

Appendix

7 A Few Words about MATLAB

One of the recommended books is *Digital Image Processing using MATLAB* by Gonzalez, Woods, and Eddins 2004. Do you need this book? Maybe. If you're interested in imaging or if you are not well acquainted with MATLAB and would like a good reference, then it will be helpful. If you have little interest in imaging or signal analysis and already know MATLAB, then it might not be for you since you'll have a good reference already.

Assignment is to read the first 2⁶ pages (Chapters 1 and 2) which will orient you on the use of MATLAB, data representations, m-file editing, plotting, and lots of other good stuff. MATLAB is the de facto programming environment of engineering, so its good to know and free for your use throughout campus including all the toolboxes. Such a deal! I give two HW I problems to get you familiar with data handling and just a few common functions.

- Get familiar with MATLAB and the command windows, and workspaces. Get a jump drive (memory stick) if you're not working on one system. Immediately set your current directory.
- Get familiar with the editor.
- Getting help. `helpbrowser`, `doc`, `help`, `lookfor`.
- Figure out how to save your workspace and load your workspace.
- Note that $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^t$ is often used in the mathematical analysis of most texts and the literature. However, MATLAB wants vector and matrix indices to be > 0 , so in MATLAB, `x = [x(1) x(2) ... x(N)]`.
- Go through the examples in the text for reading files from disk, writing files to disk in various formats (jpg, tif, bmp, eps ...), displaying files, and generally dealing with the different data classes and inter-conversions.

Example 7.1. `A=[1 3 5 7;2 4 6 8; 5 2 3 4];`

```
size(A)
```

```
x = A(:,3)
```

```
y = A(3,:)
```

```
B = A(:,1:3)
```

If you've not worked with MATLAB much be sure to consider the last two homework problems. There are many syntax pitfalls.

8 APPENDIX A: Review of Matrix Methods with MATLAB

8.1 Definitions and Properties

- An $N \times M$ dimensional matrix \mathbf{A} is a rectangular table of elements (numbers, variables, etc.) consisting of rows N and columns M .
- Vector \mathbf{x} is a 1-D matrix, and therefore may be a column vector $N \times 1$ or a row vector $1 \times M$. By default, \mathbf{x} is a column vector and \mathbf{x}^t (transpose) is a row vector.
- In a square matrix \mathbf{A} , $N = M$.
- Singular matrices have no inverse; non-square matrices are all singular.
- Nonsingular matrices are defined as matrices with inverses.
- In a transpose operation on a matrix, the row and columns are interchanged.
- A system of equations has at least two equations. For example, in a predator-prey system,

$$\dot{N}(t) = kN(t) - \alpha N(t)M(t) \quad (\text{prey})$$

$$\dot{M}(t) = \gamma\alpha N(t)M(t) - \beta M(t), \quad \alpha \geq 0, \beta > 0 \quad (\text{predator}).$$

Definition 8.1. *Products.* Let \mathbf{x} be an $N \times 1$ column vector and \mathbf{y} be an $M \times 1$ column vector. Both are real-valued vectors. \mathbf{x} is said to have order or size or dimension $N \times 1$. Two types of multiplication between these vectors can be defined. If $N = M$, then

$$a = \mathbf{x}^t \mathbf{y} = \sum_{n=1}^N x_n y_n \quad (123)$$

is defined and produces the 1×1 scalar valued result a . Superscript t denotes transpose so that \mathbf{x}^t is a $1 \times N$ row vector. Eq. (123) is also called a dot product, $a = \mathbf{x} \cdot \mathbf{y}$, which is not defined for $N \neq M$. However, the matrix product is defined for any N, M ;

$$\mathbf{A} = \mathbf{x} \mathbf{y}^t = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{21} & A_{22} & \cdots & A_{2M} \\ \vdots & \vdots & & \vdots \\ A_{N1} & A_{N2} & \cdots & A_{NM} \end{pmatrix} \quad (124)$$

produces an $N \times M$ rectangular matrix.

Definition 8.2. If \mathbf{A} is an $M \times N$ matrix and \mathbf{B} is an $N \times P$ matrix, then the product \mathbf{AB} is defined to be an $M \times P$ matrix.

Example 8.1.

$$\mathbf{AB} = \begin{pmatrix} 2 & 1 \\ -1 & 0 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 3 & 1 & 5 & -1 \\ 4 & -2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 10 & 0 & 11 & -2 \\ -3 & -1 & -5 & 1 \\ 13 & 1 & 16 & -3 \end{pmatrix}, \quad (125)$$

while \mathbf{BA} is undefined. (And MATLAB will complain; try it.)

Definition 8.3. *If two vectors with complex elements \mathbf{x} and \mathbf{y} both have dimension N , we can define a scalar inner product*

$$(\mathbf{x}, \mathbf{y}) \triangleq \mathbf{x}^\dagger \mathbf{y} = (\mathbf{x}^*)^t \mathbf{y} = \sum_{n=1}^N x_n^* y_n .$$

Since the vectors can have complex elements, we use the conjugate transpose operator \mathbf{x}^\dagger . Note that the inner product equals the dot product for real vectors. Unlike our convention, MATLAB assumes \mathbf{x} is a $1 \times N$ row vector, and so you must be careful to transcribe between the two conventions. You can obtain the inner product stated above in MATLAB by using `conj(x*y')` which is the same as inverting the order (`y*x'`), since $(\mathbf{x}, \mathbf{y}) = (\mathbf{y}, \mathbf{x})^*$. For example, let $\mathbf{x} = (1 + i, 3, -3i)^t$ and $\mathbf{y} = (4, -3 - i, -2)^t$. I get that $(\mathbf{x}, \mathbf{y}) = -5 - 13i$ algebraically and in MATLAB using the above rules.

In MATLAB the tick mark applies a conjugate transpose to a vector or matrix. To obtain a transpose without also applying a conjugate, i.e., obtain $\mathbf{x}^t \mathbf{y}$, use `(y*x.')`. Results with `'` and `.` are equal if \mathbf{x} and \mathbf{y} are both real. The complex conjugate transpose of \mathbf{A} , denoted \mathbf{A}^\dagger , is called the Hermitian transpose.

The inner product concept can be generalized to include continuous functions

$$(f, g) = \int_a^b dt f^*(t)g(t) .$$

The implication is that functions of continuous variables may be treated as vectors.

Definition 8.4. *If column vector \mathbf{x} has dimension M and \mathbf{y} has dimension N , the outer product \mathbf{xy}^\dagger forms an $M \times N$ matrix,*

$$\mathbf{xy}^\dagger \triangleq \mathbf{x}(\mathbf{y}^*)^t , \quad \text{and the matrix element } (\mathbf{xy}^\dagger)_{mn} = x_m y_n^* .$$

Note that $\mathbf{x}^t \mathbf{y} = \text{tr}(\mathbf{xy}^\dagger)$. The trace of an outer product equals the inner product. In MATLAB the trace of an $M \times N$ matrix \mathbf{A} , where $M < N$, equals the trace of the leftmost $M \times M$ matrix of \mathbf{A} .

Property 8.1. *Basic properties.*

- $\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C}$ *Associative Law for Multiplication*
- $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$ *Left Distributive Law for Multiplication*
- $(\mathbf{B} + \mathbf{C})\mathbf{A} = \mathbf{BA} + \mathbf{CA}$ *Right Distributive Law for Multiplication*
- $\mathbf{AB} = \mathbf{AC}$ does **not** imply $\mathbf{B} = \mathbf{C}$ if \mathbf{A} is singular, just as with scalar arithmetic
- $(\mathbf{A}^t)^t = \mathbf{A}$
- $(\lambda \mathbf{A})^t = \lambda \mathbf{A}^t$ *for scalar valued λ*
- $(\mathbf{A} + \mathbf{B} + \mathbf{C})^t = \mathbf{A}^t + \mathbf{B}^t + \mathbf{C}^t$ *Distributive Law for Transpose*
- $(\mathbf{ABC})^t = \mathbf{C}^t \mathbf{B}^t \mathbf{A}^t$ *Distributive Law for Transpose*

- $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$ *Identity matrix*
- $\mathbf{A}^3 = \mathbf{AAA}$ *Power law*
- $\mathbf{A} = \mathbf{A}^t$ *Symmetric matrix*
- $\mathbf{A} = -\mathbf{A}^t$ *Skew symmetric matrix*
- If \mathbf{A} and \mathbf{B} are diagonal matrices of the same order then $\mathbf{AB} = \mathbf{BA}$.
- If

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 10 & 12 \\ 14 & 16 \end{pmatrix}$$

then \mathbf{B} is a submatrix of \mathbf{A} . Note that \mathbf{A} is generated in MATLAB using

$\mathbf{A}=[1\ 2\ 3\ 4;5\ 6\ 7\ 8;9\ 10\ 11\ 12; 13\ 14\ 15\ 16]$; because MATLAB follows a row-oriented convention.

Definition 8.5. If $\mathbf{x} = [x_1x_2 \cdots x_n]$ then the magnitude or length of \mathbf{x} is $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$.

Definition 8.6. A unit vector is obtained by dividing each element by the magnitude of the vector: $\hat{\mathbf{x}} = \mathbf{x}/\|\mathbf{x}\|$.

8.1.1 Determinants and Inverses

To find the determinant of \mathbf{A} , viz., $\det\mathbf{A}$, (a) pick any one column or row of the matrix, (b) find the co-factor of each element in that column or row, (c) multiply each element with its cofactor and then sum.

Example 8.2. Find $\det \mathbf{A}$ for

$$\mathbf{A} = \begin{pmatrix} 3 & 5 & 0 \\ -1 & 2 & 1 \\ 3 & -6 & 4 \end{pmatrix}.$$

Let's expand the second column.

$$\begin{aligned} \det \mathbf{A} &= (5)(\text{cofactor of } 5) + (2)(\text{cofactor of } 2) + (-6)(\text{cofactor of } -6) \\ &= 5(-1)^{1+2} \begin{vmatrix} -1 & 1 \\ 3 & 4 \end{vmatrix} + 2(-1)^{2+2} \begin{vmatrix} 3 & 0 \\ 3 & 4 \end{vmatrix} + (-6)(-1)^{3+2} \begin{vmatrix} 3 & 0 \\ -1 & 1 \end{vmatrix} \\ &= (-5)(-7) + (2)(12) + (6)(3) = 77. \end{aligned}$$

It would have been simpler to select a row or column with the most zeros since any choice gives the same answer.

Property 8.2. Determinants.

1. Determinants are only defined for square matrices.

2. If a row or column consists entirely of zeros, the determinant is zero.
3. If two rows or columns of \mathbf{A} are interchanged, then $\det \mathbf{A}$ changes sign.
4. If two rows/columns of \mathbf{A} are identical, then $\det \mathbf{A} = 0$.
5. Multiplication of a matrix by a scalar is different from multiplication of a determinant by a scalar. For example,

$$\lambda \det \mathbf{A} = \begin{vmatrix} \lambda A_{11} & \lambda A_{12} & \lambda A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{vmatrix} \quad \text{and yet} \quad \det \lambda \mathbf{A} = \begin{vmatrix} \lambda A_{11} & \lambda A_{12} & \lambda A_{13} \\ \lambda A_{21} & \lambda A_{22} & \lambda A_{23} \\ \lambda A_{31} & \lambda A_{32} & \lambda A_{33} \end{vmatrix} = \lambda^3 \det \mathbf{A}$$

for this square matrix of order 3.

6. If matrix \mathbf{B} is found from matrix \mathbf{A} by adding to one row of \mathbf{A} , a scalar times another row (or column) of \mathbf{A} , then $\det \mathbf{A} = \det \mathbf{B}$.
7. $\det \mathbf{A}^t = \det \mathbf{A}$.
8. If \mathbf{A} and \mathbf{B} are of the same order, then $\det(\mathbf{A}) \det(\mathbf{B}) = \det(\mathbf{AB})$.

Example 8.3. Find the determinant of

$$\det \mathbf{A} = \begin{vmatrix} 10 & -6 & -9 \\ 6 & -5 & -7 \\ -10 & 9 & 12 \end{vmatrix}.$$

Add the first row to the third row (property 6 above). Then add (-1) times the second column times the third column (property 6). By expansion of cofactors,

$$\det \mathbf{A} = -3 \begin{vmatrix} 10 & -3 \\ 6 & -2 \end{vmatrix} = 6.$$

8.1.2 Matrix Inverse

The inverse of a matrix, denoted as \mathbf{A}^{-1} , is defined only for square matrices where $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$. If \mathbf{A} has an inverse, it is said to be invertible or nonsingular. If \mathbf{A} does not have an inverse it is a singular matrix.

One way to compute an inverse by hand is to first find the cofactor matrix \mathbf{A}^c . If

$$\mathbf{A} = \begin{pmatrix} 3 & 1 & 2 \\ -2 & 5 & 4 \\ 1 & 3 & 6 \end{pmatrix},$$

then

$$\mathbf{A}^c = \begin{pmatrix} (-1)^{1+1} \begin{vmatrix} 5 & 4 \\ 3 & 6 \end{vmatrix} & (-1)^{1+2} \begin{vmatrix} -2 & 4 \\ 1 & 6 \end{vmatrix} & (-1)^{1+3} \begin{vmatrix} -2 & 5 \\ 1 & 3 \end{vmatrix} \\ (-1)^{2+1} \begin{vmatrix} 1 & 2 \\ 3 & 6 \end{vmatrix} & (-1)^{2+2} \begin{vmatrix} 3 & 2 \\ 1 & 6 \end{vmatrix} & (-1)^{2+3} \begin{vmatrix} 3 & 1 \\ 1 & 3 \end{vmatrix} \\ (-1)^{3+1} \begin{vmatrix} 1 & 2 \\ 5 & 4 \end{vmatrix} & (-1)^{3+2} \begin{vmatrix} 3 & 2 \\ -2 & 4 \end{vmatrix} & (-1)^{3+3} \begin{vmatrix} 3 & 1 \\ -2 & 5 \end{vmatrix} \end{pmatrix} = \begin{pmatrix} 18 & 16 & -11 \\ 0 & 16 & -8 \\ -6 & -16 & 17 \end{pmatrix}.$$

The adjoint of \mathbf{A} , denoted \mathbf{A}^a , is defined as the transpose of the cofactor matrix \mathbf{A}^c : $\mathbf{A}^a = (\mathbf{A}^c)^t$. It can be shown that

$$\mathbf{A}^a \mathbf{A} = \mathbf{A} \mathbf{A}^a = \det(\mathbf{A}) \mathbf{I}.$$

Provided that $\det \mathbf{A} \neq 0$, we can divide through by the determinant (remember, $\det \mathbf{A}$ is a scalar!)

$$\left(\frac{\mathbf{A}^a}{\det \mathbf{A}} \right) \mathbf{A} = \mathbf{A} \left(\frac{\mathbf{A}^a}{\det \mathbf{A}} \right) = \mathbf{I}.$$

Therefore $\mathbf{A}^{-1} \triangleq \mathbf{A}^a / \det \mathbf{A}$. From the example above, $\det \mathbf{A} = 3(18) - 1(-16) + 2(-11) = 48$, and

$$\mathbf{A}^{-1} = \frac{\mathbf{A}^a}{\det \mathbf{A}} = \frac{\begin{pmatrix} 18 & 0 & -6 \\ 16 & 16 & -16 \\ -11 & -8 & 17 \end{pmatrix}}{48} = \begin{pmatrix} 3/8 & 0 & -1/8 \\ 1/3 & 1/3 & -1/3 \\ -11/48 & -1/6 & 17/48 \end{pmatrix}.$$

Notice a few things. If \mathbf{A} is diagonal, then the inverse matrix is composed of the inverses of each nonzero element. Also if \mathbf{A} is an upper triangular matrix (only elements from the diagonal up are nonzero) or lower triangular matrix, then \mathbf{A}^{-1} is also upper/lower triangular with diagonal elements given by inverses of diagonal elements of \mathbf{A} , but other elements must be found.

Property 8.3. *Matrix inverses.*

1. *The inverse of a matrix is unique.*
2. *If the determinant of a matrix is zero, the matrix is singular (does not have an inverse).*
3. $(\mathbf{A}^{-1})^{-1} = \mathbf{A}$
4. $(\mathbf{ABC})^{-1} = \mathbf{C}^{-1} \mathbf{B}^{-1} \mathbf{A}^{-1}$ to any number of matrices
5. $(\mathbf{A}^t)^{-1} = (\mathbf{A}^{-1})^t$
6. $(\lambda \mathbf{A})^{-1} = (1/\lambda) \mathbf{A}^{-1}$
7. *The inverse of a nonsingular symmetric matrix is also symmetric.*
8. $\mathbf{A}^{-n} = (\mathbf{A}^{-1})^n$ So that if $\mathbf{A} = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ \frac{1}{2} & 1 \end{pmatrix}$ then $\mathbf{A}^{-2} = (\mathbf{A}^{-1})^2 = \begin{pmatrix} 12 & -6 \\ -6 & 4 \end{pmatrix}^2 = \begin{pmatrix} 180 & -96 \\ -96 & 52 \end{pmatrix}$.

8.2 Systems of Equations

A linear system of equations written as

$$\begin{aligned} 5x - 3y + 2z &= 14, \\ x + y - 4z &= -7, \\ 7x - 3z &= 1, \end{aligned}$$

can also be written in matrix form as $\mathbf{Ax} = \mathbf{b}$ where

$$\mathbf{A} = \begin{pmatrix} 5 & -3 & 2 \\ 1 & 1 & -4 \\ 7 & 0 & -3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \text{and} \quad \mathbf{b} = \begin{pmatrix} 14 \\ -7 \\ 1 \end{pmatrix}. \quad (126)$$

Frequently we wish to solve for \mathbf{x} in terms of \mathbf{A} and \mathbf{b} . Let's quickly review a few methods.

Cramer's Rule. For the system of equations $\mathbf{Ax} = \mathbf{b}$, we can express the i th components of \mathbf{x} , i.e, the unknown x_n , as the quotient of two determinants. The denominator is $\det \mathbf{A}$, and the numerator is the determinant of \mathbf{A} in which the n th column of \mathbf{A} is replaced by vector \mathbf{b} . For example,

$$x_1 = \frac{\begin{vmatrix} b_1 & A_{12} & A_{13} \\ b_2 & A_{22} & A_{23} \\ b_3 & A_{32} & A_{33} \end{vmatrix}}{\det \mathbf{A}}, \quad x_2 = \frac{\begin{vmatrix} A_{11} & b_1 & A_{13} \\ A_{21} & b_2 & A_{23} \\ A_{31} & b_3 & A_{33} \end{vmatrix}}{\det \mathbf{A}}, \quad x_3 = \frac{\begin{vmatrix} A_{11} & A_{12} & b_1 \\ A_{21} & A_{22} & b_2 \\ A_{31} & A_{32} & b_3 \end{vmatrix}}{\det \mathbf{A}}.$$

This only works for square matrices where the number of equations equals the number of unknowns. They must also be square to have determinants. Also Cramer's Rules cannot be applied if $\mathbf{A} = 0$.

Gaussian Elimination. A computational method for solving the system $\mathbf{Ax} = \mathbf{b}$ without matrix operations is Gaussian elimination. It is best described through a simple example.

Example 8.4. *A system of equations:*

$$\begin{aligned} x + 2y + z &= 3 \\ 2x + 3y - z &= -6 \\ 3x - 2y - 4z &= -2 \end{aligned}$$

Solve the first equation for x then substitute into second and third equations.

$$\begin{aligned} x &= 3 - 2y - z \\ y + 3z &= 12 \\ 8y + 7z &= 11 \end{aligned}$$

Solving the second equation for y , substituting the result into the third, and solving that for z gives

$$\begin{aligned} x &= 3 - 2y - z \\ y &= 12 - 3z \\ z &= 85/17 = 5 \end{aligned}$$

Now working back up we find $z = 5$, $y = -3$, $x = 4$.

Simple Matrix Inverse. Solve the following system for x and y .

$$\begin{aligned} x - 2y &= -9 \\ -3x + y &= 2 \end{aligned}$$

Solution.

$$\mathbf{A} = \begin{pmatrix} 1 & -2 \\ -3 & 1 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} -9 \\ 2 \end{pmatrix}.$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} = (-1/5) \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} -9 \\ 2 \end{pmatrix} = \begin{pmatrix} 1 \\ 5 \end{pmatrix}.$$

Notice that you obtain the same answer using Cramer’s rule. However, like the Cramer’s rule approach, there are limitations. Let’s explore those a bit.

Definition 8.7. Matrix Rank. The rank of matrix \mathbf{A} is the order of the largest nonzero minor of \mathbf{A} . A minor of \mathbf{A} is the determinant of a square submatrix (see p 69 in [4]). Also, $\text{rank}(\mathbf{A})$ (which is how its expressed in MATLAB) is an estimate of the number of linearly independent rows or columns of \mathbf{A} . Note that the largest value that $\text{rank}(\mathbf{A})$ can have is the smallest dimension of the $M \times N$ order matrix \mathbf{A} , i.e., $\text{rank}(\mathbf{A}) \leq \min(M, N)$.

(a) The system $\mathbf{Ax} = \mathbf{b}$ is consistent (has at least one solution) iff $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}_\mathbf{b})$ (defined below.) (b) If the system $\mathbf{Ax} = \mathbf{b}$ is consistent, \mathbf{A} is square of order N , and the $\text{rank}(\mathbf{A}) = k$, then solutions are expressible in terms of $N - k = 0$ arbitrary unknowns. (p 77 [4]).

Example 8.5. Consider the system

$$\begin{aligned} 2x - 3y + z &= -1 \\ x - y + 2z &= 2 \\ 2x + y - 3z &= 3 \end{aligned}$$

$$\mathbf{A} = \begin{pmatrix} 2 & -3 & 1 \\ 1 & -1 & 2 \\ 2 & 1 & -3 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} -1 \\ 2 \\ 3 \end{pmatrix} \quad \mathbf{A}_\mathbf{b} = \begin{pmatrix} 2 & -3 & 1 & -1 \\ 1 & -1 & 2 & 2 \\ 2 & 1 & -3 & 3 \end{pmatrix}$$

We can show that $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}_\mathbf{b}) = 3$, hence the system is consistent (has a solution). Since $N = 3$ and $k = 3$, there are $N - k$ arbitrary unknowns, and therefore the solution is unique. Also, because the matrix is full rank, it is nonsingular. So its inverse exists, $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, where we find $x = 2, y = 2, z = 1$.

General Matrix Solution. Assume a system is found to be consistent and of rank k . This means there is at least one $k \times k$ submatrix in \mathbf{A} with a nonzero determinant. Using only the k equations in the submatrix and picking out only the k unknowns whose coefficients contributed to the nonsingular submatrix, isolate the x_1, x_2, \dots, x_k unknowns and rewrite the smaller system with corresponding b coefficients. Then solve for \mathbf{x}_k by matrix inversion.

Example 8.6.

$$\begin{aligned} x + 4y + 7z &= -1 \\ 2x + 5y + 8z &= -2 \\ 3x + 6y + 9z &= -3 \end{aligned}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

We find that $k = 2$ as seen by considering the submatrix

$$\mathbf{S} = \begin{pmatrix} 1 & 4 \\ 2 & 5 \end{pmatrix}$$

This is just one of several submatrices that could be selected. The new system is

$$\mathbf{S}\mathbf{x}' = \mathbf{b}' \quad \text{or} \quad \begin{pmatrix} 1 & 4 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 - 7z \\ -2 - 8z \end{pmatrix}$$

$$\mathbf{x}' = \mathbf{S}^{-1}\mathbf{b}' = \begin{pmatrix} -5/3 & 4/3 \\ 2/3 & -1/3 \end{pmatrix} \begin{pmatrix} -1 - 7z \\ -2 - 8z \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix} + z \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

This shows that the solution is not unique and can only be obtained in terms of z . If another submatrix was selected, the solution may be in terms of another variable.

8.3 Eigenanalysis

A nonzero vector \mathbf{x} is an eigenvector of a square matrix \mathbf{A} if there exists a scalar λ such that $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$, and λ is an eigenvalue of \mathbf{A} . Note that eigenvalues and eigenvectors are defined only for square matrices and $\mathbf{x} = \mathbf{0}$ cannot be an eigenvector of a system.

Example 8.7. Is $\mathbf{x} = [1 \ 1]^t$ an eigenvector of $\mathbf{A} = [1 \ 2; \ 3 \ 4]$; ?

$$\mathbf{A}\mathbf{x} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 3 \\ 7 \end{pmatrix} \stackrel{?}{=} \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix} .$$

The answer is “no” since there is no value of λ for which the statement above is true.

As we will see later, eigenfunctions and eigenvectors are important in the analysis of systems; for example, detectors and other components of measurement devices. Now let’s review mathematical properties through examples involving measurement signals as vectors. This is a preview of linear systems.

We first need to define the vector notations for measurement signals that are sampled in time or space. Let the signal $f(t)$ be a continuous function of time. Alternatively we may have a signal $f(x)$ that is a continuous function of one-dimensional spatial position. A uniform sampling of time on the interval T , where $t = nT$ for $n = \dots, -1, 0, 1, 2, \dots$, yields the discrete-time (sampled) signal $f_s(t) = f(nT) = f[n] = \mathbf{f} = \{f_n\}$. For spatially sampled signals, $f_s(x) = f(nX) = f[n] = \mathbf{f} = \{f_n\}$, where $x = nX$. The form $\{f_n\}$ indicates the set of samples in the vector. We will use parentheses to indicate a function of a continuous variable $f(t)$, where the independent variable usually has units; in this case time. Square brackets denote a function of a discrete variable $f[n]$, where the independent variable is unitless. We may use any of these notations depending on the context.

Example 8.8. Determine if the following discrete-time signals could be eigenfunctions of the linear measurement system defined below. (a) $f[k] = \exp[i\Omega k]$ and (b) $f[k] = 5^k u[k]$.

First let’s define the measurement system mathematically. Let $\mathbf{g}[f[k]] = \mathbf{g}[\mathbf{f}]$ be a data vector obtained through a linear mapping of the object vector \mathbf{f} into the data space. This is just a precise way of saying $f(t)$ is some phenomenon like blood flow in the body and \mathbf{g} is a time-series vector of blood flow measurements. Data samples are given through the linear function $g[n] = \sum_{k=-\infty}^{\infty} h[n-k]f[k]$

(recognize this function?). This relationship is expressed more compactly by building a matrix \mathbf{H} having rows that are time reversed and shifted vectors $\mathbf{h}' = h[n-k] = h[-(k-n)]$. The shift integer n is constant for each element in a row of \mathbf{H} . Constructing the system matrix \mathbf{H} in this manner (more details later) we can write $\mathbf{g} = \mathbf{H}\mathbf{f}$, where \mathbf{g} is an $N \times 1$ vector, \mathbf{f} is a $K \times 1$ vector and \mathbf{H} is an $N \times K$ matrix.

Solution to (a).

$$\begin{aligned} \mathbf{H}\mathbf{f} &\stackrel{?}{=} \lambda\mathbf{f} \longrightarrow \mathbf{g}[\mathbf{f}] \stackrel{?}{=} \text{constant} \times \mathbf{f} . \\ g[n] &= \sum_{k=-\infty}^{\infty} h[n-k] \exp[i\Omega k] = \sum_{k=-\infty}^{\infty} h[k] \exp[i\Omega(n-k)] \\ &= \exp[i\Omega n] \sum_{k=-\infty}^{\infty} h[k] \exp[-i\Omega k] \\ &= H(i\Omega) \exp[i\Omega n] = \lambda f[n] , \end{aligned}$$

so YES, $f[k]$ in part (a) is an eigenvector of this linear system. Notice that $H(i\Omega)$ is a constant with respect to time. (Later we'll see that $H(i\Omega) = \sum_{k=-\infty}^{\infty} h[k] \exp[-i\Omega k]$ is the discrete-time Fourier transform of $h[k]$.)

Solution to (b). First note that $u[n]$ is a step function

$$u[n] = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

It represents a switch that is turned on when the argument of $u[n]$ is zero. More later.

$$\begin{aligned} \mathbf{g}[\mathbf{f}] &\stackrel{?}{=} \text{constant} \times \mathbf{f} . \\ g[n] &= \sum_{k=-\infty}^{\infty} h[k] 5^{n-k} u[n-k] \\ &= 5^n \sum_{k=-\infty}^n h[k] 5^{-k} , \end{aligned}$$

so NO, \mathbf{f} in part (b) is not an eigenvector of the linear system since the sum depends on n .

Definition 8.8. Characteristic equation. If \mathbf{x} is an eigenvector of matrix \mathbf{A} and λ the associated eigenvalue, then

$$\begin{aligned} \mathbf{A}\mathbf{x} &= \lambda\mathbf{x} \\ (\mathbf{A} - \lambda\mathbf{I})\mathbf{x} &= \mathbf{0} , \end{aligned}$$

which is a linear homogeneous system of equations for \mathbf{x} . If $\det(\mathbf{A} - \lambda\mathbf{I}) \neq 0$, then $\mathbf{x} = (\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{0} = \mathbf{0}$. Since $\mathbf{x} = \mathbf{0}$ cannot be an eigenvector of the system, we have that \mathbf{x} is an eigenvector iff

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 . \tag{127}$$

Eq (127) is the characteristic equation for \mathbf{A} , and its roots determine the eigenvalues of \mathbf{A} .

Example 8.9. Find the eigenvalues for

$$\mathbf{A} = \begin{pmatrix} 2 & -1 & 1 \\ 3 & -2 & 1 \\ 0 & 0 & 1 \end{pmatrix} .$$

Obviously we process from the bottom row.

$$\mathbf{A} - \lambda \mathbf{I} = \begin{pmatrix} 2 & -1 & 1 \\ 3 & -2 & 1 \\ 0 & 0 & 1 \end{pmatrix} - \lambda \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2-\lambda & -1 & 1 \\ 3 & -2-\lambda & 1 \\ 0 & 0 & 1-\lambda \end{pmatrix}$$

$\det(\mathbf{A} - \lambda \mathbf{I}) = (1 - \lambda)[(2 - \lambda)(-2 - \lambda) + 3] = (1 - \lambda)(\lambda^2 - 1) = 0$ The roots give the eigenvalues $\lambda_1 = \lambda_2 = 1, \lambda_3 = -1$. We have one eigenvalue $\lambda = 1$ of multiplicity two and another $\lambda = -1$ of multiplicity one. If \mathbf{A} is square and of order N then the characteristic equation is a polynomial of order N and there are N eigenvalues, counting multiplicity.

8.3.1 Eigenvectors

For each eigenvalue, there will be at least one eigenvector found from the solution to the equation $(\mathbf{A} - \lambda_i \mathbf{I})\mathbf{x}_i = \mathbf{0}$.

Example 8.10. Using the example above, lets solve for \mathbf{x}_A corresponding to $\lambda_A = 1$.

$$\begin{aligned} (\mathbf{A} - \mathbf{I})\mathbf{x}_A &= \mathbf{0} \\ \left[\begin{pmatrix} 2 & -1 & 1 \\ 3 & -2 & 1 \\ 0 & 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right] \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_A &= \mathbf{0} \\ \begin{pmatrix} 1 & -1 & 1 \\ 3 & -3 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}_A &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} . \\ x_1 - x_2 + x_3 &= 0 \\ 3x_1 - 3x_2 + x_3 &= 0 \end{aligned}$$

Therefore $\mathbf{x}_A = [x_1(1 \ 1 \ 0)^t]$ where $x_3 = 0$ and x_1 is arbitrary. If you try this in MATLAB use $[\mathbf{V}, \mathbf{D}] = \text{eig}(\mathbf{A})$ and you'll get a matrix \mathbf{V} whose columns are eigenvectors and a diagonal matrix \mathbf{D} of eigenvalues. (Notice you get the first eigenvector $\mathbf{x}_A = (0.7071 \ 0.7071 \ 0)^t$ since MATLAB uses a value for x_1 .)

Property 8.4. Eigenvalues and Eigenvectors.

1. The trace of a matrix, $tr(\mathbf{A})$ or `trace(A)` in MATLAB, is the sum of diagonal elements.
2. The sum of the eigenvalues equals the trace of the matrix for square \mathbf{A} .
 $\sum_{n=1}^N \lambda_n = tr(\mathbf{A})$.
3. A matrix is singular iff it has a zero eigenvalue.

4. Eigenvalues of an upper (lower) triangular matrix are elements on the main diagonal; e.g., the eigenvalues for

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 4 & -1 \end{pmatrix}$$

are 1, 1, -1.

5. If λ is an eigenvalue of \mathbf{A} and \mathbf{A} has an inverse, then $1/\lambda$ is an eigenvalue of \mathbf{A}^{-1} . Note that the eigenvector \mathbf{x} corresponding to eigenvalue λ is also an eigenvector corresponding to $1/\lambda$ if the above conditions hold.
6. If λ is an eigenvalue for \mathbf{A} , then $\alpha\lambda$ is an eigenvalue of $\alpha\mathbf{A}$ for any scalar α .
7. If λ is an eigenvalue for \mathbf{A} , then λ is an eigenvalue for \mathbf{A}^t .
8. The product of eigenvalues (including multiplicity) of a matrix equals the determinant of the matrix: $\prod_{i=1}^N \lambda_i = \det\mathbf{A}$ for \mathbf{A} of order $N \times N$.
9. A set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, all with the same dimension, are linearly independent if for

$$c_1\mathbf{x}_1 + c_2\mathbf{x}_2 \dots c_N\mathbf{x}_N = \mathbf{0}$$

the only set of scalar constants that satisfies the equation is $c_1 = c_2 = \dots = c_N = 0$

10. (see [4] pp 96-102). Since any scalar multiple of an eigenvector is also an eigenvector, there are an infinite number of eigenvectors associated with each eigenvalue. However there are, at most, only N linearly independent eigenvectors for an $N \times N$ matrix.

Example 8.11. It is easy to see that the three eigenvalues for the upper triangular matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix}$$

all equal 2. (See property 4 above.) The eigenvector is

$$\begin{aligned} (\mathbf{A} - 2\mathbf{I})\mathbf{x} &= \mathbf{0} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} &= \mathbf{0} \\ x_2 &= 0 \\ x_3 &= 0 \\ 0 &= 0. \end{aligned}$$

Therefore $\mathbf{x} = (x_1 \ 0 \ 0)^t$. Setting $x_1 = 1$ shows that $\lambda = 2$ generates only one linearly independent eigenvector, $\mathbf{x} = (1 \ 0 \ 0)^t$.

11. Assume an $N \times N$ matrix \mathbf{A} with eigenvalue λ of multiplicity k . Then the number of linearly independent eigenvectors of \mathbf{A} associated with λ is given by $\rho = N - \text{rank}(\mathbf{A} - \lambda\mathbf{I})$ and $1 \leq \rho \leq k$.

From the example above, we see that $N = 3$ and $\text{rank}(\mathbf{A} - 2\mathbf{I}) = 2$. Therefore there are $3-2=1$ linearly independent eigenvector associated with $\lambda = 2$, which is what we found.

12. Eigenvectors associated with distinct eigenvalues are linearly independent.

8.4 Similar Matrices and Special Matrices (Chapter 8, [4])

8.4.1 Similar Matrices

Matrix \mathbf{A} is similar to matrix \mathbf{B} if there exists a nonsingular matrix \mathbf{P} such that

$$\mathbf{A} = \mathbf{P}^{-1}\mathbf{B}\mathbf{P}.$$

Example 8.12. Are \mathbf{A} and \mathbf{B} similar?

$$\mathbf{A} = \begin{pmatrix} 4 & 3 \\ -2 & -1 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 5 & -4 \\ 3 & -2 \end{pmatrix}$$

Solution. They will be similar if there is a nonsingular

$$\mathbf{P} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

such that $\mathbf{PA} = \mathbf{BP}$.

$$\begin{pmatrix} (4a - 2b) & (3a - b) \\ (4c - 2d) & (3c - d) \end{pmatrix} = \begin{pmatrix} (5a - 4c) & (5b - 4d) \\ (3a - 2c) & (3b - 2d) \end{pmatrix}$$

Equating elements gives

$$\begin{aligned} -a - 2b + 4c &= 0 \\ 3a - 6b + 4d &= 0 \\ -3a + 6c - 2d &= 0 \\ -3b + 3c + d &= 0 \end{aligned}$$

Solutions are $a = -2d/3$, $b = d/3$, $c = 0$, d is arbitrary. Therefore

$$\mathbf{P} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \frac{d}{3} \begin{pmatrix} -2 & 1 \\ 0 & 3 \end{pmatrix}$$

So, provided $d \neq 0$, \mathbf{A} is similar to \mathbf{B} .

Similar matrices have the same characteristic equations and therefore the same eigenvalues. However, if two matrices have the same characteristic equation, they are not necessarily similar.

A matrix is diagonalizable if it is similar to a diagonal matrix \mathbf{D} . Diagonal matrices are important because of their simplicity in calculations (inversions, and importance in statistics (covariance matrix)).

So how do we diagonalize a matrix? One way that works for simple matrices is to note that the diagonal element of \mathbf{D} are eigenvalues:

Example 8.13.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$$

has eigenvalues -1 and 5. So then a diagonal matrix similar to \mathbf{A} is either $\begin{pmatrix} -1 & 0 \\ 0 & 5 \end{pmatrix}$ or $\begin{pmatrix} 5 & 0 \\ 0 & -1 \end{pmatrix}$.

Let's look at two properties before answering the question.

Property 8.5. Let $\mathbf{B} = (\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_n \ \dots \ \mathbf{b}_N)$ be an $N \times N$ matrix composed of N column vectors. Then

$$\mathbf{AB} = (\mathbf{Ab}_1 \ \mathbf{Ab}_2 \ \dots \ \mathbf{Ab}_n \ \dots \ \mathbf{Ab}_N).$$

Property 8.6.

$$(\lambda_1 \mathbf{b}_1 \ \lambda_2 \mathbf{b}_2 \ \dots \ \lambda_N \mathbf{b}_N) = \mathbf{B} \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_N \end{pmatrix}.$$

Let \mathbf{A} be an $N \times N$ matrix with N linearly independent eigenvectors, $\mathbf{x}_1, \dots, \mathbf{x}_N$ corresponding to eigenvalues $\lambda_1, \dots, \lambda_N$. Recall that the eigenvalues can be distinct since there are N linearly independent eigenvectors (matrix is of full rank) but its possible to have multiplicity in eigenvalues for full-rank matrices.

Definition 8.9. Define \mathbf{M} as a modal matrix for \mathbf{A} (columns are eigenvectors) and \mathbf{D} as a spectral matrix for \mathbf{A} (diagonal elements are eigenvalues), where

$$\mathbf{M} \triangleq (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N) \quad \text{and} \quad \mathbf{D} \triangleq \begin{pmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_N \end{pmatrix}.$$

Note that \mathbf{M} and \mathbf{D} are not unique (see p 181 [4]). We can show using properties above that

$$\begin{aligned} \mathbf{AM} &= \mathbf{A}(\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N) \\ &= (\mathbf{Ax}_1 \ \mathbf{Ax}_2 \ \dots \ \mathbf{Ax}_N) \\ &= (\lambda_1 \mathbf{x}_1 \ \lambda_2 \mathbf{x}_2 \ \dots \ \lambda_N \mathbf{x}_N) \quad \text{eigenvector statement} \\ &= (\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N) \mathbf{D} = \mathbf{MD}. \end{aligned}$$

Since the columns of \mathbf{M} are linearly independent, \mathbf{M} is full rank and \mathbf{M}^{-1} exists. Therefore,

$$\mathbf{D} = \mathbf{M}^{-1} \mathbf{AM},$$

showing that \mathbf{D} is similar to \mathbf{M} . Letting $\mathbf{P} = \mathbf{M}^{-1}$, it follows that

$$\mathbf{A} = \mathbf{P}^{-1} \mathbf{DP} = \mathbf{MDM}^{-1}$$

so \mathbf{A} is also similar to \mathbf{D} .

Therefore the square matrix \mathbf{A} of order N is diagonalizable only if it has full rank (there are N linearly independent eigenvectors.) Also the inverse of \mathbf{P} is the modal matrix of \mathbf{A} .

Example 8.14. Is $\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}$ diagonalizable?

Solution. We found above that the eigenvalues of \mathbf{A} are -1 and 5 . The eigenvectors are

$$\begin{pmatrix} 2 & 2 \\ 4 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}_{\lambda=-1} = \mathbf{0}$$

From the system, we find $x_1 = -x_2$ so that the first eigenvector is $\mathbf{x}_1 = (1 \ (-1))^t$. Similarly we can show that $\mathbf{x}_2 = (1 \ 2)^t$. Because the eigenvalues are distinct, the eigenvectors are linearly independent and therefore \mathbf{A} is diagonalizable.

Choosing $\mathbf{M} = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix}$ we find

$$\mathbf{D} = \mathbf{M}^{-1}\mathbf{A}\mathbf{M} = \frac{1}{3} \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 5 \end{pmatrix},$$

and choosing the alternative for \mathbf{M} (reverse the columns) we find $\mathbf{D} = \begin{pmatrix} 5 & 0 \\ 0 & -1 \end{pmatrix}$. So while it's true that neither \mathbf{M} nor \mathbf{D} are unique, the columns of \mathbf{M} (eigenvectors) correspond uniquely to the columns of \mathbf{D} (eigenvalues). So \mathbf{D} is unique for a given \mathbf{M} .

8.4.2 Functions of Matrices

If

$$\mathbf{D} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{pmatrix}$$

then

$$\mathbf{D}^n = \begin{pmatrix} \lambda_1^n & & & \\ & \lambda_2^n & & \\ & & \ddots & \\ & & & \lambda_N^n \end{pmatrix}, \quad \exp(\mathbf{D}) = \begin{pmatrix} \exp(\lambda_1) & & & \\ & \exp(\lambda_2) & & \\ & & \ddots & \\ & & & \exp(\lambda_N) \end{pmatrix}$$

$$\text{and } f_k(\mathbf{D}) = \begin{pmatrix} f_k(\lambda_1) & & & \\ & f_k(\lambda_2) & & \\ & & \ddots & \\ & & & f_k(\lambda_N) \end{pmatrix}$$

for f_k as a k -th degree polynomial.

Example 8.15. Find \mathbf{A}^n for the matrix defined in the last example.

Solution.

$$\mathbf{A}^n = \begin{pmatrix} 1 & 2 \\ 4 & 3 \end{pmatrix}^n = \mathbf{M}\mathbf{D}^n\mathbf{M}^{-1} = \begin{pmatrix} 1 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} (-1)^n & 0 \\ 0 & 5^n \end{pmatrix} \frac{1}{3} \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix} = \frac{1}{3} \begin{pmatrix} -2 + 5^n & 1 + 5^n \\ 2 + 2(5)^n & -1 + 2(5)^n \end{pmatrix}.$$

Property 8.7. If $f(z)$ and \mathbf{A} have the appropriate convergence properties (see p187 and p106 in [4]) then

$$f(\mathbf{A}) = \mathbf{M}f(\mathbf{D})\mathbf{M}^{-1} .$$

Definition 8.10. (See pp 1-3 in [2].)

- A function maps scalars to scalars. For example, $y(t) = 3t^2$.
- A functional maps functions to scalars. For example, $c = \int_a^b dt f(t)$.
- An integral transformation are functionals that depend on a continuous variable, and so map functions to functions. For example, $F(u) = \int dt f(t) \exp(-i2\pi ut)$.
- A linear vector space, \mathbb{L} is a set of vectors, x, y, z , for which specific rules of addition and multiplication apply. For example, the space is closed under addition: if $x \in \mathbb{L}$ and $y \in \mathbb{L}$, then $x + y \in \mathbb{L}$. Commutativity $x + y = y + x$, associativity $(x + y) + z = x + (y + z)$, scalar multiplication $\lambda x \in \mathbb{L}$, etc... The often encountered linear vector spaces are \mathbb{R}^3 , for example, (x_1, x_2, x_3) is a 3-vector of real numbers. Also \mathbb{C}^N is an N -vector of complex numbers.

8.4.3 Definitions and Properties

Eigenvectors and eigenvalues characterize important properties that help us work with matrices. (There are things known as generalized eigenvectors that expand the application of these methods to matrices that do not otherwise conform to the assumed properties. See [4], end of chapter 8.) They also help us model signal and noise in measurement signals. Numerical methods are almost always needed to actually do the job since the computations are usually complicated and tedious. Exceptions include real symmetric matrices and other matrices that are easier to work with. Nevertheless, to use MATLAB you need to understand the principles. That is our goal here.

A few properties...

Property 8.8. The magnitude of a vector, $\|\mathbf{x}\| = (\mathbf{x}, \mathbf{x})^{1/2}$, is defined as the \mathbb{L}_2 -norm (see below). A normalized vector (unit vector) $\hat{\mathbf{x}} \triangleq \mathbf{x}/\|\mathbf{x}\|$ such that $\|\hat{\mathbf{x}}\| = 1$.

Property 8.9. Vector norms of an N -dimensional vector space are defined as

$$\begin{aligned} \|\mathbf{x}\|_1 &= \sum_n |x_n| \\ \|\mathbf{x}\|_2 &= \left[\sum_n |x_n|^2 \right]^{1/2} \\ \|\mathbf{x}\|_p &= \left[\sum_n |x_n|^p \right]^{1/p} \\ \|\mathbf{x}\|_\infty &= \lim_{p \rightarrow \infty} \|\mathbf{x}\|_p = \max_n |x_n| \end{aligned}$$

Definition 8.11. Vectors \mathbf{x} and \mathbf{y} are orthogonal if $(\mathbf{x}, \mathbf{y}) = 0$.

Definition 8.12. A set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is orthonormal if they are orthogonal to each other and each is a unit vector. That is,

$$(\mathbf{x}_i, \mathbf{x}_j) = \delta_{ij} \quad \forall i, j = 1, 2, \dots, N,$$

where δ_{ij} is a Kronecker delta defined as (also defined on p. 41)

$$\delta_{ij} \triangleq \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}.$$

Definition 8.13. An orthonormal set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ is linearly independent. To see this note that

$$\begin{aligned} c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_N \mathbf{x}_N &= \mathbf{0} \\ (c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_N \mathbf{x}_N, \mathbf{x}_i) &= (\mathbf{0}, \mathbf{x}_i) \\ c_i \delta_{ii} &= 0, \end{aligned}$$

which is only true if $c_i = 0$. This can be done for each element such that $c_1 = c_2 = \dots = c_N = 0$.

8.4.4 Special Matrices

1.

$$\mathbf{B} = \begin{pmatrix} \mathbf{A}[1, 1] & \cdots & \mathbf{A}[1, q] \\ \vdots & \ddots & \vdots \\ \mathbf{A}[p, 1] & \cdots & \mathbf{A}[p, q] \end{pmatrix}$$

is a block matrix where each element is a matrix of size $M \times N$. While the block dimension of \mathbf{B} is $p \times q$, the scalar dimension is $pM \times qN$.

2. The sequence of complex sinusoids $\exp(i2\pi umT)$, is characterized by frequency u and sampling interval T , where $t = mT$. Let's create an $M \times 1$ column vector $\mathbf{w}_{M-1}(f)$ where

$$\mathbf{w}_{M-1}(f) = \begin{pmatrix} 1 \\ \exp(i2\pi uT) \\ \vdots \\ \exp(i2\pi u(M-1)T) \end{pmatrix} \quad \text{and} \quad \mathbf{x} = \begin{pmatrix} x[0] \\ x[1] \\ \vdots \\ x[M-1] \end{pmatrix}$$

This allows us to write the discrete-time Fourier transform for an M -sample data sequence as an inner product, $X(u) = T \mathbf{w}_{M-1}^\dagger \mathbf{x}$. More on this later...

3. A symmetric matrix has the property

$$\mathbf{A}^t = \mathbf{A} \quad \text{for} \quad \mathbf{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{12} & A_{22} & A_{23} & A_{24} \\ A_{13} & A_{23} & A_{33} & A_{34} \\ A_{14} & A_{24} & A_{34} & A_{44} \end{pmatrix}$$

4. The identity matrix \mathbf{I} and reflection matrix \mathbf{J} are useful.

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix} \quad \text{and} \quad \mathbf{J} = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{pmatrix}$$

While $\mathbf{AI} = \mathbf{IA} = \mathbf{A}$, \mathbf{J} reorders the columns or rows. Pre-multiplying by \mathbf{J} changes the order of rows and post-multiplying changes the order of columns

$$\mathbf{JA} = \begin{pmatrix} A_{41} & A_{42} & A_{43} & A_{44} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{11} & A_{12} & A_{13} & A_{14} \end{pmatrix} \quad \text{and} \quad \mathbf{AJ} = \begin{pmatrix} A_{14} & A_{13} & A_{12} & A_{11} \\ A_{24} & A_{23} & A_{22} & A_{21} \\ A_{34} & A_{33} & A_{32} & A_{31} \\ A_{44} & A_{43} & A_{42} & A_{41} \end{pmatrix}$$

5. A Toeplitz matrix has identical elements along all diagonals

$$\mathbf{A} = \begin{pmatrix} A[0] & A[-1] & A[-2] & A[-3] \\ A[1] & A[0] & A[-1] & A[-2] \\ A[2] & A[1] & A[0] & A[-1] \\ A[3] & A[2] & A[1] & A[0] \\ A[4] & A[3] & A[2] & A[1] \end{pmatrix} = \mathbf{JA}^t\mathbf{J}.$$

No need for the matrix to be square.

6. A Hankel matrix has identical cross-diagonal elements,

$$\mathbf{A} = \begin{pmatrix} A[-4] & A[-3] & A[-2] & A[-1] & A[0] \\ A[-3] & A[-2] & A[-1] & A[0] & A[1] \\ A[-2] & A[-1] & A[0] & A[1] & A[2] \\ A[-1] & A[0] & A[1] & A[2] & A[3] \end{pmatrix}.$$

Note that a square Hankel matrix \mathbf{A} is Hermitian $\mathbf{A} = \mathbf{A}^\dagger$. We'll use this for modeling exponential functions.

7. A circulant matrix is square with elements that wrap around in a circular fashion. A right circulant matrix is

$$\mathbf{A} = \begin{pmatrix} A[0] & A[1] & A[2] & A[3] \\ A[3] & A[0] & A[1] & A[2] \\ A[2] & A[3] & A[0] & A[1] \\ A[1] & A[2] & A[3] & A[0] \end{pmatrix}.$$

A right circulant matrix is a special case of a Toeplitz matrix, and a left circulant matrix is a special case of a Hankel matrix. Since FFT assume periodic data, circulant matrices are often used to model linear shift-invariant measurement systems, where rows of the matrix are the impulse response of the system. $\mathbf{g} = \mathbf{Hf} + \mathbf{e}$. A circulant matrix has eigenvalues and eigenvectors related to the discrete time Fourier series (DTFS) [6].

8. (From here to end of section, see section 3.7 in [6].) A matrix \mathbf{A} for which $\mathbf{A}^\dagger\mathbf{A} = \mathbf{AA}^\dagger$ is said to be *orthogonal*.

9. If, in addition, $\mathbf{A}^\dagger \mathbf{A} = \mathbf{A} \mathbf{A}^\dagger = \mathbf{I}$ then \mathbf{A} is said to be *orthonormal* or *unitary*. Also, because $\mathbf{A}^{-1} \mathbf{A} = \mathbf{I}$, then we have $\mathbf{A}^{-1} = \mathbf{A}^\dagger$ for unitary matrices. For unitary matrices $\det \mathbf{A} = 1$ and $|\lambda_n| = 1$.
10. If \mathbf{A} is nonsingular and Hermitian then

$$\mathbf{A} = \mathbf{M} \mathbf{D} \mathbf{M}^{-1} = \mathbf{A}^\dagger = \mathbf{M}^{-\dagger} \mathbf{D}^\dagger \mathbf{M}^\dagger .$$

For this to be true the inverse of the modal matrix (consisting of columns of eigenvectors, remember?) $\mathbf{M}^{-1} = \mathbf{M}^\dagger$ and the spectral matrix $\mathbf{D} = \mathbf{D}^\dagger$. This tells us that eigenvalues of Hermitian matrices are real. Also that, if the eigenvalues are all distinct, the eigenvectors form an orthonormal set. That is, since $\mathbf{M} \mathbf{M}^\dagger = \mathbf{I}$, \mathbf{M} is unitary. Therefore Hermitian matrices are diagonalized by the unitary transformation $\mathbf{A} = \mathbf{M} \mathbf{D} \mathbf{M}^\dagger$.

8.5 Singular Value Decomposition and the Pseudoinverse

8.5.1 SVD Analysis

(From 3.7.2 in [6].) We saw above that $N \times N$ (square) Hermitian matrices could be decomposed into unitary modal matrix (whose columns are orthonormal eigenvectors) and a spectral matrix whose diagonal elements are real. The concept of singular-value decomposition proposed in the 1970's extends the above result to an $M \times N$ complex-value matrix of rank k . The SVD theorem states that for matrix \mathbf{A} there exists positive real numbers $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k > 0$ (the singular values of \mathbf{A}), an $M \times M$ unitary matrix $\mathbf{U} = (\mathbf{u}_1 \dots \mathbf{u}_M)$ and an $N \times N$ unitary matrix $\mathbf{V} = (\mathbf{v}_1 \dots \mathbf{v}_N)$ such that

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\dagger = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\dagger ,$$

where

$$\mathbf{\Sigma} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} , \quad \mathbf{D} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k), \quad \text{and} \quad \mathbf{\Sigma} \text{ and } \mathbf{A} \text{ have size } M \times N .$$

Noting that $\mathbf{A}^\dagger = \mathbf{V} \mathbf{\Sigma}^\dagger \mathbf{U}^\dagger$, $\mathbf{U}^\dagger \mathbf{U} = \mathbf{I}$, and $\mathbf{V}^\dagger \mathbf{V} = \mathbf{I}$, then

$$\mathbf{A}^\dagger \mathbf{A} = \mathbf{V} (\mathbf{\Sigma}^\dagger \mathbf{\Sigma}) \mathbf{V}^\dagger \quad , \quad \mathbf{A} \mathbf{A}^\dagger = \mathbf{U} (\mathbf{\Sigma} \mathbf{\Sigma}^\dagger) \mathbf{U}^\dagger$$

and, from the i -th column of \mathbf{U} and \mathbf{V} ,

$$\mathbf{A}^\dagger \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i \quad , \quad \mathbf{A} \mathbf{A}^\dagger \mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$$

but only for $1 \leq i \leq k$. $\mathbf{\Sigma}^\dagger \mathbf{\Sigma}$ and $\mathbf{\Sigma} \mathbf{\Sigma}^\dagger$ are diagonal matrices of size $N \times N$ and $M \times M$, respectively, with diagonal elements σ_i^2 . The products $\mathbf{A}^\dagger \mathbf{A}$ and $\mathbf{A} \mathbf{A}^\dagger$ are Hermitian matrices of sizes $N \times N$ and $M \times M$. Therefore, the columns of \mathbf{U} are orthonormal vectors of $\mathbf{A} \mathbf{A}^\dagger$ and columns of \mathbf{V} are orthonormal vectors of $\mathbf{A}^\dagger \mathbf{A}$. Both share the same eigenvalues, σ_i^2 , for $1 \leq i \leq k$. Therefore, the singular values of matrix \mathbf{A} are the positive square roots of the nonzero eigenvalues of $\mathbf{A}^\dagger \mathbf{A}$ and $\mathbf{A} \mathbf{A}^\dagger$.

Also note that

$$\mathbf{A} \mathbf{V} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\dagger \mathbf{V} = \mathbf{U} \mathbf{\Sigma} \quad \text{and} \quad \mathbf{U}^\dagger \mathbf{A} = \mathbf{\Sigma} \mathbf{V}^\dagger \quad \text{or}$$

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i \quad \text{and} \quad \mathbf{A}^\dagger\mathbf{u}_i = \sigma_i\mathbf{v}_i \quad \text{for } 1 \leq i \leq k.$$

This shows that vectors \mathbf{u}_i and \mathbf{v}_i are related through their common singular value σ_i .

Example 8.16. We can use SVD to decompose a waveform into its signal and noise components for the purpose of filtering the noise. For example, in the Fig 62, we have a sinusoidal voltage signal $y(t)$ of amplitude 1.0 V and frequency 4.167 Hz that is added to random noise. The result is $g(t)$. As shown in the MATLAB code, we form a matrix \mathbf{G} where each row contains the same signal, $y(t)$, but an independent noise realization $e(t)$. In the upper right corner of the figure, waveform vector 50 out of 100 is plotted. If we then plot the diagonal of matrix \mathbf{D} computed using SVD, we see in the lower left section of Fig 62 that σ_1 is more than three times larger than the other singular values. Of course, that makes sense since the pure tone signal $y(t)$ has more energy than other components of the spectrally white noise. We then set to zero all the singular values except σ_1 and reconstitute the waveform matrix, now called \mathbf{GG} . Plotting the 50th filtered waveform (lower right corner of Fig 62. shows that we have reduced noise without significant signal distortion.

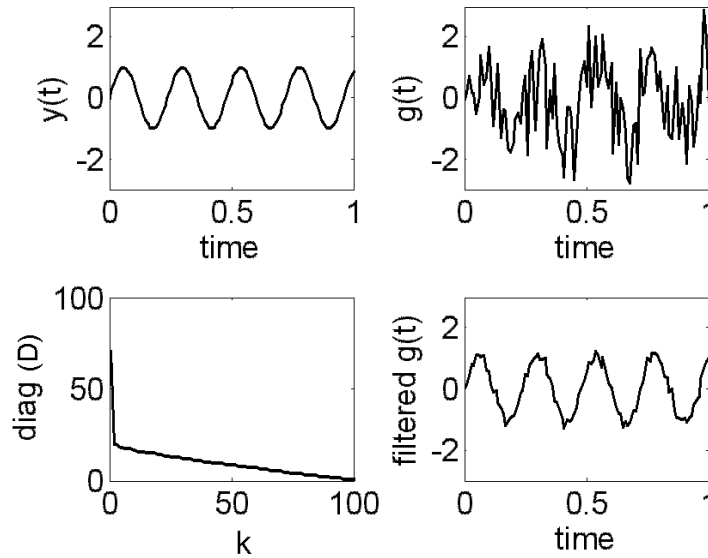


Figure 62: Example of SVD filtering.

```
t=0:0.01:1; y=sin(2*pi*t/0.24); subplot(2,2,1);plot(t,y);
%pause
for i=1:length(t);
    Y(i,:)=y;
end E=randn(size(Y)); G=Y+E; subplot(2,2,2);plot(t,G(50,:))
```

```
%pause
[U D V]=svd(G); subplot(2,2,3);plot(diag(D));
%pause
DD=zeros(size(D)); DD(1,:)=D(1,:); GG=U*DD*V';
subplot(2,2,4);plot(t,GG(50,:));
```

8.5.2 Moore-Penrose Pseudoinverse

(This is the simplistic version from [6] p77. There is a more comprehensive discussion in [2] on pp38-48.)

The Moore-Penrose pseudoinverse \mathbf{A}^+ of the $M \times N$ matrix \mathbf{A} of rank k can be defined in terms of the SVD matrix components via

$$\mathbf{A}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^\dagger = \sum_{i=1}^k \sigma_i^{-1} \mathbf{v}_i \mathbf{u}_i^\dagger, \quad \text{where} \quad \mathbf{\Sigma}^+ = \begin{pmatrix} \mathbf{D}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

The pseudoinverse may be the only solution when \mathbf{A} is not square or of full rank. It provides the minimum-norm least-squares solution $\mathbf{x} = \mathbf{A}^+\mathbf{b}$ to the problem of finding the $N \times 1$ vector \mathbf{x} that simultaneously minimizes the mean-squared error (\mathbb{L}_2 norm) $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ for that $M \times 1$ vector and the solution vector length $\|\mathbf{x}\|_2$. The minimum-length constraint ensure a unique solution.

If \mathbf{A} is $N \times N$ and $k = N$ (full rank), then $\mathbf{A}^+ = \mathbf{A}^{-1}$. However computation of \mathbf{A}^+ is expensive, so it should be used only when \mathbf{A}^{-1} isn't defined.

If $M > N$ and the rank(\mathbf{A}) = $k = N$, then $\mathbf{A}^+ = (\mathbf{A}^\dagger \mathbf{A})^{-1} \mathbf{A}^\dagger$ so that $\mathbf{x} = (\mathbf{A}^\dagger \mathbf{A})^{-1} \mathbf{A}^\dagger \mathbf{b}$.

If $N > M$ and the rank(\mathbf{A}) = M , then $\mathbf{A}^+ = \mathbf{A}^\dagger (\mathbf{A} \mathbf{A}^\dagger)^{-1}$ and $\mathbf{x} = \mathbf{A}^\dagger (\mathbf{A} \mathbf{A}^\dagger)^{-1} \mathbf{b}$.

Example 8.17. *Let's look at the example from MATLAB when you look up `pinv` in the doc request. Click on MATLAB and then find `pinv` in the index.*

A magic matrix is square with the properties that the sum of the rows, columns and diagonals all equal the same number. The characteristic sum for a magic square of order N is `sum(1:N^2)/N`, provided $N \geq 3$. When $n = 3$, the characteristic sum is 15.

The problem is to solve for \mathbf{x} the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$.

The MATLAB example selects an 8×8 magic matrix with characteristic sum 260. It then cuts off the right-most two columns to produce an 8×6 matrix for \mathbf{A} and uses direct and indirect methods to solve for \mathbf{x} . rank(\mathbf{A}) = 3.

```
A = magic(8); A=A(:,1:6); b=260*ones(8,1); x=pinv(A)*b
```

yields the pseudoinverse result

```
x =
    1.1538
    1.4615
    1.3846
    1.3846
    1.4615
    1.1538
```

whereas the direct result of pre-multiplying by the inverse of \mathbf{A} gives

```
y=A\b;
```

```
Warning: Rank deficient, rank = 3  tol = 1.8829e-013.
```

```
y =
    4.0000
    5.0000
         0
         0
         0
   -1.0000
```

Now you need to decide which is preferable. Both solutions are exact in the sense that $\text{norm}(\mathbf{A}*\mathbf{x}-\mathbf{b})$ and $\text{norm}(\mathbf{A}*\mathbf{y}-\mathbf{b})$ are on the order of the roundoff error given by the precision of the machine. However the \mathbf{x} solution is the minimum-norm least squares solution $\text{norm}(\mathbf{x}) = 3.2817$.

Note that $\text{norm}([1 \ 1 \ 1 \ 1 \ 1]) = \text{sqrt}(6) = 2.4495$. However, the \mathbf{y} solution has the minimum number of non-zero components (3) which equals the rank of \mathbf{A} . You need to decide if you want the solution components to be closest to the 8×8 solution (that would be \mathbf{x}) or having non-zero components equal to the rank of the matrix (that would be \mathbf{y}).

8.6 Condition Number and Computational Errors

8.6.1 Condition Number

The condition number of a matrix measures the sensitivity of the solution of a system of linear equations to errors in the data. It gives an indication of the accuracy of the results from matrix inversion and the linear equation solution. In numerical analysis, the condition number associated with a problem is a measure of that problem's amenability to digital computation, that is, how numerically *well-posed* the problem is. A problem with a low condition number is said to be *well-conditioned*, while a problem with a high condition number is said to be *ill-conditioned*.

The condition number associated with the linear equation $\mathbf{A}\mathbf{x} = \mathbf{b}$ gives a bound on how inaccurate the solution \mathbf{x} will be after approximate solution. This bound excludes the effects of round-off error; conditioning is a property of the matrix, not the algorithm or floating point accuracy of the computer used to solve the corresponding system. (See http://en.wikipedia.org/wiki/Condition_number).

The condition number of \mathbf{A} is found from the ratio of \mathbb{L}_2 norms,

$$C(\mathbf{A}) = \|\mathbf{A}^{-1}\|_2 \|\mathbf{A}\|_2 = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})},$$

where $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ are the maximum and minimum singular values.

If \mathbf{A} is a normal matrix ($\mathbf{A}^\dagger \mathbf{A} = \mathbf{A} \mathbf{A}^\dagger$), then we also have

$$C(\mathbf{A}) = \left| \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \right|,$$

where λ_{\max} and λ_{\min} are the max and min eigenvalues of \mathbf{A} as measured from the modulus of the complex values (`abs()` in MATLAB).

8.6.2 A Little about Error Bounds

Bai [7], p10,11, shows that if we have a scalar function $f(x)$ and we use an algorithm to estimate that function, $\hat{f}(x)$, we will often want to estimate the upper bound on the error $\hat{f}(x) - f(x)$.

Assume $\hat{f}(x) = f(x + e)$ for some small value of e . If the algorithm obtains exactly the right solution if we perturb the argument by the amount e , then the algorithm is *numerically stable*. The error is

$$\text{error} = \hat{f}(x) - f(x) = f(x + e) - f(x) \approx f'(x) \cdot e .$$

where $f'(x)$ is the derivative wrt x . The upper bound on error is

$$|\text{error}| \leq |f'(x)| \cdot |e| ,$$

where $|f'(x)|$ is the *condition number* that depends only on f and x and not the algorithm used to compute them. $|e|$ is the *backward error*.

We can also have that x is a matrix $x \rightarrow \mathbf{A}$ and $e \rightarrow \mathbf{E}$ so that $x + e \rightarrow \mathbf{A} + \mathbf{E}$ is a perturbed matrix where the elements of \mathbf{E} are relatively “small”. Then $f(x + e)$ could be the eigenvalues, eigenvectors, or other functions of the argument matrix that we might want to estimate. The value $|e|$ is replaced by $\text{norm}(\mathbf{E}) = \|\mathbf{E}\|_2$. There are several ways of estimating $\|\mathbf{E}\|_2$ that we won't go into here. (Perhaps Professor Zhong will.) One estimate is from the product of the precision of the computer ϵ times the norm of the input matrix $\|\mathbf{A}\|_2$ [7].

From the MATLAB doc: The precision of the machine can be estimated in MATLAB using the function `eps`. `d = eps(X)` is the positive distance from `abs(X)` to the next larger in magnitude floating point number of the same precision as `X`. `X` may be either double precision or single precision.

Examples from MATLAB :

double precision

`eps(1/2) = 2^(-53)`

`eps(1) = 2^(-52)`

`eps(2) = 2^(-51)`

`eps(realmax) = 2^971`

`eps(0) = 2^(-1074)`

`eps(Inf) = NaN`

`eps(NaN) = NaN`

single precision

`eps(single(1/2)) = 2^(-24)`

`eps(single(1)) = 2^(-23)`

```
eps(single(2)) = 2^(-22)
```

```
eps(realmax('single')) = 2^104
```

```
eps(single(0)) = 2^(-149)
```

8.6.3 MATLAB Notes

Applications for solving systems of equations. Should you use $x = \text{inv}(A)*b$ or $x = A \setminus b$?

From the MATLAB doc files, the following test was performed on a 300 MHz laptop.

```
N = 500;
```

```
Q = orth(randn(N,N));
```

“The operation $B = \text{orth}(A)$ returns an orthonormal basis for the range of A . The columns of B span the same space as the columns of A , and the columns of B are orthogonal, so that $B'*B = \text{eye}(\text{rank}(A))$ and $\text{eye}(N) = I$ of size $N \times N$. The number of columns of B is the rank of A . Also $\text{randn}(N,N)$ generates a square matrix of random numbers (standard normal distribution, zero mean and unit variance).”

```
d = logspace(0,-10,N);
```

“generates N logarithmically spaced points between 10^0 and 10^{-10} .”

```
A = Q*diag(d)*Q';
```

```
x = randn(n,1);
```

```
b = A*x;
```

```
tic, y = inv(A)*b; toc
```

```
err = norm(y-x)
```

```
res = norm(A*y-b)
```

```
produce elapsed_time =  
1.4320
```

```
err =  
7.3260e-006
```

```
res =  
4.7511e-007
```

while the statements

```
tic, z = A\b, toc

err = norm(z-x)

res = norm(A*z-b)

produce elapsed_time =
    0.6410
err =
    7.1209e-006
res =
    4.4509e-015
```

It takes almost 2.5 times as long to compute the solution with $y = \text{inv}(A)*b$ as with $z = A\b$. Both produce computed solutions with about the same error, $1.e-6$, reflecting the condition number of the matrix. But the size of the residuals, obtained by plugging the computed solution back into the original equations, differs by several orders of magnitude. The direct solution produces residuals on the order of the machine accuracy, even though the system is badly conditioned. The behavior of this example is typical. Using $A\b$ instead of $\text{inv}(A)*b$ is two to three times as fast and produces residuals on the order of machine accuracy relative to the magnitude of the data.

8.7 Pr

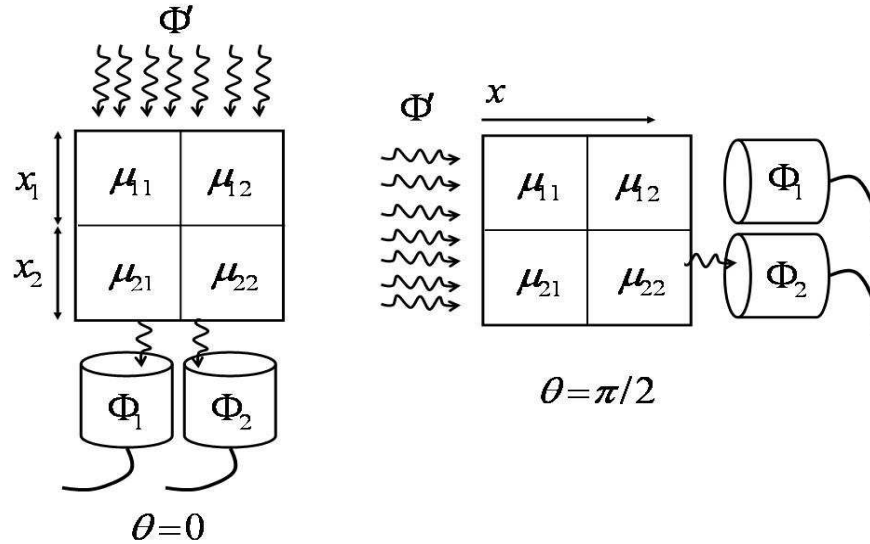


Figure 63: Simple CT experiment

1. The figure shows a very simple x-ray computed tomography (CT) experiment where data are acquired over four volume elements (voxels) and just two viewing angles, $\theta = 0$ and $\pi/2$. A photon flux Φ' is incident on a cube of tissue from above. The x-ray photons that make it through the cube are detected by one of two detectors that record the transmitted photon fluxes Φ_1 and Φ_2 . For a moment, let's focus on the $\theta = 0$ experiment on the left. Those passing through the left side of the cube are absorbed according to the attenuation coefficients $\mu_{11} + \mu_{21}$ to give us $\Phi_{1,0}$, where the subscripts 1,0 refer to detector 1 at scanning angle 0. The equation is given by Beer's law

$$\Phi_{1,0} = \Phi' \exp(-\mu_{11} x_1) \exp(-\mu_{21} x_2) = \Phi' \exp\left(-\sum_{i=1}^2 \mu_{i1} x_i\right).$$

Our goal is to estimate the μ 's. It is convenient to linearize the equation using

$$\varphi_{1,0} = -\ln\left[\frac{\Phi_{1,0}}{\Phi'}\right] = \sum_{i=1}^2 \mu_{i1} x_i.$$

We complete the generalization for the $k = 1, 2$ columns to obtain the system of equations

$$\varphi_{k,0} = \sum_{i=1}^2 \mu_{ik} x_i,$$

or in matrix form $\varphi_0 = \mathbf{M}_0 \mathbf{x}$.

This describes the entire data set of projections for acquisition at $\theta = 0$. φ_0 and \mathbf{x} are 2×1 column vectors and \mathbf{M}_0 is a 2×2 matrix. Note that the arrangement of elements in \mathbf{M} depend

on the scanning angle, so we need the subscript.

(a) Give the elements of matrix \mathbf{M}_0 .

(b) Using the derivation pattern above and the geometry in the figure, find the system of equations for $\theta = \pi/2$. Express $\mathbf{M}_{\pi/2}$ in terms of \mathbf{M}_0 . (Note: $\theta = \pi/2$ in the frame of the measurement system. Also note that $x_1 = x_2$ so that $(x_1 \ x_2) = (x_2 \ x_1)$).

(c) This situation brings up a more general question about rotations of data in a plane. Let the coordinate axis for the object be co-located with the rotation axis as shown below. If a pixel of interest is located at $\mathbf{x} = (a \ b)^t$, then where is it located in the coordinate system \mathbf{x}' ?

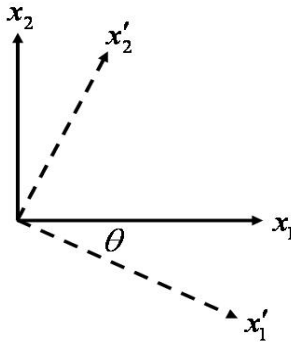


Figure 64: Coordinate rotation.

- When designing any diagnostic system, you can evaluate its “performance” (the ability of the system to do its job) if you know the probability density functions (pdfs) for the data vector \mathbf{g} under two conditions. Let’s set the conditions to be hypothesis H_0 (the patient is well) and hypothesis H_1 (the patient is sick). We haven’t reviewed probability theory yet, I know, but let’s see if you can compute the log-likelihood function anyway. Let \mathbf{g} be an $M \times 1$ column vector of data acquired from the measurement device under evaluation. Also let $p(\mathbf{g}|H_1)$ be the probability of obtaining data vector \mathbf{g} for a patient that is sick. Assume the data follows a Gaussian distribution with zero mean and covariance matrix \mathbf{K} of order $M \times M$. The conditional pdfs under the two hypotheses are given by

$$p(\mathbf{g}|H_i) = [(2\pi)^M \det \mathbf{K}_i]^{-1/2} \exp\left(-\frac{1}{2}\mathbf{g}^t \mathbf{K}_i^{-1} \mathbf{g}\right).$$

Let $\lambda(\mathbf{g})$ be the log-likelihood function given by

$$\lambda(\mathbf{g}) = \ln\left(\frac{p(\mathbf{g}|H_1)}{p(\mathbf{g}|H_0)}\right),$$

which is a scalar value that can be compared to a threshold when we wish to evaluate the system's performance. All I want you to do is calculate λ and reduce it to its simplest form.

3. Matrix \mathbf{Q} is composed of complex values,

$$\mathbf{Q} = \begin{pmatrix} G_{00} & G_{01}^* \\ G_{10} & G_{11} \end{pmatrix}. \quad \text{Also } \gamma_{01} = \frac{G_{01}}{\sqrt{G_{00}G_{11}}} \quad \text{and } G_{01} = G_{10}.$$

However G_{00} and G_{11} are real. Find \mathbf{Q}^{-1} only in terms of G_{00} , G_{11} , γ_{01} and associated conjugates.

4. Which of the following are eigenfunction of the linear system given by $y[n] = \sum_{k=-\infty}^{\infty} a[k]x[n-k]$?
- (a) $\exp(i\omega n) + \exp(i2\omega n)$, (b) 5^n , (c) $5^n \exp(i2\omega n)$.
5. Find a set of linearly independent eigenvectors for

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 3 & 2 \\ 4 & 2 & 3 \end{pmatrix}.$$

Do so by finding the eigenvalues, determining the number of linearly independent eigenvectors for each, and then compute the eigenvectors.

6. Show that if two matrices are similar they have the same characteristic equation. (This is to exercise your matrix algebra skills.)
7. Show that $\mathbf{y}\mathbf{x}^T\mathbf{z} = (\mathbf{x}, \mathbf{z})\mathbf{y}$.
8. MATLAB Exercise.

```
clear all;

load penny; whos

imagesc(P);axis square; colormap(gray);

q=del2(P);imagesc(q);axis square; axis off
```

Describe what this sequence does.

9. MATLAB Exercise. For this next demo, go to my webpage and grab the jpeg image 100_0121.jpg, where you will find a handsome professor and his gorgeous granddaughter. If you can't grab that file, then use another color jpeg image file.

```
clear all;

x=imread('W:\temp\100_0121.jpg');
```

```

whos

imshow(x);

y=double(x);figure;

subplot(3,1,1);imagesc(x(:,:,1));axis image;axis off;
subplot(3,1,2);imagesc(x(:,:,2));axis image;axis off;
subplot(3,1,3);imagesc(x(:,:,3));axis image;axis off;

colormap(gray)

```

Describe these four images? Why is the original jpeg file a 3-D matrix?

10. Decomposing matrix \mathbf{A} using SVD we have $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$ where we assume $N > M$. (a) Compute the Moore-Penrose pseudoinverse \mathbf{A}^+ in terms of its components. (b) With the result above, show that $\mathbf{A}\mathbf{A}^+ = \mathbf{I}$.
11. MATLAB Exercise. Examine the example for `pinv(A)` in the MATLAB documentation. This is the one where \mathbf{A} is a magic matrix of order 8 truncated to 8×6 . Since \mathbf{A} is not square and not full rank, \mathbf{A}^{-1} does not exist. Use four methods for estimating the matrix inverse using MATLAB. Then multiply \mathbf{A} by the four estimates of its inverse and display the results in a 2×2 matrix of images. Perhaps you might use `subimage(2,2,1); imagesc(x); axis square;` or similar commands. If the inverse estimates are perfect, you should see an image of \mathbf{I} ; a diagonal stripe of “1’s” in a field of “0’s”. Give a short explanation of which method works best and under what conditions.
12. MATLAB Exercise. Later in the course, when we examine linear systems analysis, we’ll run into a convolution operator. We’ll find it pretty handy for explaining how many instruments distort information in recorded data. But for now, let’s examine how to use matrix math to compute a convolution. The convolution equation for discretely sampled data is

$$g[m] = \sum_{n=-\infty}^{\infty} h[m-n] f[n].$$

This tells us that the m -th sample of data measured by our instrument $g[m]$ depends on the properties of the object function, e.g., the body, that we hope to measure f as well as properties of the instrument via h . Let f be a square wave,

$$f[n] = \begin{cases} f_0; & 401 \leq n \leq 600 \\ 0; & \text{otherwise} \end{cases} \quad \text{for } n = 1, 2, \dots, 1000.$$

To get you started in MATLAB, try `n=1:1000;f=zeros(size(n));f(401:600)=1;plot(n,f)`. Then, for the same time range, $n = 1, 2, \dots, 1000$, choose

$$h[n] = \begin{cases} 1/N; & N_0 - N/2 + 1 \leq n \leq N_0 + N/2 \\ 0; & \text{otherwise} \end{cases},$$

and let $N_0 = 500$ and $N = 50$.

(a) Generate n , f and h and convolve them using `conv(h,f)`. Select the center 1000 points

and plot them.

(b) Create a 1000×1000 circulant matrix, \mathbf{H} , where each row is a shifted and wrapped version of h . Then matrix multiply \mathbf{H} and \mathbf{f} to get \mathbf{g} . Is it the same as in part a?

(c) Repeat parts (a) and (b) for $N = 200$.

Plot all the results neatly in a matrix of plots and hand that in.

9 APPENDIX B: Power Spectrum and Autocorrelation Functions. Sec 2.5 [9]

Real-life signals have finite length and energy and can be sampled in time or space. This is why we have lots of tools available to us for Fourier analysis (Fig 9). Pick the appropriate tool for the job. In all cases, Fourier analysis is a decomposition of waveforms into sinusoids (complex exponentials). When the signals are bandlimited, their energy is concentrated into a finite band of frequencies. It is therefore reasonable to assume the waveforms are nearly periodic²⁴ over time period T_0 , so that $f(t) = f(t + nT_0)$ (See Fig 9). The *Fourier series* decomposition of $f(t)$ gives

$$f(t) = \frac{1}{T_0} \sum_{k=-\infty}^{\infty} F_k \exp(i2\pi kt/T_0),$$

where the scalar Fourier coefficients F_k are

$$F_k = \int_{-T_0/2}^{T_0/2} dt f(t) \exp(-i2\pi kt/T_0). \tag{128}$$

The above two equations hold as long as the signal has finite time-averaged power:

$$\frac{1}{T_0} \int_{-T_0/2}^{T_0/2} dt |f(t)|^2 < \infty.$$

Notice that when $T_0 \rightarrow \infty$, then $1/T_0 \rightarrow du$, $k/T_0 \rightarrow u$ in the Fourier series equations and Eq (128) reduces to the forward CT-FT result of Eq (90). If the average power is finite, I can switch back and forth between FS and CT-FT as needed provided I can make the assumption of periodicity when using the Fourier series.

In the limit that $T_0 \rightarrow \infty$, it is clear that the function $f(t)/\sqrt{T_0}$ has a Fourier transform, call it F'_k . Therefore let's rewrite the Fourier series equation pair more symmetrically [9],

$$\begin{aligned} f(t) &= \frac{1}{\sqrt{T_0}} \sum_{k=-\infty}^{\infty} F'_k \exp(i2\pi kt/T_0) \\ F'_k &= \frac{1}{\sqrt{T_0}} \int_{-T_0/2}^{T_0/2} dt f(t) \exp(-i2\pi kt/T_0). \end{aligned}$$

While the equations are now symmetric, they look strange and are cumbersome. Nevertheless, we proceed by noting the quantity $F'_k/\sqrt{T_0}$ is infinitesimally small in the limit and of order \sqrt{du} , which depends on frequency and is a function of $f(t)$. We use the notation

$$\lim_{T_0 \rightarrow \infty} \frac{F'_k}{\sqrt{T_0}} \triangleq F(u, du).$$

Now we can write (note dt is missing in following because $F(u, du)$ contains the differential)

$$\begin{aligned} f(t) &= \int_{-\infty}^{\infty} F(u, du) \exp(i2\pi ut) \\ F(u, du) &= \lim_{T_0 \rightarrow \infty} \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} dt f(t) \exp(-i2\pi ut). \end{aligned} \tag{129}$$

²⁴Signals that are limited in duration in one domain are of infinite duration in the other.

We see that $f(t)$ is a linear combination of complex amplitudes $F(u, du)$ and phasors $\exp(-i2\pi ut)$. The average power in each frequency component is the product of $F(u, du) \exp(-i2\pi ut)$ with its conjugate, which equals $|F(u, du)|^2$.

Since signal and noise waveforms from biological sources are often random processes, we need to ‘average’ the power in each frequency channel to characterize properties (ensemble, spatial, temporal averages or some combination). Define

$$dP'_F(u) = \mathcal{E}\{|F(u, du)|^2\} = S_F(u) du \quad (130)$$

as the increment of power in random process F at frequency u by assuming a WSS process so that time averages and ensemble averages are equal.²⁵ $S_F(u)$ is the *power spectral density* of waveform $f(t)$ as computed from an ensemble of time-series realizations of the random process F . The power spectral density function $S_F(u)$ and ensemble correlation function $\phi_F(\tau)$ are Fourier transform pairs.

From Eq 39, for WSS waveforms $f(t)$, and keeping in mind that $F(u, du)$ are infinitesimals of order du , we have the autocorrelation function

$$\begin{aligned} \phi_F(\tau) &= \mathcal{E}\{f(t)f^*(t-\tau)\} = \mathcal{E}\left\{\int_{-\infty}^{\infty} F(u, du) \exp(i2\pi u t) \int_{-\infty}^{\infty} F^*(u', du') \exp(-i2\pi u'(t-\tau))\right\} \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathcal{E}\{F(u, du)F^*(u', du')\} \exp(i2\pi u'\tau) \exp(i2\pi(u-u')t) \\ &= \int_{-\infty}^{\infty} \mathcal{E}\{|F(u, du)|^2\} \exp(i2\pi u\tau), \quad \text{since } \mathcal{E}\{F(u, du)F^*(u', du')\} = \mathcal{E}\{|F(u, du)|^2\} \delta(u-u'), \\ &= \int_{-\infty}^{\infty} du S_F(u) \exp(i2\pi u\tau) \quad \text{From Eq (130)}. \end{aligned} \quad (131)$$

The fact that Fourier frequency components are uncorrelated is very handy. The bottom line is that the autocorrelation function and power spectral density function are Fourier transform pairs. As you might expect,

$$\begin{aligned} S_F(u) &= \int_{-\infty}^{\infty} d\tau \phi_F(\tau) \exp(-i2\pi u\tau) \quad \text{and} \\ \phi_F(\tau) &= \int_{-\infty}^{\infty} du S_F(u) \exp(i2\pi u\tau). \end{aligned}$$

This is the result of the *Wiener-Khintchin theorem* that is proven in many standard texts.

Notice that from Eq (131) at $\tau = 0$ we have

$$\phi_F(0) = \mathcal{E}\{|f^2(t)|\} = \text{var}(f) + \mu_f^2 = \int_{-\infty}^{\infty} du S_F(u).$$

So it seems that the power spectrum may be thought of as a spectrum of the mean-squared value of $f(t)$. If the process has zero mean, then $S_F(u)$ is the *spectrum of variance*.

²⁵One way to structure experimental data for this analysis is to consider an ensemble of time series to be an $N \times M$ matrix of data samples \mathbf{F} (object or measurement data). The rows are individual time series written as $1 \times M$ row vectors \mathbf{f}_i^t . Columns contain the N ensemble samples from any one instant of time. Note that the matrix \mathbf{F} , the random process representation F , and the Fourier coefficients F_k are all different. Sorry, the notation is cumbersome.

Lots and lots of books discuss power spectral estimation. One method is to compute the auto-correlation function and take the Fourier transform. Sometimes that is the best method. Another way more often used is to estimate $S_F(u)$ directly from a group of time series each of duration T_0 . Assume we have N data vectors of length M . The frequency increment is now Δu and therefore the frequency value is $k\Delta u$. From Eq (130), and keeping in mind our WSS assumption,

$$S_F(k\Delta u) = \frac{\mathcal{E}\{|F(u, du)|^2\}}{\Delta u} = \frac{1}{N} \sum_{n=1}^N \frac{(|F_n(k\Delta u)|/T)^2}{\Delta u} = \frac{1}{NT_0} \sum_{n=1}^N |F_n(k\Delta u)|^2 \quad (132)$$

since $\Delta u = 1/T_0$. So you can take two approaches. One is to record M points of data per time series and repeat the experiment N times. Then take the DFT of each time series and average them. Another approach is to record one time series with NM data points. If the process is ergodic, the time series can be divided into M -point segments, the DFT of each segment is computed and the results averaged. The nice this about using the equation above is that the error in the frequency components is reduced by averaging more waveforms (increasing N). Its not obvious, but adding more points to the waveform undergoing a Fourier transform only adds more points to the frequency spectrum. The uncertainty in estimating frequency components is unaffected by transforming more points. Only by averaging transforms from independent waveforms can the coefficient errors be reduced, but that may occur only at the cost of frequency resolution. (Future years, add part from [6] on spectral errors.)

References

- [1] R Shiavi, *Introduction to Applied Statistical Signal Analysis: Guide to Biomedical and Electrical Engineering Applications*, Elsevier, Amsterdam, 2007.
- [2] HH Barrett, KJ Myers, *Foundations of Image Science*, Wiley-Interscience, Hoboken NJ, 2004.
- [3] RD Evans *The Atomic Nucleus*, McGraw-Hill, New York, 1955.
- [4] R Bronson, *Matrix Methods: An Introduction*, Academic Press, New York, 1969.
- [5] RC Gonzalez, RE Woods, SL Eddins, *Digital Image Processing Using MATLAB*, Pearson-Prentice-Hall, Upper Saddle River NJ, 2004.
- [6] SL Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [7] Z Bai, J Demmel, J Dongarra, A Ruhe, H van der Vorst, *Templates for the solution of Algebraic Eigenvalue Problems*, SIAM, Philadelphia, PA, 2000.
- [8] S Ross, *A First Course in Probability*, Macmillan Publ Co, NY, 1976.
- [9] RN McDonough, AD Whalen, *Detection of Signals in Noise*, 2/e, Academic Press, San Diego, 1995.
- [10] JA Gubner, *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge Univ Press, Cambridge UK, 2006.
- [11] H Urkowitz *Signal Theory and Random Processes*, Artech House, Inc., Norwood MA, 1983.
- [12] AV Oppenheim, RW Schaffer, JR Buck, *Discrete-Time Signal Processing*, 2/e Prentice-Hall, Upper Saddle River NJ, 1999.
- [13] HH Barrett, W Swindell, *Radiological Imaging: The Theory of Image Formation, Detection, and Processing*, Academic Press, New York, 1981.
- [14] E Parzen, *Modern Probability Theory and Its Applications*, Wiley & Sons, New York, 1960.
- [15] A Macovski, *Medical Imaging Systems*, Prentice-Hall, Upper Saddle River NJ, 1983.
- [16] EN Bruce, *Biomedical Signal Processing and Signal Modeling*, Wiley & Sons, New York, 2001.
- [17] RN Bracewell, *The Fourier Transform and Its Applications*, 2/e, McGraw-Hill, New York, 1978.
- [18] W Simon, *Mathematical Techniques for Biology and Medicine*, Dover, New York, 1986.
- [19] NW Tschoegl, *The Phenomenological Theory of Linear Viscoelastic Behavior: An Introduction*, Springer-Verlag, Heidelberg, 1989.
- [20] SJ Bornheimer, MR Maurya, MG Farquhar, S Subramaniam, "Computational modeling reveals how interplay between components of a GTPase-cycle module regulates signal transduction," *Proc Natl Acad Sci U S A*. Nov 9;101(45):15899-904. 2004

- [21] M Buchwald, RJ Britten, "Incorporation of ribonucleic acid bases into the metabolic pool and RNA of E Coli," *Biophys J*, 3:155-166, 1963.
- [22] IS Gradshteyn, IM Ryzhik, *Table of Integrals, Series, and Products*, 5/e, A Jeffrey, ed., Academic Press, San Diego, 1994.
- [23] M Zamir, *The Physics of Pulsatile Flow*, Springer-Verlab, New York, 2000.
- [24] CJ Hartley, LH Michael, ML Entman, "Noninvasive measurement of ascending aortic blood velocity in mice," *J Am J Physiol* 268:H499-H505, 1995.
- [25] PR Bevington, DK Robinson, *Data Reduction and Error Analysis for the Phycial Sciences*, 2/e, WCB/McGraw-Hill, Boston, 1992.
- [26] JS Bendat, AG Piersol, *Random Data: Analysis and Measurement Procedures*, 2/e, Wiley-Interscience, New York, 1986.

9.1 Problems

1. MATLAB Exercise. Create a 3 s square pulse of amplitude 1.0 centered on a 10 s time axis and sampled at a rate of 100 Hz. Call it $f(t)$ or, more accurately, \mathbf{f} .

Next, generate a Gaussian pulse of **unit area**, centered at 5 s, and having a width parameter $\sigma = 0.1$ s for the same 10 s time axis. Call that $h(t)$ or \mathbf{h} . Convolve the two time-domain functions in MATLAB and plot all three in the first 3 slots of a 4 by 1 subplot. Pay attention to the amplitudes to be sure they are correctly scaled and all of the time axes should be over 0 to 10 s. (That's right, you need to keep track of all the dumb little constants so the peak amplitudes are correct.)

Use Fourier techniques in MATLAB to “deconvolve” \mathbf{h} from \mathbf{g} to obtain $\hat{\mathbf{f}}$, an estimate of the original object function \mathbf{f} . That is,

$$\hat{f}(t) = \mathcal{F}^{-1} \left\{ \frac{G(u)}{H(u)} \right\} .$$

I'll tell you right now, this will not work as written above without you paying attention to very important details, including computational errors, assumptions in MATLAB about the frequency space, etc. Once you do the best you can, then plot the result in the fourth slot of the subplot. I want to see the plots and your brief explanation of what efforts were required to obtain the answer you show.

2. Assume a linear system of the form

$$g[\ell, m] = \int d\mathbf{x} h(\mathbf{x}, [\ell, m]) f(\mathbf{x}) + e[\ell, m] .$$

This is a continuous-to-discrete linear transformation from an object (f is a random variable) that is a function of position in object space, \mathbf{x} , into data, also a random variable g , that is a function of position in data space at integer coordinates $[\ell, m]$. Note that $0 \leq \ell \leq L$ and $0 \leq m \leq M$. The mapping from object to data space is through a deterministic system impulse response h that may not be shift invariant. The *sensitivity* of the system is sometimes quantified by the signal-to-noise ratio (SNR), given as the ratio of signal energy to noise energy,

$$\text{SNR} = \mathcal{E} \left\{ \sum_{\ell, m} \left[\mathcal{E}\{g[\ell, m]\}_{e|f} \right]^2 \right\}_f / \mathcal{E} \left\{ \sum_{\ell, m} e^2[\ell, m] \right\}_e ,$$

The notation $\mathcal{E}\{\mathbf{g}\}_{e|f}$ denotes ensemble average of data g over the noise process e for a fixed realization of the random object f .

Assuming that $\mathbf{f} \sim \text{MVN}(\mathbf{0}, \sigma_f^2 \mathbf{I})$ and $\mathbf{e} \sim \text{MVN}(\mathbf{0}, \sigma_e^2 \mathbf{I})$, show that

$$\text{SNR} = \frac{\sigma_f^2 \langle E_h \rangle}{\sigma_e^2} .$$

You need to know that the pulse energy E_h and its time average are

$$E_h[\ell, m] = \int dx h^2(x, [\ell, m]) \quad \text{and} \quad \langle E_h \rangle = \frac{1}{LM} \sum_{\ell, m} E_h[\ell, m] .$$