Linear Algebra Primer

This section reviews mathematical methods in linear algebra that are useful for orbit control and response matrix analysis in storage rings and linear accelerators. Although most physical problems are non-linear by nature, they often approximate linear systems for small amplitude motion or short timescales. Linearization turns otherwise intractable, non-linear problems into manageable linear problems. A host of robust analytical techniques from linear algebra speed up the process of finding numerical solutions by orders of magnitude.

INTRODUCTION

Linear algebra is a branch of mathematics that concerns solving systems of equations in the *linear* approximation. The most basic system can be cast in the form

$$Ax = b$$

where A is an *m x n* transformation matrix that takes column vector x into column vector b. Typical examples include beam transport or relativistic transformations. Note that column vector x has dimension n (the number of columns in A) and column vector b has dimension m (number of rows in A).

For our applications, x will be a column vector of variables, and b a column vector of constraints. In orbit control, A is the corrector-to-BPM response matrix (R), x contains the desired corrector strengths (θ) and b contains the orbit shift (x).

Theory: $Ax = b$
Orbit Control: $R\theta = x$ (b => x, x => θ)
[Apologies for the syntax confusion surrounding 'x']

For response matrix analysis, A contains numerical derivatives, x contains model parameters and b contains measured response matrix data.

Each element of the matrix A can be written as a partial derivative that tranforms from one variable set to another. A_{ij} is a gain factor from the j^{th} input to the i^{th} output. The i^{th} row of A concerns the i^{th} output. The j^{th} column of A concerns the j^{th} input. The value of the elements in A come from physics and geometry. An orbit response matrix has matrix elements

$$A_{ij} = \frac{\Delta x_i}{\Delta \theta_j}$$

where x_i is the orbit shift at the i^{th} BPM and θ_j is the kick at the j^{th} corrector.

SIMULTANEOUS EQUATIONS
Each row of Ax=b can be viewed as an equation for a plane in n-dimensions.

$$d = ax + by + cz \qquad \text{(plane in 3-dimensions)}$$

or

$$b_i = \Sigma A_{ij} x_j = \Sigma \frac{\Delta b_i}{\Delta x_j} x_j \qquad \text{(plane in n-dimensions)}$$

The solution to the set of *simultaneous equations* is the location where n-dimensional planes intersect. *Solving* a matrix problem requires finding elements of the n-dimensional column vector x.

For orbit correction applications, each row of Rθ = x reads

'orbit shift = linear superposition of corrector kicks'

Given an orbit constraint x_{orbit}, the goal is to find the corresponding set of corrector magnets to move the beam to the desired position. Examples include steering of the entire closed orbit in circular accelerators and closed orbit 'bumps' in linear or circular accelerator structures. We will focus on the typical over-constrained case where there are more constraints than variables but also look at the under-constrained case where there are more variables than constraints.

COLUMNS OF A
Another way to look at the system Ax=b is that column vector b is a linear combination of the columns in A. In this case, the elements of x are the coefficients of the column vectors,

$$b = A_{.1}x_1 + A_{.2}x_2 + ... + A_{.n}x_n$$

In terms of linear algebra, *the vector b lies in the column space of A* - there is a linear combination of the columns in A that add up to produce column vector b.

This approach has a physically intuitive interpretation for orbit control: each column of A is an orbit shift produced by one corrector magnet. A linear superposition of corrector magnets produces a linear combination of orbit perturbations that add up to give the orbit shift, b. Later we will make a singular value decomposition of A and the linear superposition will be on eigenvectors instead of individual correctors.

OVER- AND UNDER-CONSTRAINED PROBLEMS

In real life applications, often there is no exact value for x that satisfies Ax=b, particularly when the number of constraints is greater than the number of variables (m>n). Mathematically, there is no solution to the set of equations for intersecting planes. Equivalently, column vector b does not lie in the column space of A. To find a solution we turn to 'least squares' to minimize the geometric distance between column vectors Ax and b, i.e. to minimize the length of the error vector $\varepsilon = \min\|Ax-b\|^2$. Alternatively we turn to the more robust singular value decomposition that minimizes the length of the solution vector x. SVD also works when the matrix A is 'ill-conditioned' or 'rank-deficient' indicating the rows or columns of the response matrix are not linearly independent.

FUNDAMENTAL SUBSPACES OF A

Study of the fundamental subspaces of a matrix A takes us somewhat off the beaten path but provides a conceptual framework that will be useful when we get to singular value decomposition. SVD factorizes the matrix A into a product of matrices that contain basis vectors for each of the four fundamental subspaces of A. Recall from linear algebra that a *subspace is spanned* by a *basis set*, in this case sets of *linearly independent vectors associated with matrix A.* The SVD factorization of a matrix A is reminiscent of the eigenvalue/eigenvector factorization:

$A = UWV^T$ (SVD)

$A = X\Lambda X^{-1}$ (Eigenvector)

where U and V contain 'eigenvectors' and W contains 'eigenvalues' on the diagonal. The eigenvectors in U and V span the four fundamental subspaces of A

The easiest example of a subspace of A is the column space: if a 4x4 matrix has four linearly independent column vectors, they span a 4-dimensional space. In other words, any column vector b in that space can be written as a linear combination of the four column vectors in the matrix. Matrix notation writes this in compact form,

$$Ax=b = (\text{column}_1)x_1 + (\text{column}_2)x_2 + (\text{column}_3)x_3 + (\text{column}_4)x_4$$

Ax=b means finding combinations of columns of A that produce b.

To be up front, the subspaces are:

1. column space of A: (Ax=b)
3. null space of A (Ax=0)
2. row space of A ($A^Ty=b$)
4. left null space of A ($y^TA=0$)

What are these subspaces and what do they mean for orbit control? Recall that an *m x n* matrix ‘A’ takes an n-dimensional row vector into an m-dimensional column vector. In other words, it takes row space (R^n) into column space (R^m). There are *two* subspaces in R^n and *two* subspaces in R^m. In general, all four bases in all four sub-spaces are needed to diagonalize the *m x n* matrix A! This is the job of SVD. We will see the mathematical structure and use diagrams to look at this further in the section on SVD. Paraphrasing Strang (*A.M.M. 100,1993*), here are short descriptions of the subspaces of A:

Column Space of A – Think of matrix A as a function acting on the input vector x. The column space is the range of A: Ax=b. The *column space is spanned by the linearly-independent columns of A.* Analogous to the practice of differential equations, Ax=b has *particular solutions* ‘x’ which are linear combinations of the columns of A.

Row Space of A – The *row space is spanned by the linearly independent rows of A.* The row space and column space have the same rank, r=rank(A). Taking the transpose of A we have $A^Ty=b$. In this case, b is a linear combination of the columns of A^T (rows of A). b lies in the range of A^T.

Null Space or Kernel of A – *The null space is spanned by the set of vectors satisfying Ax=0.* They are the homogenous solutions. In orbit correction, if you apply a 'null' corrector pattern (in the null space) the orbit does not move at the BPMS(!). In order for matrix A to contain null vectors it must be rank-deficient, i.e. some rows or columns are linear combinations of each other.

Left Null Space of A – The left null space is spanned by the set of vectors satisfying $A^Ty=0$ or $y^TA=0$ (hence ‘left’ nullspace). Errors in least-squares analysis are relegated to the left null space of A.

ORTHOGONALITY AND DIMENSIONALITY OF SUBSPACES OF A

Just as vectors can be orthogonal (inner product zero) entire vector spaces can be orthogonal. When two vector spaces are orthogonal, all vectors in one space are orthogonal to all vectors in the other. For an *m x n* matrix A of rank(r), the four subspaces have the following properties:

<u>Input side</u> (x-vectors in Ax=b with dimension R^n)

Row Space (dimension r) is orthogonal to Null Space (dimension n - r)

Interpretation I: given a particular solution x_p to $Ax_p=b$, any homogeneous x_h solution $Ax_h=0$ is orthogonal: $x_p \cdot x_h = 0$

Interpretation II: corrector sets can be decomposed into components lying in the row space and components lying in the null space. Eliminating the null space component does not move the beam at the BPMS but reduces the overall strength of the corrector set (corrector ironing).

Output side (b-vectors in Ax=b with dimension R^m.)

Column Space (dimension r) is orthogonal to Left Null Space (dimension m - r)

Interpretation I: given a least-squares problem Ax=b, the solution x must is only for the component of b in the column space of A. The error vector lies in the (complementary) left null space.

Interpretation II: An orbit in the left null space of a response matrix R cannot be corrected by the correctors associated with R.