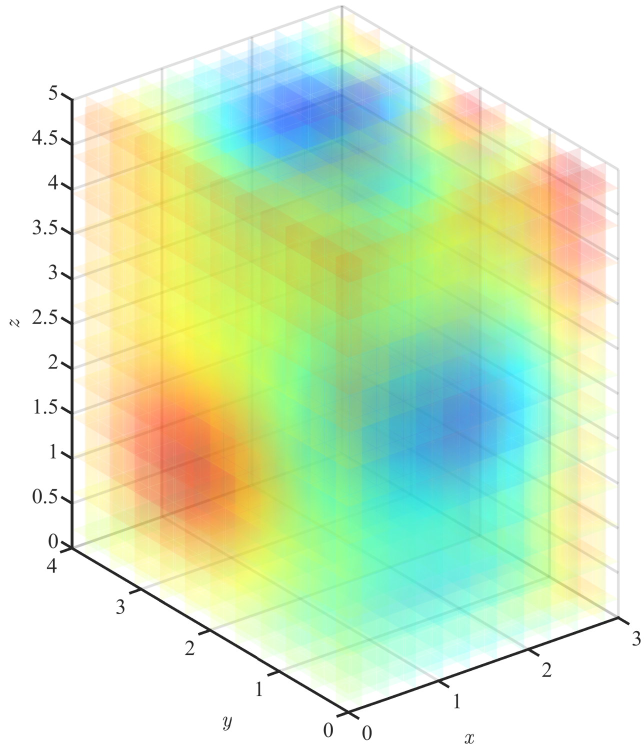

Figure. 1. The synthetic potential function defined by Equation (37) from which the synthetic velocity data is computed. The $130 \times 110 \times 140$ grid of data is visualised as "slices" of the volume.

grid points), and about 345 seconds (about 5.7 minutes) for a $521 \times 512 \times 563$ hypersurface (roughly $150M$ grid points). It is, however, of note that a portion of the computation can be done offline, if the user's aim is fast repeated reconstruction from measured velocity fields.

V. Outlook

We have shown that a commonly used algorithm for the reconstruction of a surface from a 2D gradient field can be extended to 3D gradient fields. The newly proposed 3D reconstruction algorithm can also support regularization methods such as Tikhonov regularization, constrained regularization, and spectral regularization, following [10].

References

[1] R.A. Adams. *Calculus: Several Variables*. Pearson Addison Wesley, sixth edition, 2006.

[2] R.H. Bartels and G.W. Stewart. Algorithm 432: Solution of the matrix equation AX + XB = C. *Comm. ACM*, 15:820–826, 1972.

[3] R.L. Burden and J.D. Faires. *Numerical Analysis*. Thomson Learning, Inc., eighth edition, 2005.

[4] R. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE PAMI*, 10(4):439–451, 1988.

[5] G.-H. Go, D.-G. Lee, J. Oh, G. Song, D. Lee, and M. Jang. Meta Shack–Hartmann wavefront sensor with large sampling density and large angular field of view: phase imaging of complex objects. *Light: Science & Applications*, 13(1):187, 2024.

[6] G.H. Golub, S. Nash, and C.F. Van Loan. A Hessenberg-Schur method for the problem AX+XB = C. *IEEE Trans. on Automatic Control*, 24(6):909–913, 1979.

[7] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4th edition, 2013.

[8] M. Harker and P. O'Leary. Least squares surface reconstruction from measured gradient fields. In *CVPR 2008*, pages 1–7. IEEE, 2008.

[9] M. Harker and P. O'Leary. Matlab toolbox for the regularized surface reconstruction from gradients. In *Proc.SPIE*, volume 9534, page 95341E, 2015.

[10] M. Harker and P. O'Leary. Regularized reconstruction of a surface from its measured gradient field. *Journal of Mathematical Imaging and Vision*, 51(1):46–70, 2015.

[11] M. Harker and P. O'Leary. Numerical solution of the anomalous diffusion equation in a rectangular domain via hypermatrix equations. *IFAC-PapersOnLine*, 50(1):9730–9735, 2017.

[12] B.K.P. Horn and M.J. Brooks. The variational approach to shape from shading. *Computer Vision, Graphics, and Image Processing*, 33:174–208, 1986.

[13] Y.J. Jeon, G. Gomit, T. Earl, L. Chatellier, and L. David. Sequential least-square reconstruction of instantaneous pressure field around a body from tr-piv. *Experiments in Fluids*, 59(2):27, 2018.

[14] B. Karaçalı and W. Snyder. Noise reduction in surface reconstruction from a given gradient field. *International Journal of Computer Vision*, 60(1):25–44, 2004.

[15] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *IEEE ICCV*, pages 994–1001, 2005.

[16] C. Lanczos. *Linear Differential Operators*. Dover, Mineola, NY, 1997.

[17] L.-H. Lim. *Tensors and Hypermatrices, in Handbook of Linear Algebra*, book section 15. CRC Press, Boca Raton, FL, second edition, 2014.

[18] H.-S. Ng, T.-P. Wu, and C.-K. Tang. Surface-from-gradients without discrete integrability enforcement: A Gaussian kernel approach. *IEEE PAMI*, 32(11):2085–2099, 2010.

[19] C.C. Paige and M.A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least-squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, 1982.

[20] Y. Quéau, J.-D. Durou, and J.-F. Aujol. Normal integration: A survey. *Journal of Mathematical Imaging and Vision*, 60(4):576–593, 2018.

[21] Y. Quéau, J.-D. Durou, and J.-F. Aujol. Variational methods for normal integration. *Journal of Mathematical Imaging and Vision*, 60(4):609–632, 2018.

[22] A. Radić, S.M. Lambrick, N.A. von Jeinsen, A.P. Jardine, and D.J. Ward. 3d surface profilometry using neutral helium atoms. *Applied Physics Letters*, 124(20), 2024.

[23] B. Radler, M. Harker, P. O'Leary, and T. Lucyshyn. Non-contact measurement of circular surfaces via photometric stereo in polar coordinates. In *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pages 1–6, 2016.

[24] A. Robles-Kelly and E.R. Hancock. A graph-spectral method for surface height recovery. *Pat. Rec.*, 38:1167–1186, 2005.

[25] P.H. Schönemann. On the formal differentiation of traces and determinants. *Multivariate Behavioral Research*, 20:113–139, 1985.

[26] G.W. Stewart. *Matrix Algorithms*, volume II: Eigensystems. SIAM, Philadelphia, 2001.

[27] C.F. Van Loan. The ubiquitous Kronecker product. *Journal of Computational and Applied Mathematics*, 123:85–100, 2000.

[28] R.J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980.

[29] Z. Wu and L. Li. A line integration based method for depth recovery from surface normals. In *IEEE ICPR*, pages 591–595. IEEE, 1988.

[30] D. Zhu and W.A.P. Smith. Least squares surface reconstruction on arbitrary domains. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 530–545. Springer International Publishing, 2020.

Appendix

The Frobenius norm of a hypermatrix is defined as the square root of sum of the squared values of its entries, as in Equation (8). In this paper, we require the derivative of the Frobenius norm of a hypermatrix (a scalar) with respect to a hypermatrix. The basic notions of the differentiation of a scalar valued function with respect to a matrix can be found in [25]. The basic cost function (13) is composed of terms of the form,

$$\epsilon\left(\mathsf{X}\right) = \left\|\mathsf{R}\left(\mathsf{X}\right)\right\|_{\mathrm{F}}^{2} \;, \tag{40}$$

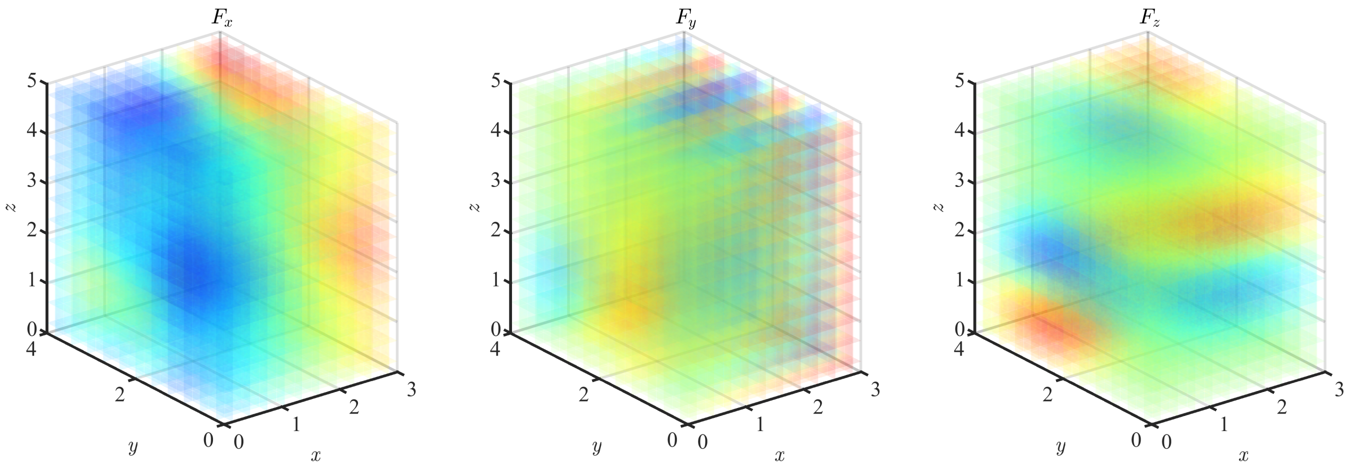

Figure. 2. The x, y, and z components of the synthetic velocity field with added synthetic Gaussian noise with a standard deviation of 5% of the amplitude of the potential function.

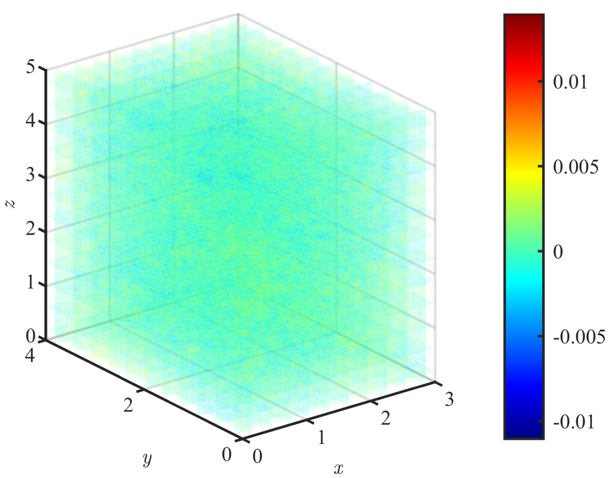

Figure. 3. Residual error hypermatrix, Equation (39).

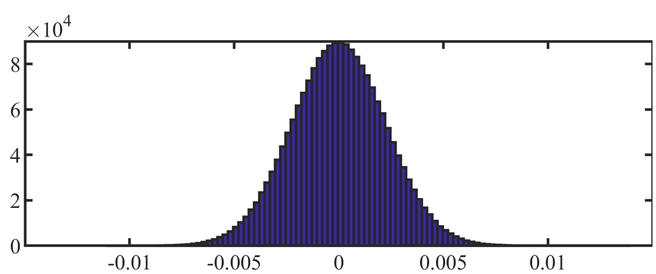

Figure. 4. Histogram of the residual error hypermatrix.

where the residual hypermatrix is the linear relation,

$$\mathsf{R}(\mathsf{X}) = (\mathsf{A}, \mathsf{B}, \mathsf{C}) \cdot \mathsf{X} - \mathsf{F} \,. \tag{41}$$

Writing the cost function in the form,

$$\epsilon(\mathsf{X}) = \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} r_{ijk}(\mathsf{X})^2 \ , \tag{42}$$

and differentiating with respect to the p-q-r entry of the hypermatrix X, we obtain,

$$\frac{\partial \epsilon(\mathsf{X})}{\partial x_{pqr}} = 2 \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} r_{ijk}(\mathsf{X}) \frac{\partial r_{ijk}(\mathsf{X})}{\partial x_{pqr}} \ . \tag{43}$$

From Equation (41), the i-j-k entry of the residual hypermatrix is,

$$\begin{aligned} r_{ijk}(\mathsf{X}) &= \left(\boldsymbol{e}_i^{\mathsf{T}}, \boldsymbol{e}_j^{\mathsf{T}}, \boldsymbol{e}_k^{\mathsf{T}}\right) \cdot \left((\mathsf{A}, \mathsf{B}, \mathsf{C}) \cdot \mathsf{X} - \mathsf{F}\right) \\ &= \left(\boldsymbol{e}_i^{\mathsf{T}}\mathsf{A}, \boldsymbol{e}_j^{\mathsf{T}}\mathsf{B}, \boldsymbol{e}_k^{\mathsf{T}}\mathsf{C}\right) \cdot \mathsf{X} - f_{ijk} \ . \end{aligned} \tag{44}$$

Differentiation with respect to x_{pqr} yields,

$$\begin{aligned} \frac{\partial r_{ijk}(\mathsf{X})}{\partial x_{pqr}} &= \left(\boldsymbol{e}_i^{\mathsf{T}}\mathsf{A}, \boldsymbol{e}_j^{\mathsf{T}}\mathsf{B}, \boldsymbol{e}_k^{\mathsf{T}}\mathsf{C}\right) \cdot \left((\boldsymbol{e}_p, \boldsymbol{e}_q, \boldsymbol{e}_r) \cdot 1\right) \\ &= \left(\boldsymbol{e}_i^{\mathsf{T}}\mathsf{A}\boldsymbol{e}_p, \boldsymbol{e}_j^{\mathsf{T}}\mathsf{B}\boldsymbol{e}_q, \boldsymbol{e}_k^{\mathsf{T}}\mathsf{C}\boldsymbol{e}_r\right) \cdot 1 \\ &= a_{ip} b_{jq} c_{kr} \ . \end{aligned} \tag{45}$$

Substituting this result into the gradient of the cost function, we have,

$$\frac{\partial \epsilon(\mathsf{X})}{\partial x_{pqr}} = 2 \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} a_{ip} b_{jq} c_{kr} \ r_{ijk}(\mathsf{X}) \ . \tag{46}$$

By the definition of matrix-hypermatrix multiplication in Equations (9) and (10), we obtain the gradient of the cost function with respect to the hypermatrix X as,

$$\begin{aligned} \frac{\partial \epsilon(\mathsf{X})}{\partial \mathsf{X}} &= 2\left(\mathsf{A}^{\mathsf{T}}, \mathsf{B}^{\mathsf{T}}, \mathsf{C}^{\mathsf{T}}\right) \cdot \mathsf{R}(\mathsf{X}) \\ &= 2\left(\mathsf{A}^{\mathsf{T}}, \mathsf{B}^{\mathsf{T}}, \mathsf{C}^{\mathsf{T}}\right) \cdot \left((\mathsf{A}, \mathsf{B}, \mathsf{C}) \cdot \mathsf{X} - \mathsf{F}\right) \ . \end{aligned} \tag{47}$$

Expanding, we have equivalently,

$$\frac{\partial \epsilon(\mathsf{X})}{\partial \mathsf{X}} = 2\left(\mathsf{A}^{\mathsf{T}}\mathsf{A}, \mathsf{B}^{\mathsf{T}}\mathsf{B}, \mathsf{C}^{\mathsf{T}}\mathsf{C}\right) \cdot \mathsf{X} - 2\left(\mathsf{A}^{\mathsf{T}}, \mathsf{B}^{\mathsf{T}}, \mathsf{C}^{\mathsf{T}}\right) \cdot \mathsf{F} \ . \tag{48}$$