Fast implication of divergence correction for volumetric PIV data

Chengyue Wang¹, Qi Gao¹, Runjie Wei², Tian Li¹,³, Jinjun Wang¹
¹: Key Laboratory of Fluid Mechanics, Ministry of Education, Beijing University of Aeronautics and Astronautics, Beijing, China
²: MicroVec., Inc, Beijing, China
³: Shenyang Aircraft Design and Research Institute, Shenyang, China
* Correspondent author: qigao@buaa.edu.cn

Keywords: divergence correction, DCS, fast algorithm, weighted DCS

ABSTRACT

Forcing the experimental volumetric velocity fields to satisfy the mass conservation has been proved beneficial for improving the quality of the measurement data. A lot of correction methods including divergence correction scheme (DCS) have been proposed to remove the divergence errors of measurement velocity fields. For tomographic particle image velocimetry (TPIV) data, the measurement uncertainty for the velocity component along the light thickness direction is typically much larger than the other two components. Such biased measurement errors would weaken the performance of traditional correction methods. The paper proposes a variant for existing DCS by adding weight parameters on three velocity components. The weighted DCS (WDCS) has strong advantages on correcting velocity components with different biased noise levels. Furthermore, a fast algorithm for DCS or WDCS is developed, making the correction process significantly low-cost to implement. Numerical test validates the accuracy and computational efficiency for the fast algorithm, and also reveals the advantages of WDCS.

1. Introduction

With the development of flow velocimetry techniques, especially the volumetric particle image velocimetry like tomographic PIV (TPIV) (Elsinga et al. 2006), it is more and more sophisticated to capture all three velocity components over a three-dimensional measurement domain (3D3C velocity field). The 3D3C velocity field enables one to calculate the full velocity gradient tensor (VGT), allowing quantitatively accessing the quality of measurements by checking mass conservation. For incompressible flow, the mass conservation equation reads \( \nabla \cdot \mathbf{u} = 0 \), which means the divergence of velocity field should be zero at each grid. However, noise and intrinsic errors would inevitably cause spurious finite divergence in the velocity field. The remaining divergence residuals not only hinder the studies of the invariants of the VGT (Chong et al. 1990, Ooi et al. 1999), but also cause potential errors in pressure reconstruction based on the measured velocity field. Furthermore, some measurement-based numerical simulations (Sciacchitano et al. 2012, Schneiders et al. 2013) also benefit from a divergence-free correction on
input velocity fields. Therefore, it is an important processing task to remove the divergence residuals or reconstruct new divergence-free velocity fields from the original experimental velocity field.

A number of error-correction methods have been proposed to improve the experimental velocity fields. Azijli and Dwight (2015) classified these methods into three categories according to their working principles: Helmholtz representation, reconstruction with a solenoidal basis, optimization problem. The Helmholtz representation reconstructs the divergence-free field by solving a Poisson equation (Song et al. 1993, Yang et al. 1993, Suzuki et al. 2009, Clark 2012). Boundary conditions should be specified to make the solution unique and an ill-posed problem might occur if the measured velocity field is not globally mass conserving (Azijli and Dwight 2015). Song et al. (1993) employed a 7-points discretization for the Laplacian operator $\nabla^2 = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 + \partial^2 / \partial z^2$. However, the 7-point difference scheme is a low-order approximation, which would lead to residuals when common central difference scheme is employed to estimate the divergence. Reconstructing using divergence-free bases is another way to remove the divergence residuals. Solenoidal radial basis functions (RBF) (Vennell and Beatson 2009, McNally 2011, Busch et al. 2013) and wavelets (Ko et al. 2000, Deriaz and Perrier 2006) have been employed in correcting experimental velocities. Schiavazzi et al. (2014) extended the bases-reconstructing algorithm to unstructured data by using solenoidal atoms and a novel encoding-decoding process. The method is implemented by a sequential matching pursuit (SMP) algorithm. Azijli and Dwight (2015) employed a solenoidal Gaussian process regression (SGPR) to allow a solenoidal smoothing on velocity fields. SGPR could be implemented with a very fast algorithm due to Toeplitz structure of the system matrix. De Silva et al. (2013) proposed a divergence correction scheme (DCS), reconstructing the divergence-free velocity field by solving an optimization problem with a goal function of the distance between the original measurement field and the corrected field, and the constrain of the mass conversation. It is proved that DCS could improve the accuracy of turbulence statistics by very limited change on the measurement data. Wang et al. (2016) proposed a divergence-free smoothing (DFS) method, which is also based on an optimization problem. DFS introduces a smoothing manner into the correction algorithm by combining DCS and the all-in-one method (Garcia 2011), which makes it have a better performance in the post-process of PIV data.

Among all these correction methods, SGPR and DFS intentionally exploit suitable smoothing effect to reduce noise and divergence errors. SMP implicitly introduces some smoothing by the encoding-decoding process. Smoothing treatment could reduce noise and improve robustness by allowing larger correction on the velocity fields, but also lead to the spatial attenuation or resolution reduction. Furthermore, some quantities of a flow are not sensible to noise error. For
example, the averaged velocity and the reconstructed pressure would not be improved by introducing the extra smoothing in correction process (Azijli and Dwight 2015, Wang et al. 2016). Therefore, the least correction without any extra smoothing effect still takes a place in the divergence-correction arsenals, considering their inherent fidelity to the original data. Ideally, the least correction method should reduce homogeneous random noise by \(1 - \sqrt{2/3} \approx 18\%\), since the constrains of mass conversation is \(1/3\) of the total numbers of the velocity components (Schiavazzi et al. 2014). DCS method is a standard least correction method, considering its explicit meaning of goal function. Because that both Poisson equation and DCS are essentially equivalent to the Helmholtz representation theorem (Azijli and Dwight 2015), the Poisson equation could also be viewed as the least correction method. Comparatively, DCS takes its advantage of no boundary condition troubles, but needs to be improved on computing efficiency (Wang et al. 2016).

For TPIV velocity data, the measurement error for the velocity component along the light thickness direction is much larger than the other two components (Elsinga et al. 2006). However, all the existing correction methods deal with the three velocity components equally, which would cause the spread of errors from the worst velocity component to others (de Silva et al. 2013). In some extreme situations, the divergence correction would reduce the error on one velocity component at the price of adding errors on the other components, which is a potential risk for the divergence correction. Improvement on the correction methods is necessary to deal with this biased measurement errors on velocity components.

In this work, a weighted DCS (WDCS) is proposed to control the spread of correction errors caused by the biased uncertainty for the three velocity components. Advantages of WDCS are validated by a numerical assessment. A fast solving algorithm for WDCS or DCS is further developed to make the correction procedure low-cost. The following of this article is structured as follows. Sec. 2 briefly retrospects the DCS method and introduces the WDCS. Sec. 3 gives a fast algorithm for solving DCS and WDCS. In Sec. 4, a numerical test is performed to validate the efficiency of the fast algorithm and the accuracy of WDCS, followed by the conclusion in Sec. 5.

2. DCS and WDCS

A typical TPIV velocity dataset distributes on a \(n \times n \times n \times n\) spatial grid. Indices of \(i, j, k\) indicate the grid node numbers in \(x, y, z\) directions, respectively. The algorithm of DCS reconstructs the divergence-free velocity field by minimizing the distance to the measurement data under the constrain of mass conversation, which can be described as a constrained optimization problem (de Silva et al. 2013) of
\[
\begin{align*}
\min & \sum_{i/j/k=1}^{\text{nx/ny/nz}} (u_{\exp}^{i/j/k} - u_c^{i/j/k})^2 + (v_{\exp}^{i/j/k} - v_c^{i/j/k})^2 + (w_{\exp}^{i/j/k} - w_c^{i/j/k})^2 \\
\text{s.t.} & \sum_{i}=1^{\text{nx}} d_{nx}^{ii'} u_{c}^{i/j/k} + \sum_{j'=1}^{\text{ny}} d_{ny}^{jj'} u_{c}^{i/j/k} + \sum_{k'=1}^{\text{nz}} d_{nz}^{kk'} u_{c}^{i/j/k} = 0 \quad , \\
& \quad (i/j/k = 1,2,..,\text{nx/ny/nz})
\end{align*}
\]  

where \( u^{i/j/k}, v^{i/j/k}, w^{i/j/k} \) denote the three velocity components at the grid of \((i,j,k)\), with the subscripts of \( \exp \) and \( c \) indicating experimental data and corrected data, respectively. \( d_{nx}^{ii'}, d_{ny}^{jj'}, d_{nz}^{kk'} \) are one-dimensional discrete differential operators for \( x, y, z \) directions, which depends on the concrete discrete scheme. For a 2-order central differential scheme with a forward/backward differential substitute at the boundaries, the values of \( d_{nx}^{ii'} \) for \( i=1,2,..,\text{nx} \) and \( i'=1,2,..,\text{nx} \) form a \( \text{nx}\times\text{nx} \) matrix \( d_{nx} \) as:

\[
\begin{bmatrix}
-1 & 1 \\
-1/2 & 0 & 1/2 \\
\vdots \\
-1/2 & 0 & 1/2 \\
-1 & 1
\end{bmatrix}_{\text{nx}} .
\]

Similarly, the values of \( d_{ny}^{jj'}, d_{nz}^{kk'} \) also form the matrices of \( d_{ny}, d_{nz} \) with dimensions of \( \text{ny}\times\text{ny} \) and \( \text{nz}\times\text{nz} \), respectively.

Considering that the measurement error is usually biased on different velocity components for TPIV data, it is wise to add some weight parameters to adjust the fidelity for the three components, like

\[
\begin{align*}
\min & \sum_{i/j/k=1}^{\text{nx/ny/nz}} [\alpha_1 (u_{\exp}^{i/j/k} - u_c^{i/j/k})]^2 + [\alpha_2 (v_{\exp}^{i/j/k} - v_c^{i/j/k})]^2 + [\alpha_3 (w_{\exp}^{i/j/k} - w_c^{i/j/k})]^2 \\
\text{s.t.} & \sum_{i}=1^{\text{nx}} d_{nx}^{ii'} u_{c}^{i/j/k} + \sum_{j'=1}^{\text{ny}} d_{ny}^{jj'} u_{c}^{i/j/k} + \sum_{k'=1}^{\text{nz}} d_{nz}^{kk'} u_{c}^{i/j/k} = 0 \quad , \\
& \quad (i/j/k = 1,1,..,\text{nx/ny/nz})
\end{align*}
\]

where \( \alpha_1, \alpha_2, \alpha_3 \) denote the weight parameters corresponding to three velocity components.

The weight parameters \( \alpha_1, \alpha_2, \alpha_3 \) control the correcting degrees on three velocity components. Smaller parameter component denotes less confidence on the corresponding experimental velocity component and leads to larger correction on it. Therefore, it is natural to specify the weight parameters by a ratio inversely proportional to the uncertainty for corresponding velocity components. The test in Sec. 4 validates that such choice for weighting parameters is optimal. When all the weighting parameters take values of ones, WDCS reduces to DCS. Thus, DCS could be considered as a special case of WDCS in implementation.

3. Fast solving algorithm for WDCS

The optimization problem of Eq. 1 usually contains \( 10-10^2 \) variables, which is very challenging for solving with common optimization methods. De Silva et al. (2013) employed the MATLAB’s
Optimization Toolbox to solve the problem of Eq. 1, which requires expensive computing resources. For a common computer device, it is almost impossible to deal with a typical velocity field of $100 \times 100 \times 20$ by DCS with the MATLAB’s Toolbox, because of its huge memory usage and time consumption.

In this work, a direct and fast solving algorithm is proposed to solve this optimization problem with very high efficiency. Concrete derivation for this algorithm is attached as in Appendix 1. The algorithm begins with the eigenvalue decompositions for $d_{nx}d_{nx}^T$, $d_{ny}d_{ny}^T$, and $d_{nz}d_{nz}^T$, which would not cost much time since all of them are small matrices with dimensions of $\sim 100 \times 100$. The eigenvalue decompositions are formulated as

$$d_{nx}d_{nx}^T = \Phi_{nx} \Lambda_{nx} \Phi_{nx}^T,$$  (4)
$$d_{ny}d_{ny}^T = \Phi_{ny} \Lambda_{ny} \Phi_{ny}^T,$$  (5)
$$d_{nz}d_{nz}^T = \Phi_{nz} \Lambda_{nz} \Phi_{nz}^T.$$  (6)

Denote that $\phi_{nx}^{ij}$, $\phi_{ny}^{ij}$, $\phi_{nz}^{ij}$ are the elements in $i$-th row and $j$-th column $(i = 1, 2, 3; j = 1, 2, 3)$ of $\Phi_{nx}$, $\Phi_{ny}$, $\Phi_{nz}$ and $\Lambda_{nx}^k$, $\Lambda_{ny}^k$, $\Lambda_{nz}^k$ are the $k$-th diagonal elements for $\Lambda_{nx}$, $\Lambda_{ny}$, $\Lambda_{nz}$, respectively. The fast algorithm is then implemented by a series of formulas as

$$S_{\text{exp}}^{i'j'k'} = \sum_{i=1}^{nx} d_{nx}^{ij} u_{\text{exp}}^{ij} + \sum_{j=1}^{ny} d_{ny}^{ij} u_{\text{exp}}^{ij} + \sum_{k=1}^{nz} d_{nz}^{ij} u_{\text{exp}}^{ij},$$  (7)
$$\Gamma^{\text{imm}} = \alpha_1^2 \Lambda_{nx}^1 + \alpha_2^2 \Lambda_{ny}^m + \alpha_3^2 \Lambda_{nz}^n,$$  (8)
$$\mu_{ij}^{jk} = \sum_{l,m,n=1}^{nx/ny/nz} (\Phi_{nx}^{il} \Phi_{ny}^{jm} \Phi_{nz}^{kn}(\Gamma^{\text{imm}})^{-1} \sum_{i'=j'=k'=1}^{nx/ny/nz} \Phi_{nx}^{il'} \Phi_{ny}^{jm'} \Phi_{nz}^{kn'} S_{\text{exp}}^{i'j'k'}).$$  (9)

In these equations, $(i,j,k)$, $(i',j',k')$ and $(l,m,n)$ are all node indices for three coordinating directions from $(1,1,1)$ to $(nx,ny,nz)$. $S_{\text{exp}}^{i'j'k'}$ in Eq. 7 is the divergence residual at node $(i', j', k')$ for the experimental data. It is worthy to note that $\Gamma^{\text{imm}}$ usually has one zero element, which would lead to infinity in the calculation of Eq. 9. To avoid this issue, the zero elements of $\Gamma^{\text{imm}}$ should be replaced by some non-zero values, like one. It could be proved that the values of the replaced elements would not lead to a different result for the corrected velocity field.

Denote $n = nx \times ny \times nz$ as the total numbers of grid nodes. The algorithm has a complex of $O(n(nx + ny + nz))$, which is much smaller than the complex of general conjugated gradient (CG) method of $O(kn^2)$ ($k$ is the numbers of iterations). For the memory consumption, the algorithm is very economical, since the total number of all the procedure variables is in order of $O(n)$. Furthermore, the algorithm is able to be parallelly implemented with easy matrix multiplication in MATLAB, leading to a very high computational efficiency.
4. Numerical Test

In this section, a set of direct numerical simulation (DNS) data of turbulent channel flow (Orszag 1971, Kim et al. 1987, Lee et al. 2013) is used to test the accuracy and computational efficiency of the fast algorithm. Comparison between DCS and WDCS is also made on a noise-reduce test.

The DNS velocity field distributes in a zone of $8\pi\times3\pi H\times2H$, where $H$ is the half-channel height. This computation zone contains $2048\times1536\times512$ spatial grid points in the streamwise, spanwise, and wall-normal directions, respectively. In our test, only some independent small sub-volumes of the DNS field were extracted for the statistical assessment. Detailed information is summarized in Table 2. Following de Silva et al. (2013), a linear interpolation is employed in both streamwise and wall-normal direction to obtain a velocity field on a uniform grid along all three directions, making the DNS data more close to a practical TPIV configuration.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Summary for information about the test DNS sub-volumes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Streamwise grid spacing $\Delta x^+$</td>
<td>6.13</td>
</tr>
<tr>
<td>Spanwise grid spacing $\Delta y^+$</td>
<td>6.13</td>
</tr>
<tr>
<td>Wall-normal grid spacing $\Delta z^+$</td>
<td>6.13</td>
</tr>
<tr>
<td>Wall-normal position of the subzone</td>
<td>399.8-516.3</td>
</tr>
<tr>
<td>Maximum velocity magnitude $</td>
<td>u_{DNS^+}</td>
</tr>
</tbody>
</table>

As pointed out previously, the errors for TPIV velocities are usually biased in three components. In this test, both unbiased and biased noise are assessed. For the unbiased type, noises are equally added to all three velocity components. For the biased type, the noise level of the $w$ velocity component is twice of the other two components ($u$ and $v$). The way of adding two types of noise is further explained by the following equations:

Unbiased noise:

$$
\begin{align*}
    u_n^+ &= u_{DNS^+} + \Gamma \\
    v_n^+ &= v_{DNS^+} + \Gamma \\
    w_n^+ &= w_{DNS^+} + \Gamma
\end{align*}
$$

Biased noise:

$$
\begin{align*}
    u_n^+ &= u_{DNS^+} + \Gamma \\
    v_n^+ &= v_{DNS^+} + \Gamma \\
    w_n^+ &= w_{DNS^+} + 2\Gamma
\end{align*}
$$

$u_{DNS^+}, v_{DNS^+}, w_{DNS^+}$ and $u_n^+, v_n^+, w_n^+$ denote the three original DNS and noisy velocity components, with a superscript ‘+’ indicating they have been normalized by the friction velocity. $\Gamma$ is the Gaussian-distributed noise field, with a mean value of $0$ and a standard variance of $1\%$ of the maximum velocity magnitude $|u_{DNS^+}| = \sqrt{u_{DNS^+}^2 + v_{DNS^+}^2 + w_{DNS^+}^2}$. 
4.1 Validating the fast algorithm

The accuracy of the fast algorithm needs to be tested and validated before applying it in the implementation of DCS and WDCS. 100 small DNS velocity fields with a grid size of 10×10×10 are used in this test. As a comparative reference, the fields are also corrected by solving the optimization problem (Eq. 3) using the ‘fmincon’ function in MATLAB. The exit criterion is based on a maximum constraint violation of $10^{-6}$ and a termination tolerance on the goal function value of $10^{-6}$. Double precision type variables are adopted in the calculation. The velocity fields resulting from two implementation schemes are denoted as $\mathbf{u}_{\text{fast}}$ and $\mathbf{u}_{\text{fmincon}}$, with the subscripts ‘fast’ and ‘fmincon’ indicating the fast algorithm and the ‘fmincon’ function in MATLAB. A comparison between the results of two implementation ways is provided in Table 2. It shows that both could reduce the divergence error below $10^{-15}$. The maximum deviations for all three components are $\sim 10^{-6}$, which is a good validation for the accuracy of the fast solving algorithm.

The computational efficiency for the fast algorithm is also tested. The test was performed on a computer with quad-core i7 CPU of 3.7GHz and 32 GB RAM. The time costs of WDCS to correct three volumes are displayed in Table 3. For a typical TPIV data, the processing time is in seconds. For a huge field of 500×500×50, the time cost is less than 5 seconds. The high efficiency of this fast algorithm makes DCS or WDCS very competitive among the divergence-correction methods.
4.2 Comparing between DCS and WDCS

In this test, a comparison between the accuracy of DCS and WDCS is made based on the processed results of 100 DNS fields (100×100×20) with biased and unbiased noise. Both DCS and WDCS are implemented with the fast algorithm. To quantitatively compare DCS and WDCS, an error reduction percentage is used to assess the accuracy of two methods, which is defined as the difference between the error of noisy field and the error of corrected field, normalized by the error of noisy field as formulated by Eq. 13.

\[
\begin{align*}
    p_u &= \frac{\|u_n^+-u_{D\text{NS}}^+\| - \|u_c^+-u_{D\text{NS}}^+\|}{\|u_n^+-u_{D\text{NS}}^+\|} \times 100\% \\
    p_v &= \frac{\|v_n^+-v_{D\text{NS}}^+\| - \|v_c^+-v_{D\text{NS}}^+\|}{\|v_n^+-v_{D\text{NS}}^+\|} \times 100\% \\
    p_w &= \frac{\|w_n^+-w_{D\text{NS}}^+\| - \|w_c^+-w_{D\text{NS}}^+\|}{\|w_n^+-w_{D\text{NS}}^+\|} \times 100\% 
\end{align*}
\]

(13)

where \(u, v, w\) are three velocity components and the subscripts DNS, \(n, c\) indicate the original DNS field, the noisy field and the corrected field. The symbol \(\| \|\) means a norm operation on the measurement field, which is calculated by rooting the mean squares of the all the velocity values on the one DNS field.

| Table 4 Error reduction percentages for DCS and WDCS (%) |
|---------------|----------|---|-----|
| Noise type    | Method   | u | v   | w   |
| Unbiased      | DCS      | 17.68 | 17.70 | 19.57 |
| Biased        | WDCS     | 10.85 | 10.84 | 35.84 |
|               |          | 3.76  | 3.79  | 27.75 |

The error reduction percentages of DCS and WDCS have been provided in Table 4. The data are based on the statistical average on the testing results of 100 DNS fields. It shows that for the unbiased noise, the error reduction percentages are also unbiased, about 18% for each component, which is consistent with the value given in the introduction. For the biased noise, the percentage for \(w\) component increased while the ones for other two components drop significantly. The WDCS improves the error reduction percentages for all the velocity components by about 7 percentages, which is a promising result for WDCS.

In practical situations, the noise levels or uncertainties for the velocity components are unknown. Estimation on the uncertainties is needed to choose the corresponding weight parameters, which might cause some deviation from the optimal values. A further test on the robustness of the weight parameters is necessary. In this test, we employed a series of parameters \(\alpha = [1, 1, \alpha_3]\) with \(\alpha_3\) continuously changed from 0.1 to 1 by a step of 0.1. The corresponding
error reduction percentages are displayed in Fig. 2. The figure shows that the optimal $\alpha_3$ is consistent with our former speculation, which leads to the largest error-reduce percentages on the three velocity components simultaneously. When $\alpha_3$ reduces to 0.2, $p_w$ drops significantly below the result for DCS ($\alpha_3 = 1$). Negative percentage for $p_w$ occurs when $\alpha_3$ is less than 0.1, which means that the correction increases the error of $w$ rather than reduces it. For a wide range of $\alpha_3$, WDCS takes its advantages for all the three components. A deviation of 0.1 from the optimal $\alpha_3$ causes very limited loss on the best percentage of noise reduction for WDCS, which allows 15%~20% deviation on estimating the uncertainty of $w$ component.

![Fig. 1 Error reduction percentages for WDCS with different weight parameters](image)

5 Conclusions

The current work proposes a weighted DCS to deal with the biased uncertainty on TPIV velocity fields. The weighting parameters are proposed to be inversely proportional to the uncertainties for three velocity components. A fast algorithm for DCS and WDCS is proposed. Typically, the algorithm finishes the correction in seconds, which makes WDCS and DCS very efficient to be implemented. WDCS improves the accuracy of DCS in reducing biased noise. Numerical test validates the effectiveness of WDCS. Based on these results, the author suggests that the WDCS should become a routine procedure for TPIV data process.
6. Acknowledgements

This work is supported by the National Natural Science Foundation of China (11472030, 11327202, 11490552) and the Fundamental Research Funds for Central Universities (YWF-16-JCTD-A-05).

Appendix 1

The determining problem for weighted DCS shown by Eq. 2 could be rewritten as the matrix form:

\[
\begin{aligned}
\min (U_{\text{exp}} - U_c)^T \alpha \gamma (U_{\text{exp}} - U_c), \\
\text{s.t. } A U_c = 0 \\
\end{aligned}
\]  

where \( U_{\text{exp}} \) and \( U_c \) are column matrices collecting all the three components of velocities from all the spatial positions, with subscripts of ‘exp’ and ‘c’ denoting the experimental data and corrected field. \( \alpha \) is the corresponding weighting matrix and \( A \) is the divergence operator. The specific forms for \( \alpha \) and \( A \) are dependent on the arrangement order of velocity elements in \( U_{\text{exp}} \) and \( U_c \). In this work, the velocity elements are arranged by the following order: the \( l \)-th velocity component element in the spatial position \((i, j, k)\) is arranged at \((i + (j - 1) \times nx + (k - 1) \times nx \times ny + (l - 1) \times nx \times ny \times nz)\) - th position in the vector matrix \( U_{\text{exp}} \) or \( U_c \). In this way,

\[
\alpha = \text{diag}(\alpha_1, I_{nx}, \alpha_2, I_{ny}, \alpha_3, I_{nz}).
\]

with \( I_{nx}, I_{ny}, I_{nz} \) denoting identify matrices with the dimensions of \( nx, ny, nz \), respectively. \( \otimes \) denotes the Kronecker tensor product between two matrices. \( d_{nx}, d_{ny}, d_{nz} \) are derivative operators with dimensions of \( nx, ny, nz \), respectively, which has been introduced in the article.

By introducing a Lagrange multiplier \( \lambda \), the constrained minimization problem (Eq. 14) could be transformed as the solution of a set of linear equations as:

\[
\begin{aligned}
AU_c = 0, \\
A^T \lambda + \alpha^2 U_c = \alpha^2 U_{\text{exp}}. \\
\end{aligned}
\]

Solving \( U_c \) in the second equation and replacing it in the first equation, a new set of equations are obtained as

\[
\begin{aligned}
A \alpha^{-2}A^T \lambda = A U_{\text{exp}}, \\
U_c = U_{\text{exp}} - \alpha^{-2}A^T \lambda. \\
\end{aligned}
\]

The critical procedure to solve these equations is to derive the inverse of \( A \alpha^{-2}A^T \). The specific form of \( A \alpha^{-2}A^T \) is

\[
\begin{aligned}
A \alpha^{-2}A^T &= \alpha_1^{-2} I_{nx} \otimes d_{nx} d_{nx}^T + \alpha_2^{-2} I_{ny} \otimes d_{ny} d_{ny}^T + \alpha_3^{-2} I_{nz} \otimes d_{nz} d_{nz}^T, \\
\end{aligned}
\]

Considering \( d_{nx} d_{nx}^T, d_{ny} d_{ny}^T, d_{nz} d_{nz}^T \) are all real symmetric matrices, they could be decomposed as

\[
\begin{aligned}
d_{nx} d_{nx}^T &= \Phi_{nx} \Lambda_{nx} \Phi_{nx}^T, \\
d_{ny} d_{ny}^T &= \Phi_{ny} \Lambda_{ny} \Phi_{ny}^T, \\
d_{nz} d_{nz}^T &= \Phi_{nz} \Lambda_{nz} \Phi_{nz}^T, \\
\end{aligned}
\]

where \( \Phi_{nx}, \Phi_{ny}, \Phi_{nz} \) and \( \Lambda_{nx}, \Lambda_{ny}, \Lambda_{nz} \) are the eigenvalue and eigenvector matrices of \( d_{nx} d_{nx}^T, d_{ny} d_{ny}^T, d_{nz} d_{nz}^T \). \( \Phi_{nx}, \Phi_{ny}, \Phi_{nz} \) are all orthogonal matrices, which would significantly simplify the complex inverse operation.

Employing the decomposition results of \( d_{nx} d_{nx}^T, d_{ny} d_{ny}^T, d_{nz} d_{nz}^T \), and considering the operation properties for the Kronecker tensor product, \( A \alpha^{-2}A^T \) could be decomposed as

\[
A \alpha^{-2}A^T = (\Phi_{nx} \otimes \Phi_{ny} \otimes \Phi_{nz}) \Gamma (\Phi_{nx} \otimes \Phi_{ny} \otimes \Phi_{nz})^T, 
\]

where

\[
\begin{aligned}
\Gamma &= \alpha_1^{-2} I_{nx} \otimes I_{ny} \otimes \Lambda_{nx} + \alpha_2^{-2} I_{nx} \otimes I_{ny} \otimes \Lambda_{ny} + \alpha_3^{-2} I_{nx} \otimes I_{ny} \otimes \Lambda_{nz}.
\end{aligned}
\]

The matrix \( \Phi_{nx} \otimes \Phi_{ny} \otimes \Phi_{nz} \) is an orthogonal matrix, which facilitates the inverse operation by a simple transposing.

Therefore, the Lagrange multiplier could be derived by an explicit function, as

\[
\lambda = (\Phi_{nx} \otimes \Phi_{ny} \otimes \Phi_{nx}) \Gamma^{-1} (\Phi_{nx} \otimes \Phi_{ny} \otimes \Phi_{nx})^T A U_{\text{exp}}.
\]

The final correction velocity \( U_c \) could be determined by the second equation in Eq. 18.

For more application about the operation of Kronecker tensor product and similar decomposition work, the reader should refer to Golub et al. (1998).
Reference


