Matrix Operations for the Simulation and Immediate Reverse-Engineering of Time Series Data

Michael A. Idowu
SIMBIOS Centre
School of Contemporary Sciences
University of Abertay
Dundee, Scotland, UK
m.idowu@abertay.ac.uk

James L. Bown
SIMBIOS Centre
Institute of Arts, Media and Computer Games
University of Abertay
Dundee, Scotland, UK
j.bown@abertay.ac.uk

***Abstract*—We present a new method for constructing and decomposing square matrices. This method, based on the computed parameterisation of their implied determinants and minors, operates on the product of factors of a new form of matrix decomposition. This method may be employed to build new matrices with fixed determinant(s). We demonstrate that this new approach is fundamentally well-connected to the Cholesky decomposition if applied on symmetric matrices. We also demonstrate that it is related to the LU decomposition method via a diagonal matrix multiplier. Also through this new method a direct relation between Cholesky decomposition and LU factorisation is shown. This method, presented for the first time, is useful for (re)constructing matrices with a predefined determinant and simulating inverse problems. The inference method introduced here also is based on new matrix manipulation techniques that we have developed for the identification of systems from reproducible time series data.**

***Keywords*—mathematical modelling; simulation; optimisation algorithms; network inference; inverse problems; matrix algebra and calculus; time series analysis; system identification and parameter estimation methods**

I. Introduction

In systems biology, where theoretical models are important aids in interpreting complex systems dynamics, a robust framework that is inexpensive and able to simplify the creation and evaluation of system identification and parameter estimation problems and solutions is valuable. The framework we have developed is matrix-based and sophisticated enough for the identification of ODE models from time series data. We demonstrate that through simple matrix manipulation techniques, powerful and effective computational tools, complementary to existing reverse engineering and modelling packages, may be developed. These techniques are useful for understanding complex network structures and dynamics.We present a new method for constructing and decomposing square matrices. This method, based on the computed parameterisation of their implied determinants and minors, operates on the product of factors of a new form of matrix decomposition and may be employed to build new matrices with fixed determinant(s).

Suppose the square matrix A is to be partitioned into LDU decomposition factors, i.e., A is to be transformed into equivalent lower, diagonal, and upper matrix factors, then one may employ the Gaussian elimination method [1]. Importantly, here we show that A may be decomposed and recomposed in terms of its implied determinant and minors. Given A may be factorised in terms of its minor and determinants, we examine the determinant of A we suggest that if the entries of the LDU factors of A are modified such that the determinant value remains fixed, then any matrix that is reconstructed as a product of those modified LDU factors will have its determinant equal to that fixed value. Thus we propose that by this new method multiple nonsingular matrices with a predefined determinant may be created. This algorithm, presented here for the first time, provides a robust method for constructing nonsingular or singular matrix with a predefined determinant and is applicable to matrices of different sizes. This method provides us with a new tool for decomposing square matrices. We demonstrate that for symmetric matrices our new LDU decomposition method is related to the Cholesky decomposition method [1] and to the LU decomposition method [1].

A. A new matrix decomposition and composition method

It was Householder [2] who first hinted that when an LDU factorization exists and is unique there might be a closed (explicit) formula for the elements of the L, D, and U factors in terms of the ratios of the determinants of certain submatrices of the original matrix A. However, Householder did not explain how to determine this.

Matrix decomposition is used in matrix algebra to solve systems of linear equations. An LDU factorisation of a matrix, A, is basically a decomposition of the matrix to the form $A = L.D.U$ where L and U are lower- and upper-unit triangular matrices, respectively, and D is a diagonal matrix. To develop a method for constructing matrices with a predefined determinant, we discovered that an optimised variant of the LDU technique is necessary. This decomposition technique, initially viewed as an optimised reverse engineering method of matrix composition, may be viewed from the perspective that the product of $L_d.D_d.U_d$ factors is also useful for generating reproducible time series data. Expressed in terms of the parameterisation of its implied minors and determinant, it can be used to create nonsingular jacobian matrices. As demonstrated later, the L_d and U_d factors are triangular matrices but not necessarily unit-trianglular. From the definition $A = L_d.D_d.U_d$, reading from LHS to RHS, we may view the decomposition process as transforming the matrix A to $L_d.D_d.U_d$, while on the other hand, interpreting from RHS to LHS, a composition method of creating a matrix with known properties is effected.

1) Definitions: If A is an *n* x *n* matrix, and *i* and *j* are positive integers less than or equal to *n*, then an *i* x *j* minor (often denoted $M_{i,j}$) of A is the determinant of the (*n-i*) x (*n-i*) matrix obtained from A by deleting from A its i^{th} row and j^{th} column. This means that the minor $M_{3,2}$ of a 3 x 3 matrix A may be alternatively represented as $det_{r_{1,2}c_{1,3}}$, that is, $M_{3,2} = det_{r_{1,2}c_{1,3}}$.

Let L_d be the lower triangular matrix component of the $L_d.D_d.U_d$ factorisation of A. L_d has only zero-values above its diagonal; all the entries in the first column of L_d are the same as entries in the first column of A and all other non-zero entries are either determinants of submatrices formed with $A_{1,1}$ or minors of A. Each diagonal entry of L_d at position (i) is the determinant of the upper left *i*-by-*i* submatrix of A represented as det_i [1], where i = 1, 2, ... , length(A);

$$L_d = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{2,1} & det_2 & 0 \\ a_{3,1} & det_{r_{1,3}c_{1,2}} & det_3 \end{bmatrix}, \tag{1}$$

Let D_d be a diagonal matrix that is the reciprocal of the product of the determinant of the submatrix formed with $A_{1,1}$ at the i^{th}-position and that of submatrix formed with $A_{1,1}$ from the $(i-1)^{th}$ point above the diagonal if it exists such that

$$D_d = \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 \\ 0 & \frac{1}{(a_{1,1}*det_2)} & 0 \\ 0 & 0 & \frac{1}{(det_2*det_3)} \end{bmatrix} \tag{2}$$

Let U_d be an upper triangular matrix with all entries below the diagonal set to zero, all entries in the first row of U_d are equivalent to entries in first row of A; all other entries in U_d are minors of the sub-blocks formed with $A_{1,1}$, such that U_d is of the form:

$$U_d = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & det_2 & det_{r_{1,2}c_{1,3}} \\ 0 & 0 & det_3 \end{bmatrix} \tag{3}$$

We wish to show that

$$A_3 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = L_d * D_d * U_d \tag{4}$$

, and that a generalised method for constructing matrices with fixed determinant exists.

2) Representation of matrix entries by minors: We claim that entries of matrices may be representated by algebraic sums of products of their minors. For example, let A_3 be a 3 x 3 square matrix (with 3x3=9 parameters) such that

$$A_3 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & \frac{(det_2+a_{1,2}*a_{2,1})}{a_{1,1}} & \frac{(det_{r_{1,2}c_{1,3}}+a_{1,3}*a_{2,1})}{a_{1,1}} \\ a_{3,1} & \frac{(det_{r_{1,3}c_{1,2}}+a_{1,2}*a_{3,1})}{a_{1,1}} & \frac{r_{3,3}}{(a_{1,1}*det_2)} \end{bmatrix} \tag{5}$$

where
$r_{3,3} = (a_{1,1} * det_3 + det_{r_{1,2}c_{1,3}} * det_{r_{1,3}c_{1,2}} + a_{1,3} * a_{3,1} * det_2)$.

In the same way, A_4, a 4 x 4 square matrix (with 16 parameters), may be derived. In generalising the algorithm to create square matrices of sizes ≥ 4, only a slight modification is required. We illustrate how to obtain this generalisation by creating A_4 as an example.

3) Construction of 4x4 matrices: It is important to show how the $L_d.D_d.U_d$ method may be applied also to the composition or decomposition of a 4x4 matrix. A_4 is derived as a product of $L_d * D_d * U_d$ factors:

$$A_4 = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \\ a_{4,1} & a_{4,2} & a_{4,3} & a_{4,4} \end{bmatrix} = L_d * D_d * U_d$$

where $L_d = \begin{bmatrix} a_{1,1} & 0 & 0 & 0 \\ a_{2,1} & det_2 & 0 & 0 \\ a_{3,1} & det_{r_{1,3}c_{1,2}} & det_3 & 0 \\ a_{4,1} & det_{r_{1,4}c_{1,2}} & det_{r_{1,2,4}c_{1,2,3}} & det_4 \end{bmatrix}$,

$D_d = \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 & 0 \\ 0 & \frac{1}{(a_{1,1}*det_2)} & 0 & 0 \\ 0 & 0 & \frac{1}{(det_2*det_2)} & 0 \\ 0 & 0 & 0 & \frac{1}{(det_3*det_4)} \end{bmatrix}$ and

$U_d = \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} & a_{4,1} \\ 0 & det_2 & det_{r_{1,2}c_{1,3}} & det_{r_{1,2}c_{1,4}} \\ 0 & 0 & det_3 & det_{r_{1,2,3}c_{1,2,4}} \\ 0 & 0 & 0 & det_4 \end{bmatrix}$.

It is not surprising seeing that exactly 4x4 = 16 distinct parameters are required to construct a 4x4 matrix with a predefined determinant we choose.

II. Matrix construction with fixed determinant(s)

In the simulation (artificial creation) of inverse problems for the development and optimisation of inference methods, it is important to ensure that the target jacobian matrix to be inferred is well-conditioned, i.e. it is nonsingular. Nonsingularity may be eliminated by ensuring that the determinant of the matrix is not zero (and not close to it). One of the primary reasons for using nonsingular matrix models in simulating time series data is to ensure that the orginal jacobian matrix being used is reproducible and can be inferred. Only nonsingular matrices may guarantee that the uniqueness of a potential solution during the inference process is not lost.

A. Example #1: matrix composition

To construct a matrix with determinant equal to -8, first establish the product of L_d, D_d and U_d:

[1] $det*label*$ parameter is a determinant or minor of A.

$$L_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & -4 & 0 & 0 \\ 9 & -8 & -4 & 0 \\ 13 & -12 & 0 & -8 \end{bmatrix},$$

$$D_d = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 \\ 0 & 0 & 0 & \frac{1}{32} \end{bmatrix},$$

$$U_d = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -4 & -8 & -12 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -8 \end{bmatrix}$$

The matrix composition method $A_{4_a} = L_d.D_d.U_d$ gives the result:

$$A_{4_a} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 12 & 12 \\ 13 & 14 & 15 & 18 \end{bmatrix}$$

B. Example #2: matrix composition with fewer parameters

To create a matrix with determinant of -8 from fewer nonzero parameters, the requirement is that the parameter det_4 must be set to -8 and that none of the diagonal entries of D_d is zero.

$$A_{4_b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & -1 & 0 & 0 \\ 6 & 0 & \frac{1}{2} & 0 \\ 7 & 0 & 0 & -8 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & -\frac{1}{4} \end{bmatrix}$$
$$* \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & -8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 9 & 15 & 20 \\ 6 & 12 & \frac{35}{2} & 24 \\ 7 & 14 & 21 & 12 \end{bmatrix}$$

C. Example #3: symmetric matrix composition

A nonsingular symmetric 4x4 matrix may be constructed by replacing U_d with the transpose of L_d.

$$A_{4s_1} = L_d * D_d * L_d^T \tag{6}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 5 & -4 & 0 & 0 \\ 9 & -8 & -4 & 0 \\ 13 & -12 & 0 & -8 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 \\ 0 & 0 & 0 & \frac{1}{32} \end{bmatrix}$$
$$* \begin{bmatrix} 1 & 5 & 9 & 13 \\ 0 & -4 & -8 & -12 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -8 \end{bmatrix} = \begin{bmatrix} 1 & 5 & 9 & 13 \\ 5 & 21 & 37 & 53 \\ 9 & 37 & 66 & 93 \\ 13 & 53 & 93 & 135 \end{bmatrix}$$

D. Example #4: variant symmetric matrix composition

Alternatively, a nonsingular symmetric 4x4 matrix may be constructed by replacing L_d with the transpose of U_d.

$$A_{4s_2} = U_d^T * D_d * U_d \tag{7}$$

$$A_{4s_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & -4 & 0 & 0 \\ 3 & -8 & -4 & 0 \\ 4 & -12 & 0 & -8 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -\frac{1}{4} & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 \\ 0 & 0 & 0 & \frac{1}{32} \end{bmatrix}$$
$$* \begin{bmatrix} 1 & 2 & 3 & 4 \\ 0 & -4 & -8 & -12 \\ 0 & 0 & -4 & 0 \\ 0 & 0 & 0 & -8 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 0 & -2 & -4 \\ 3 & -2 & -6 & -12 \\ 4 & -4 & -12 & -18 \end{bmatrix}$$

III. Multiple matrices with a predefined (fixed) determinant

We have demonstrated (using 4 examples) how multiple 4x4 matrices with a fixed determinant of -8 may be constructed using our $L_d.D_d.U_d$ method and showed in the previous section how the following matrices all with the same determinant could be easily created. Specifically A_{4_a}, A_{4_b}, A_{4s_1}, and A_{4s_2} all have equal determinants, i.e. $det(A_{4_a}) = det(A_{4_b}) = det(A_{4s_1}) = det(A_{4s_2}) = -8$. The method, shown here for 4x4 matrices, operates independent of matrix size.

IV. Decomposition of matrices

A. $L_d.D_d.U_d$ Decomposition of a Symmetric Matrix

If a square matrix is equal to its transpose then such matrix is said to be symmetric. If M_s is symmetric, then $M_s = M_s^T$ and is regarded to be of the form

$$M_s = \begin{bmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{1,2} & m_{2,2} & m_{2,3} \\ m_{1,3} & m_{2,3} & m_{3,3} \end{bmatrix} \tag{8}$$

As shown earlier, the $L_d.D_d.U_d$ decomposition of symmetric matrices, e.g. M_s, has the following properties: $L_d^T = U_d$ and $Ld = U_d^T$. This means that the $L_d.D_d.U_d$ decomposition of

$$M_s = L_d.D_d.U_d = U_d^T.D_d.U_d = L_d.D_d.L_d^T \tag{9}$$

B. Relation between Cholesky decomposition and our $L_d.D_d.U_d$ decomposition

The Cholesky decomposition of a symmetric, positive-definite matrix M_s is a factorisation of M_s into $L_c.L_c^*$ where L_c is a lower triangular matrix with positive diagonal entries, and L_c^* is the conjugate transpose of L_c.

From the last definition $M_s = L_d.D_d.U_d = U_d^T.D_d.U_d = L_d.D_d.L_d^T$ we can derive the Cholesky decomposition factors as follows:

$$M_s = L_d * D_d^{1/2} * D_d^{1/2} * L_d^T$$

Since D_d is a diagonal matrix, therefore

$$M_s = L_d * D_d^{1/2} * (D_d^{1/2})^T * L_d^T$$

$$M_s = (L_d * D_d^{1/2}) * (L_d * D_d^{1/2})^T = L_c.L_c^*$$

$$M_s = (U_d * D_d^{1/2})^T * (U_d * D_d^{1/2}) = L_c.L_c^*$$

where $L_c = (L_d * D_d^{1/2}) = (U_d * D_d^{1/2})^T$ for any square matrix M_s. This establishes the relation between our $L_d.D_d.U_d$ decomposition and the Cholesky's.

C. Relation between LU factorisation and our $L_d.D_d.U_d$ decomposition

Our analysis shows that by solving the system Ax=b with Gaussian elimination, a lower triangular matrix of the form

$$L = \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{1,2}}{m_{1,1}} & 1 & 0 \\ \frac{a_{1,3}}{m_{1,1}} & \frac{det_{r_{1,3}c_{1,2}}}{det_2} & 1 \end{bmatrix}$$

results, which has the same outcome as

$$L = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{1,2} & det_2 & 0 \\ a_{1,3} & det_{r_{1,3}c_{1,2}} & det_3 \end{bmatrix} \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 \\ 0 & \frac{1}{det_2} & 0 \\ 0 & 0 & \frac{1}{det_3} \end{bmatrix}$$

$$= L_d * \begin{bmatrix} \frac{1}{m_{1,1}} & 0 & 0 \\ 0 & \frac{1}{det_2} & 0 \\ 0 & 0 & \frac{1}{det_3} \end{bmatrix}$$

using our convention. Since

$$L_d = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{1,2} & det_2 & 0 \\ a_{1,3} & det_{r_{1,3}c_{1,2}} & det_3 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{1,2}}{a_{1,1}} & 1 & 0 \\ \frac{a_{1,3}}{a_{1,1}} & \frac{det_{r_{1,3}c_{1,2}}}{det_2} & 1 \end{bmatrix} \begin{bmatrix} a_{1,1} & 0 & 0 \\ 0 & det_2 & 0 \\ 0 & 0 & det_3 \end{bmatrix} = L{*}D_m \tag{10}$$

where $D_m = \begin{bmatrix} a_{1,1} & 0 & 0 \\ 0 & det_2 & 0 \\ 0 & 0 & det_3 \end{bmatrix}$ is the diagonal matrix containing only the principal minors of the target matrix. Therefore we establish the relation between LU factorisation and our $L_d.D_d.U_d$ decomposition method through the following derivation:

Let $L * D_m = L_d$ where D_m is the right matrix multiplier that transforms the L factor of the well-known LU decomposition to our L_d factor.

Because both the L_d and L factors are derived from the same Gaussian elimination process, it turns out that D_m is a diagonal matrix. Therefore, $L = L_d.D_m^{-1}$.

Since $D_m^{-1}.D_m = I$ where D_m is any square matrix, D_m^{-1} its implied inverse and I is the identity.

$$L_d.D_d.U_d = L_d.I.D_d.U_d$$

$$L_d.D_d.U_d = L_d.(D_m^{-1}.D_m).D_d.U_d$$

$$L_d.D_d.U_d = (L_d.D_m^{-1}).(D_m.D_d.U_d)$$

Since $L = (L_d.D_m^{-1})$, therefore

$$L_d.D_d.U_d = (L).(D_m.D_d.U_d)$$

And since we want to preserve the integrity of the matrix A in both our decomposition and the LU decomposition processes, the following property holds:

$$A = L_d * D_d * U_d = LU$$

Therefore, LU factorisation itself may be seen to have a new interpretation in terms of our decomposition method, i.e.

$$LU = (L_d.D_m^{-1}) * (D_m.D_d.U_d)$$

$$L = (L_d.D_m^{-1})$$

$$U = (D_m.D_d.U_d)$$

where the matrix D_m is derived to be

$$D_m = \begin{bmatrix} a_{1,1} & 0 & 0 \\ 0 & det_2 & 0 \\ 0 & 0 & det_3 \end{bmatrix}$$

$$L{*}U = \begin{bmatrix} 1 & 0 & 0 \\ \frac{a_{1,2}}{m_{1,1}} & 1 & 0 \\ \frac{a_{1,3}}{m_{1,1}} & \frac{det_{r_{1,3}c_{1,2}}}{det_2} & 1 \end{bmatrix} * \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & \frac{det_2}{a_{1,1}} & \frac{det_{r_{1,2}c_{1,3}}}{a_{1,1}} \\ 0 & 0 & \frac{det_3}{det_2} \end{bmatrix} \tag{11}$$

where L is lower triangular and U is upper triangular as expected.

D. Other variants of our $L_d.D_d.U_d$ decomposition method

Other variants of $L_d.D_d.U_d$ decomposition factors may exist in terms of

$$A = L_v.D_v.U_v = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{2,1} & 1 & 0 \\ a_{3,1} & \frac{det_{r_{1,3}c_{1,2}}}{det_2} & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 \\ 0 & \frac{det_2}{a_{1,1}} & 0 \\ 0 & 0 & \frac{det_3}{det_2} \end{bmatrix} * \begin{bmatrix} a_{1,1} & a_{,1,2} & a_{1,3} \\ 0 & 1 & \frac{det_{r_{1,2}c_{1,3}}}{det_2} \\ 0 & 0 & 1 \end{bmatrix}$$

which has a symmetric matrix equivalent as

$$As = {U_{vs}}^T.D_{vs}.U_{vs} = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{1,2} & 1 & 0 \\ a_{1,3} & \frac{det_{r_{1,2}c_{1,3}}}{det_2} & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 \\ 0 & \frac{det_2}{a_{1,1}} & 0 \\ 0 & 0 & \frac{det_3}{det_2} \end{bmatrix} * \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ 0 & 1 & \frac{det_{r_{1,2}c_{1,3}}}{det_2} \\ 0 & 0 & 1 \end{bmatrix}$$

or

$$As = L_{vs}.D_{vs}.{L_{vs}}^T = \begin{bmatrix} a_{1,1} & 0 & 0 \\ a_{2,1} & 1 & 0 \\ a_{3,1} & \frac{det_{r_{1,3}c_{1,2}}}{det_2} & 1 \end{bmatrix} * \begin{bmatrix} \frac{1}{a_{1,1}} & 0 & 0 \\ 0 & \frac{det_2}{a_{1,1}} & 0 \\ 0 & 0 & \frac{det_3}{det_2} \end{bmatrix} * \begin{bmatrix} a_{1,1} & a_{2,1} & a_{3,1} \\ 0 & 1 & \frac{det_{r_{1,3}c_{1,2}}}{det_2} \\ 0 & 0 & 1 \end{bmatrix}$$

Note, these may not be the most reduced forms of factorisation and may require further optimisation of entries.

V. Applications of $L_d.D_d.U_d$ method to systems of linear equations

A. Solving systems of linear systems

The $L_d.D_d.U_d$ decomposition can be applied to solve a system of linear equations such as $A.x = b$ by first computing the $L_d.D_d.U_d$ decomposition of A as $A = (L_d * D_d^{1/2}) * (L_d * D_d^{1/2})^T$ if A is symmetric and positive definite. For example, finding x_u (below) the decomposition can be applied to solving the equation $(L_d{*}D_d^{1/2}){*}x_u = b$ where $x_u = (L_d{*}D_d^{1/2})^T{*}x$. This means that the $L_d.D_d.U_d$ decomposition method may be applied in solving

$$A.x = b$$

problems without having to compute the actual inverse of A.

B. Application to time series inverse problem analysis

In a system of linear differential equations an inverse problem may be defined in the form

$$\dot{X} = A * X$$

where $\dot{X}$ and X are known vectors of same length, and A is the unknown matrix that must be identified. Note that there is difference between a general system of n linear differential equations with unknown (jacobian) matrix parameters and a general system of n linear equations with unknown vector parameters. The latter is much simpler to solve as explained above. However, the formulation of the inverse problem remains the same in structure. Its algebraic representation is as follows:

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \vdots \\ \dot{X}_n \end{bmatrix} = \begin{bmatrix} \frac{\partial X_1}{\partial X_1}X_1 + & \frac{\partial X_1}{\partial X_2}X_2 \ldots + & \frac{\partial X_1}{\partial X_m}X_m \\ \frac{\partial X_2}{\partial X_1}X_1 + & \frac{\partial X_2}{\partial X_2}X_2 \ldots + & \frac{\partial X_2}{\partial X_m}X_m \\ \vdots & \vdots & \vdots \\ \frac{\partial X_m}{\partial X_1}X_1 + & \frac{\partial X_m}{\partial X_2}X_2 \ldots + & \frac{\partial X_m}{\partial X_m}X_m \end{bmatrix} \tag{12}$$

where the partial derivative parameters are the entries of the jacobian matrix A; A contains the relative rates of change with respect to the dependent variables [1]. In light of this definition, a time series inverse problem may be defined (in a mathematical sense) as a general system of m linear differential equations with an unknown m x m jacobian matrix A may be rewritten in the form $\dot{X} = A.X$, which has the matrix equation form

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \vdots \\ \dot{X}_m \end{bmatrix} = \begin{bmatrix} \frac{\partial X_1}{\partial X_1} & \frac{\partial X_1}{\partial X_2} \cdots & \frac{\partial X_1}{\partial X_m} \\ \frac{\partial X_2}{\partial X_1} & \frac{\partial X_2}{\partial X_2} \cdots & \frac{\partial X_2}{\partial X_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial X_m}{\partial X_1} & \frac{\partial X_m}{\partial X_2} \cdots & \frac{\partial X_m}{\partial X_m} \end{bmatrix} . \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} \tag{13}$$

where $\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{bmatrix} = X(_t)$ is a known state vector, i. e. the t^{th} vector of the given time series X. Rewritten in a multi-state definition, if m number of (state) measurements are taken after the initial condition, it becomes $\begin{bmatrix} \dot{X}_{1_0}\dot{X}_{1_1}\ldots\dot{X}_{1_{n-1}} \\ \dot{X}_{2_0}\dot{X}_{2_1}\ldots\dot{X}_{2_{n-1}} \\ \vdots \\ \dot{X}_{m_0}\dot{X}_{m_1}\ldots\dot{X}_{m_{n-1}} \end{bmatrix} =$

$$\begin{bmatrix} \frac{\partial X_1}{\partial X_1} & \frac{\partial X_1}{\partial X_2} \cdots & \frac{\partial X_1}{\partial X_m} \\ \frac{\partial X_2}{\partial X_1} & \frac{\partial X_2}{\partial X_2} \cdots & \frac{\partial X_2}{\partial X_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial X_m}{\partial X_1} & \frac{\partial X_m}{\partial X_2} \cdots & \frac{\partial X_m}{\partial X_m} \end{bmatrix} * \begin{bmatrix} X_{1_0}X_{1_1}\ldots X_{1_{n-1}} \\ X_{2_0}X_{2_1}\ldots X_{2_{n-1}} \\ \vdots \\ X_{m_0}X_{m_1}\ldots X_{m_{n-1}} \end{bmatrix}$$

which is equivalent to

$$\begin{bmatrix} \frac{X_{1_1}-X_{1_0}}{t_c} & \frac{X_{1_2}-X_{1_1}}{t_c} \ldots & \frac{X_{1_n}-X_{1_{n-1}}}{t_c} \\ \frac{X_{2_1}-X_{2_0}}{t_c} & \frac{X_{2_2}-X_{2_1}}{t_c} \ldots & \frac{X_{2_n}-X_{2_{n-1}}}{t_c} \\ & \vdots & \\ \frac{X_{m_1}-X_{m_0}}{t_c} & \frac{X_{m_2}-X_{m_1}}{t_c} \ldots & \frac{X_{m_n}-X_{m_{n-1}}}{t_c} \end{bmatrix} =$$

$$\begin{bmatrix} \frac{\partial X_1}{\partial X_1} & \frac{\partial X_1}{\partial X_2} \cdots & \frac{\partial X_1}{\partial X_m} \\ \frac{\partial X_2}{\partial X_1} & \frac{\partial X_2}{\partial X_2} \cdots & \frac{\partial X_2}{\partial X_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial X_m}{\partial X_1} & \frac{\partial X_m}{\partial X_2} \cdots & \frac{\partial X_m}{\partial X_m} \end{bmatrix} . \begin{bmatrix} X_{1_0}X_{1_1}\ldots X_{1_{n-1}} \\ X_{2_0}X_{2_1}\ldots X_{2_{n-1}} \\ \vdots \\ X_{m_0}X_{m_1}\ldots X_{m_{n-1}} \end{bmatrix}$$

assuming that the state vector measurements are captured at regular time intervals of t_c. Note that $e^{t_c} \approx 1 + t_c$, of $e^{t_c} = 1 + t_c + \frac{{t_c}^2}{2!} + \frac{{t_c}^3}{3!} + ...$ improves as t_c gets closer to 0 [1]. The solution to this system of linear differential equations is

$$\begin{bmatrix} X_{1_1}X_{1_2}\ldots X_{1_n} \\ X_{2_1}X_{2_2}\ldots X_{2_n} \\ \vdots \\ X_{m_1}X_{m_2}\ldots X_{m_n} \end{bmatrix} =$$

$$exp^{\begin{bmatrix} \frac{\partial X_1}{\partial X_1} & \frac{\partial X_1}{\partial X_2} \cdots & \frac{\partial X_1}{\partial X_m} \\ \frac{\partial X_2}{\partial X_1} & \frac{\partial X_2}{\partial X_2} \cdots & \frac{\partial X_2}{\partial X_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial X_m}{\partial X_1} & \frac{\partial X_m}{\partial X_2} \cdots & \frac{\partial X_m}{\partial X_m} \end{bmatrix} .t_c} * \begin{bmatrix} X_{1_0}X_{1_1}\ldots X_{1_{n-1}} \\ X_{2_0}X_{2_1}\ldots X_{2_{n-1}} \\ \vdots \\ X_{m_0}X_{m_1}\ldots X_{m_{n-1}} \end{bmatrix}$$

Further analysis of such inverse problems might require matrix decomposition, e.g., the system above may be redefined as

$$X(_{after}) = exp^{[A*t_c]} * X(_{before})$$

C. Solving time series inverse problem using matrix manipulation

In solving time series inverse problems of the form

$$X(_{after}) = exp^{[A*t_c]} * X(_{before})$$

where the jacobian matrix A must be identified and t_c is the time interval of very small magnitude, Idowu and Bown [3] developed the transposive regression method for finding $E = exp^{[A*t_c]}$.

1) Using the transposive regression method:

Steps

1) $E * X(_{before}) = X(_{after})$
2) $X(_{before})^T * E_1^T = X(_{after})^T$
3) $X(_{before}) * X(_{before})^T * {E_1}^T = X(_{before}) * X(_{after})^T$
4) $E^T = [X(_{before}) * X(_{before})^T]^{-1} * X(_{before}) * X(_{after})^T$
5) $E = ([X(_{before}) * X(_{before})^T]^{-1} * X(_{before}) * X(_{after})^T)^T$
6) $E =$ $X(_{after}) * X(_{before})^T * ([X(_{before}) * X(_{before})^T]^{-1})^T$

In steps 1-2 recasting the problem by matrix transposition is essential, because each state is represented by a column-vector either in $X(_{before})$ or $X(_{after})$, where $X(_{before})$ is an array of states before the transformation $X(_{before}) = [X(_0)\ X(_1)\ \ldots\ X(_{t-1})]$, and $X(_{after})$ is an array of states after the transformation $X(_{after}) = [X(_1)\ X(_2)\ \ldots\ X(_t)]$. Steps 3-4 illustrate an application of the Moore-Penrose pseudoinverse, a widely known type of matrix pseudoinverse, independently introduced by Moore [4], Bjerhammar [5], and Penrose [6]. Finally, in steps 5-6, retranspositions put E in proper order.

D. *Finding matrix logarithmic inverse approximation*

The difficulty in finding A is in calculating the principal matrix logarithm of E; that is, the exact inverse of $exp^{(A*t_c)}$. In [3] we developed a technique for finding the jacobian matrix A from the resultant matrix $exp^{(A*t_c)}$ without using the well-known eigenvector and eigenvalue method.

$$exp^{(J*t_c)} = E_1 \approx E_2$$

$$J = \frac{logm(E_1)}{t_c} \approx \frac{logm(E_2)}{t_c}$$

where logm(...) represents the matrix logarithm function. We introduce a scaling factor μ to t_c in order to approximate ∇p such that $\nabla p \approx t_c * \mu$ to satisfy $E^{\mu} = exp^{(A*\nabla p)} \approx I + (A * \nabla p)$. Therefore, A can be approximated $\frac{({E_m}^{\mu} - I)}{(t_c * \mu)}$.

VI. Conclusion

We have demonstrated that through simple matrix manipulation techniques, it is possible to derive methods for creating and deconstructing matrices with known determinants. A robust matrix-based framework to enable simplified creation and evaluation of system identification and parameter estimation problems and solutions is being developed, and the method presented here is an essential part of that framework. This framework, an important tool for developing and managing optimisation methods, is required in systems biology and applicable to other areas such as artificial intelligence, network science, etc. The inference method is sophisticated enough for the identification of ODE models from time series data. In future work, we will demonstrate how these matrix decomposition and composition methods presented here may be applied to the development of new techniques for understanding complex network structures.

Acknowledgment

We gratefully acknowledge the support from the Northwood Trust.

References

[1] G. Strang, *Linear Algebra and its Applications*, 3rd ed., ISBN 0155510053, 1988.

[2] A. S. Householder, *The Theory of Matrices in Numerical Analysis*, New York Dover Publications, ISBN 0486617815, 1975.

[3] M. A. Idowu, J. L. Bown, "Towards an Exact Reconstruction of a Time-Invariant Model from Time Series Data", *Journal of Comp. Sci. and Syst. Biol.* vol. 4, pp. 055-070, doi:10.4172/jcsb.1000077, 2011.

[4] E. H. Moore, "On the Reciprocal of the General Algebraic Matrix", *Bulletin of the American Mathematical Society* vol. 26, pp. 394395, projecteuclid.org/euclid.bams/1183425340, 1920.

[5] A. Bjerhammar, "Application of Calculus of Matrices to Method of Least Squares; with Special References to Geodetic Calculations", *Trans. Roy. Inst. Tech.*, Stockholm 49, 1951.

[6] R. Penrose, "A Generalized Inverse for Matrices", *Proceedings of the Cambridge Philosophical Society*, vol. 51, pp. 406413, doi:10.1017/S0305004100030401, 1955.