Spectral Estimation and Adaptive Signal Processing:

Lecture 0 - Background

Danilo Mandic
room 813, ext: 46271

Department of Electrical and Electronic Engineering
Imperial College London, UK

d.mandic@imperial.ac.uk, URL: www.commsp.ee.ic.ac.uk/~mandic
Outline

Background on:-

• Linear Algebra
• Estimation Theory
• Gaussianity
• Maximum Likelihood Estimation
• Sequential and Block Estimators
• Ergodicity
• Phase space, attractor dynamics
Is there such a thing as a nonlinear signal?

**Linear System**

System which obeys superposition and scaling principles:

\[ f(ax + by) = af(x) + bf(y) \]

**Nonlinear System**

System which does not obey these principles

**Linear Signal**

Signal generated by a linear system driven by white (Gaussian) noise

**Nonlinear Signal**

Any (!) signal which is not linear
Real World Signals – ‘Nonlinearity’ and ‘Stochasticity’

- Nonlinearity
- Chaos
- Linearity
- Determinism
- Stochasticity

(a) Periodic oscillations
(b) Small nonlinearity
(c) Route to chaos
(d) Route to chaos
(e) Small noise
(f) HMM and others

D. P. Mandic
Spectral Estimation & Adaptive Signal Processing, Spring 2014
So - how about a real-world sinewave?

*Spectral estimation or adaptive filtering to estimate it?*

Matlab code:

```matlab
z1=0;
p1=[0.5+0.866i,0.5-0.866i];
[num1,den1]=zp2tf(z1,p1,1);
zplane(num1,den1);
s=randn(1,1000);
s1=filter(num1,den1,s);
figure;
subplot(311),plot(s),subplot(313),plot(s1),subplot(312),;
zplane(num1,den1)
```

The AR model of a sinewave

\[ x(k) = a_1 x(k-1) + a_2 x(k-2) + w(k) \]

\( a_1 = -1, \ a_2 = 0.98, \ w \sim N(0,1) \)
System Nonlinearity Detection

**Parametric System NL Testing**

- develop parametric nonlinear model
- drive unknown system with known input
- fit model parameters to match system output

**Nonparametric System NL Testing**

- drive unknown system with different stimuli
- test the superposition and scaling principles
  1. apply short and long stimulus $\rightarrow R_s, R_l$
  2. can $R_l$ be predicted from time shifted $R_s$
  2. apply small and large amplitude stimulus $\rightarrow R_{sa}, R_{la}$
  2. can $R_{la}$ be predicted from superposition of $R_{sa}$
Analysis in the Phase Space

How do we differentiate between signals, second order statistics is not enough

The chaotic Lorenz signal and its linear estimate have the same spectral properties.
The Lorenz signal has an attractor in phase space (higher order properties).
Practical issues in correlation and spectrum estimation: Data length and data windows
Some common windows for different window lengths:

Time domain  | Spectrum N=64 | Spectrum N=128 | Spectrum N=256
--- | --- | --- | ---
Rectangular window (64 samples) | Spectral leakage – 64-sample window | Spectral leakage – 128-sample window | Spectral leakage – 256-sample window

Bartlett window (64 samples) | Spectral leakage – 64-sample window | Spectral leakage – 128-sample window | Spectral leakage – 256-sample window

Hamming window (64 samples) | Spectral leakage – 64-sample window | Spectral leakage – 128-sample window | Spectral leakage – 256-sample window

© D. P. Mandic  
Spectral Estimation & Adaptive Signal Processing, Spring 2014
Types of Estimators

Window length matters in sequential estimation too!

Block based and sequential

Example: block based - sample mean

\[ \hat{\theta}_{N-1} = \frac{1}{N} \sum_{k=0}^{N-1} x[k] \]

It is block based since all the data are required to form an estimate.

Sequential estimator a new estimate can be calculated as each new sample arrives.

we want to express \( \hat{\theta}(N) = f(\hat{\theta}(N - 1), x(N)) \)

\[ \hat{\theta}[N] = \frac{1}{N+1} \sum_{k=0}^{N} x[k] = \frac{1}{N+1} \left[ \sum_{k=0}^{N-1} x[k] + x[N] \right] \]

\[ = \frac{N}{N+1} \hat{\theta}[N - 1] + \frac{1}{N+1} x[n] \]

\[ = \left( 1 - \frac{1}{N+1} \right) \hat{\theta}[N - 1] + \frac{1}{N+1} x[n] \]

\[ = \hat{\theta}[N - 1] + \frac{1}{N+1} \left[ x[N] - \hat{\theta}[N - 1] \right] \]

This update equation is the common form of all adaptive algorithms

New quantity = old quantity + Gain \times Error

\( \frac{1}{N+1} = \text{gain}, \ x[N] - \hat{\theta}[N - 1] = \text{error}. \) This is the basis for the LMS (Least mean square), NLMS (Normalised LMS), RLS (recursive least squares), Kalman filtering.
Convergence of sequential estimators
for simplicity we consider a Gaussian $\sim N(0, 1)$

**Sequential DC level estimation.**
*(mean of a Gaussian signal)*

Signal: uncorrelated Gaussian noise

**Sequential variance estimation.**
*(variance of a Gaussian)*

Signal: uncorrelated Gaussian noise
Second order data modelling

We start from: \( y(n) = a_1(n)x(n - 1) + a_2(n)x(n - 2) + \cdots + a_p(n)x(n - p) \)

Teaching signal: \( d(n) \),  
Output error: \( e(n) = d(n) - y(n) \)

Fixed coeff. \( a \) & \( x(n) = y(n) \)

Autoregressive modelling

\[
\begin{align*}
    r_{xx}(1) &= a_1 r_{xx}(0) + \cdots + a_p r_{xx}(p - 1) \\
    r_{xx}(2) &= a_1 r_{xx}(1) + \cdots + a_p r_{xx}(p - 2) \\
    &\vdots \quad = \quad \vdots \\
    r_{xx}(p) &= a_1 r_{xx}(p - 1) + \cdots + a_p r_{xx}(0) \\
    \cdots &\quad \cdots \\
    r_{xx} &= R_{xx} a
\end{align*}
\]

Solution: \( a = R_{xx}^{-1} r_{xx} \)

Yule–Walker equation

Fixed optimal coeff. \( w_o = a_{opt} \)

\[
J = E\left\{ \frac{1}{2} e^2(n) \right\} = \sigma_d^2 - 2w^T p + w^T R w
\]

is quadratic in \( w \) and for a full rank \( R \), it has one unique minimum.

Now:

\[
\frac{\partial J}{\partial w} = -p + R \cdot w = 0
\]

Solution: \( w_o = R^{-1} p \)

Wiener–Hopf equation
we need to calculate those inverses for our main models

**The structure of an inverse matrix is also important!**

- A **symmetric** matrix has a **symmetric** inverse
- A **Toeplitz** matrix has a **persymmetric** inverse (symmetric about the cross diagonal)
- A **Hankel** matrix has a **symmetric** inverse

A useful formula for matrix inversion is **Woodbury’s identity** (matrix inversion Lemma)

\[
(A + UV^T)^{-1} = A^{-1} - [A^{-1}U(I + V^TA^{-1}U)^{-1}V^TA^{-1}]
\]

If \( u \) and \( v \) are vectors, then this simplifies into

\[
(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}
\]

which will be important in the derivation of adaptive algorithms.

In a special case \( A = I \) and we have

\[
(I + uv^T)^{-1} = I - \frac{uv^T}{1 + v^Tu}
\]
Matrix Inversion Lemma - Another Approach

Construct an augmented matrix \([A, B; C, D]\) and its inverse \([E, F; G, H]\), so that

\[
\begin{bmatrix}
  A & B \\
  C & D
\end{bmatrix}^{-1} = \begin{bmatrix}
  E & F \\
  G & H
\end{bmatrix}
\]

Consider the following two products

\[
\begin{align*}
\begin{bmatrix}
  A & B \\
  C & D
\end{bmatrix}\begin{bmatrix}
  E & F \\
  G & H
\end{bmatrix} &= \begin{bmatrix}
  AE + BG & AF + BH \\
  CE + HC & CF + DH
\end{bmatrix} = \begin{bmatrix}
  I & 0 \\
  0 & I
\end{bmatrix}
\end{align*}
\]

and

\[
\begin{align*}
\begin{bmatrix}
  E & F \\
  G & H
\end{bmatrix}\begin{bmatrix}
  A & B \\
  C & D
\end{bmatrix} &= \begin{bmatrix}
  EA + FC & EB + FD \\
  GA + HC & GB + HD
\end{bmatrix} = \begin{bmatrix}
  I & 0 \\
  0 & I
\end{bmatrix}
\end{align*}
\]

Combine the corresponding terms to get another form

\[
(A - BD^{-1}C)^{-1} = A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}
\]
Matrix Inversion Lemma - Handy Derivation

General form:
\[(A + BCD)^{-1} = (A[I + A^{-1}BCD])^{-1} = [I + A^{-1}BCD]^{-1} A^{-1}
\]
\[= [I - (I + A^{-1}BCD)^{-1} A^{-1}BCD] A^{-1} \quad (\text{using P1})
\]
\[= A^{-1} - (I + A^{-1}BCD)^{-1} A^{-1}BCDA^{-1}
\]

where the identity P1:
\[(I + P)^{-1} = (I + P)^{-1}(I + P - P) = I - (I + P)^{-1}P
\]

We can also use identity P2:
\[P + PQP = P(I + QP) = (I + PQ)P \quad \Rightarrow \quad (I + PQ)^{-1}P = P(I + QP)^{-1}
\]
to have another matrix inverse:
\[(A + BCD)^{-1} = A^{-1} - A^{-1}BCDA^{-1}(I + BCDA^{-1})^{-1}
\]

There are many other equivalent forms - you can derive them easily when needed
The meaning of eigenanalysis

Let \( A \) be an \( n \times n \) matrix, where \( A \) is a linear operator on vectors in \( \mathbb{R}^n \), such that \( A x = b \)

An **eigenvector** of \( A \) is a vector \( v \in \mathbb{R}^n \) such that \( A v = \lambda v \), where \( \lambda \) is called the corresponding eigenvalue.

**A only changes the length of \( v \), not its direction!**

\[
\begin{align*}
A v &= \lambda v \\
A \mathbf{x} &= \lambda \mathbf{x} \\
A \mathbf{x} &= \mathbf{b}
\end{align*}
\]
Eigenvalues

For an $n \times n$ matrix $A$, its eigenvalues are found from the $n$-th order polynomial in $\lambda$ defined by

$$Ax = \lambda x \quad \Rightarrow \quad \det (A - \lambda I) = 0$$

where $I$ is the $n \times n$ identity matrix.

The corresponding $n$ eigenvectors satisfy $Av = \lambda v$ and are generally normalised to have unit norm $\|v\|_2 = 1$.

For distinct eigenvalues, these eigenvectors are linearly independent.

A symmetric matrix is positive definite iff all its eigenvalues are positive.

The Spectral Theorem allows for a symmetric matrix to be written as

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T$$

and the

$$\text{Trace}(A) = \sum_{i=1}^{n} \lambda_i$$

Any connection with signal power?
Intuitively, what is the eigenvector of \( C \)? Is there a point on the unit sphere that a 45° rotation transforms into a multiple of itself?

This is the north pole \([0, 0, 1]\) \(\Rightarrow\) an eigenvector of \( C \) is the vector \([0, 0, 1]\).

\(^1\)By 45° in this case. For example multiplying \( C \) by the vector \([1, 0, 0]\) yields the vector \([0.707, 0.707, 0]\), which is rotated 45°.
Eigenanalysis – Image representation

**Eigenvector** of a transformation ⇒ a vector which, in the transformation is multiplied by a constant factor, called the *eigenvalue of that vector*

The *red* vector is an eigenvector of the transformation, and the *blue* is not.
Eigenanalysis – Eigenvectors

The red vector was neither stretched nor compressed ⇒ its eigenvalue is 1.
All vectors with the same vertical direction (parallel) are also eigenvectors, with the same eigenvalue.

Together with the zero vector, they form the eigenspace for this eigenvalue.

The eigenvalue problem in linear algebra:-

□ Given a matrix $\mathbf{A}$, are there nonzero vectors $\mathbf{x}$ such that product $\mathbf{A}\mathbf{x}$ is a multiple of $\mathbf{x}$, that is $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$?

□ If so, then the factor $\lambda$ by which $\mathbf{x}$ is multiplied is called an eigenvalue of the matrix $\mathbf{A}$.

□ The corresponding vector $\mathbf{x}$ is called an eigenvector of the matrix $\mathbf{A}$ corresponding to this eigenvalue.
Transformation

\[ A = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} \]

Eigenvectors:

\[ Ax = \lambda x \implies (A - \lambda I)x = 0 \]

For non-trivial solutions \( \rightarrow \det(A - \lambda I) = 0 \implies \)

\[ \det \left( \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) = \det \left( \begin{bmatrix} 1 - \lambda & 0 \\ -\frac{1}{2} & 1 - \lambda \end{bmatrix} \right) = 0 \implies \lambda = 1 \]
Eigenvectors for Mona Lisa

We have found $\lambda = 1$, the eigenvalue of matrix $A$.

We can now solve for eigenvectors

$$
\begin{bmatrix}
1 - \lambda & 0 \\
-\frac{1}{2} & 1 - \lambda
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 0
$$

Substituting $\lambda = 1$ we have

$$
\begin{bmatrix}
0 \\
c
\end{bmatrix}
$$

where $c$ is an arbitrary constant.

All vectors of this form, pointing straight up or down, are eigenvectors of $A$.

In general $A$ will have two distinct eigenvalues, and thus two distinct eigenvectors.

Most vectors will have both their lengths and direction changed by $A$ whereas eigenvectors will have only their lengths changed.
Example:-

Let \( A = \begin{bmatrix} 2 & -4 \\ -1 & -1 \end{bmatrix} \). The characteristic polynomial \( det(A - \lambda I) \) is

\[
p(\lambda) = det \left( \begin{bmatrix} 2 - \lambda & -4 \\ -1 & -1 - \lambda\end{bmatrix} \right)
\]
\[
= (2 - \lambda)(-1 - \lambda) - (-4)(-1) = \lambda^2 - \lambda - 6 = (\lambda - 3)(\lambda + 2)
\]

Thus the eigenvalues of \( A \) are \( \lambda_1 = 3 \) and \( \lambda_2 = -2 \).

To find eigenvectors \( \mathbf{v} = [v_1, \ldots, v_n]^T \) corresponding to an eigenvalue \( \lambda \)

solve \( (A - \lambda I) \mathbf{v} = 0 \) for \( \mathbf{v} \)

For \( \lambda_1 = 3 \) we thus have

\[
\begin{bmatrix} 2 - 3 & -4 \\ -1 & -1 - 3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \mathbf{v}_1 = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \end{bmatrix}
\]

Similarly, for \( \lambda_2 = -2 \) we have \( \mathbf{v}_2 = [1, 1]^T \).

\( \mathbf{v}_1 \) and \( \mathbf{v}_2 \) are bases of the eigenspace spanned by these vectors
Example 2: Connection with Linear Algebra

South Ken has two pizza places and \( N \to \infty \) of pizza–loving students.

- 5000 people buy one pizza each every week.

Tony’s Pizza place has the better pizza and 80% of people who buy pizza each week at Tony’s return the following week. Mike’s Pizza does not have a good sauce and only 40% of the customers return the following week.

We can represent this situation by a discrete dynamical system

\[
x_{n+1} = A x_n \quad \text{where} \quad A = \begin{bmatrix} 0.8 & 0.6 \\ 0.2 & 0.4 \end{bmatrix}
\]

Let us start from \( x_0 = [2500, 2500]^T \), then we have

\[
x_1 = \begin{bmatrix} 3500 \\ 1500 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 3700 \\ 1300 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 3740 \\ 1260 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 3748 \\ 1252 \end{bmatrix}, \quad x_5 = \begin{bmatrix} 3750 \\ 1250 \end{bmatrix}
\]

Also \( x_6 = \cdots = x^* = \cdots = x_\infty = [3750, 1250]^T \! \)!

Clearly \( x^* = [3750, 1250]^T \) is the eigenvector of \( A \), and \( A x^* = x^* \)
Example 2: Will Mike have to close down?

In Matlab

\[ [v, \lambda] = \text{eig}(A) \quad \% \text{also look at the demo 'eigshow(A)'} \]

\[
\begin{bmatrix}
0.9487 & -0.7071 \\
0.3162 & 0.7071 \\
1.0000 & 0 \\
0 & 0.2000
\end{bmatrix}
\]

\[ v_1 = [0.9487, 0.3162]^T, \quad v_2 = [-0.7071, 0.7071]^T, \quad \lambda_1 = 1, \quad \lambda_2 = 0.2. \]

Notice that the elements of \( v_1 \) are related as \( 3 \div 1 \), the same as the ratio of Tony’s and Mike’s customers.

Since \( x_n = A^nx_0 \Leftrightarrow A_{n \to \infty} = [v_1 : v_2] \quad \Leftrightarrow \quad \text{equilibrium!} \]

This is closely related to fixed point theory since \( A \) is a Markov matrix.
Estimated Theory

**Problem:** To estimate one or more parameters from some given discrete-time signal \( \{x[n]\} \).

**Mathematical Context:**
Given an \( N \)-point data set which depends upon an unknown parameter \( \theta \) (scalar), define an "estimator" as some function \( g \) of the dataset, that is
\[
\hat{\theta} = g(x[0], x[1], \ldots, x[N - 1])
\]

Note: difference between estimator and estimate

- **Estimate** = particular value of \( \hat{\theta} \) for one observation data set
- **Estimator** = rule that assigns a value of \( \hat{\theta} \) for a given realisation of
\[
x = [x[0], x[1], \ldots, x[N - 1]]
\]

There are two types of estimators

- **Classical:** the unknown parameter(s) is(are) fixed
- **Bayesian:** the unknown parameter(s) is(are) random
The bias–variance dilemma

The mean square error (MSE) of an estimate $\hat{\theta}$ of a parameter $\theta$ is given by

$$MSE(\hat{\theta}) = E\{ (\hat{\theta} - \theta)^2 \} \quad \text{average deviation from the true value}$$

For every estimator:

**Bias:** $B = E\{ \hat{\theta} \} - \theta$

**Variance:** $\text{var} = E\{(\hat{\theta} - E\{\hat{\theta}\})^2\}$

Therefore:

$$\text{MSE} = E\{ (\hat{\theta} - \theta)^2 \} = E\{ [\hat{\theta} - E\{\hat{\theta}\} + E\{\hat{\theta}\} - \theta]^2 \}$$

$$= E\{ [\hat{\theta} - E\{\hat{\theta}\}]^2 \} + E\{ B^2(\hat{\theta}) \} + 2E\{ [\hat{\theta} - E\{\hat{\theta}\}] B(\hat{\theta}) \}$$

due to the linearity of the $E\{ \cdot \}$ operator and that $E\{ B(\hat{\theta}) \} = B(\hat{\theta})$

$$= E\{ [\hat{\theta} - E\{\hat{\theta}\}]^2 \} + B^2(\hat{\theta}) + 2E\{ [\hat{\theta} - E\{\hat{\theta}\}] \} B(\hat{\theta})$$

$$= \text{var}(\hat{\theta}) + B^2(\hat{\theta})$$
Data Model: Gaussianity

Justification: Central Limit Theorem

If we form a sum of independent measurements
⇒ the distribution of the sum tends to a Gaussian distribution

\[ p(x) = \frac{1}{\sqrt{2\pi \sigma_x^2}} e^{-\frac{(x-\mu_x)^2}{2\sigma_x^2}} \quad x \sim \mathcal{N}(\mu_x, \sigma_x^2) \]

⇒ distribution defined by its mean and variance!!! SOS

If \( x \sim \mathcal{N}(0, \sigma_x^2) \) then \( E\{x^{2n-1}\} = 1, 3, \ldots, (2n - 1)\sigma_x^{2n}, \quad \forall n \)

In the vector case (\( N \) Gaussian random variables)

\[ p(x[0], x[1], \ldots, x[N - 1]) = \frac{1}{(2\pi)^{N/2} \det(C_{xx})^{1/2}} e^{-\frac{1}{2}(x-\mu_x)^T C_{xx}^{-1}(x-\mu_x)} \]

where \( C_{xx} = E\{(x - \mu_x)(x - \mu_x)^T\} \) is the covariance matrix.
Gaussian Random Variables:- Properties

Note

\[ \text{cov}(x_i, x_j) = E[(x_i - E(x_i))(x_j - E(x_j))] = E[x_i x_j] - E[x_i]E[x_j] \]

If \( x_i \) and \( x_j \) are independent \( \Rightarrow C_{xx} \) is diagonal.

For mathematical tractability, we will frequently assume that the component of \( x \) are statistically independent, such that:-

\[ C_{xx} = \text{diag}(\sigma_0^2, \sigma_1^2, \ldots, \sigma_{N-1}^2) \]

and if all the variances and mean values are identical, then the elements of \( x \) are said to be iid, that is Independent Identically Distributed.

\[ \Rightarrow C_{xx} = \sigma_x^2 I, \text{ where } I = N \times N \text{ identity matrix} \]

\[ \text{For } x \text{ IID } \Rightarrow p(x) = \frac{1}{(2\pi)^{N/2}\sigma_x^N} e^{-\frac{1}{2\sigma_x^2} \sum_{i=0}^{N-1} (x[i] - \mu_x)^2} \]
Properties of a Gaussian

\[ p(x) = \sqrt{\frac{1}{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]

- 68% of values lie within \( \mu - \sigma \) and \( \mu + \sigma \)
- 95% of values lie within \( \mu - 2\sigma \) and \( \mu + 2\sigma \)
- >99% of values lie within \( \mu - 3\sigma \) and \( \mu + 3\sigma \)
Linear Algebra

Solving a set of linear equations is necessary in the formulation of many algorithms for spectral estimation and adaptive signal processing!

A set of linear equations is most conveniently represented in matrix form:-

\[ Ax = b \]

where the dimensions are

- \( A : m \times n \)
- \( x : n \times 1 \)
- \( b : m \times 1 \)

Depending on \( m \) and \( n \) we have three special cases:-

\( m = n \)

\( m < n \) \( \Rightarrow \) under-determined case

\( m > n \) \( \Rightarrow \) over-determined case
The Three Cases

• \( m = n \) - **Exactly determined set of equations** If \( A \) is invertible or nonsingular, i.e. it has full column rank, with \( m \) linearly independent columns, then

\[
x = A^{-1}b
\]

If \( A \) is singular, then there may be either no solution or many solutions.

• \( m < n \) - **There are more equations than unknowns (underdetermined)** therefore, in general, no solution exists. A least squares solution that minimises \( \| Ax - b \|_2^2 \) can be found

\[
A^T Ax = A^T b
\]

If \( A \) is invertible, then

\[
x = (A^T A)^{-1} A^T b \quad \text{The least squares solution}
\]

with minimum least squares error \( b^T b - b^T A x \).

• \( m > n \) - **the overdetermined case**
A unique minimum norm solution is found by \( \min \| x \|_2^2 \) such that \( A x = b \).
Provided that the rows of \( A \) are linearly independent, we have

\[
x = A^T (A A^T)^{-1} b
\]

where \( A^T (A A^T)^{-1} \) is the pseudo-inverse of \( A \).
Special Matrix Forms

The exchange matrix

\[ J = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix} \] - notice \( J \times J = I \)

\( \Rightarrow J \) has its own inverse; and \( J^T A J \) reflects each element of the square matrix \( A \) about its central element.

Toeplitz matrix - has a tremendous amount of structure

Toeplitz \( J \) = \[ \begin{bmatrix} \begin{array}{cccc} 1 & 3 & 5 & 7 \\ 2 & 1 & 3 & 5 \\ 4 & 2 & 1 & 3 \\ 6 & 4 & 2 & 1 \end{array} \end{bmatrix} \] \hspace{1cm} \text{Hankel } J = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 3 & 5 & 7 & 4 \\ 5 & 7 & 4 & 2 \\ 7 & 4 & 2 & 1 \end{bmatrix}

for which \( a_{ij} = a_{i+1,j+1}, \ \forall i < n, j < n \).

Notice that all entries are defined once the first column and first row have been specified.

A convolution matrix is an example of a Toeplitz matrix.

Hankel matrix - similar to Toeplitz for which \( a_{ij} = a_{i+1,j-1}, \ \forall i < n, j \leq n \).

An example of a Hankel matrix is the exchange matrix.
Taylor Series Expansion

Most 'smooth' functions can be expanded into their Taylor Series Expansion (TSE)

\[ f(x) = f(x_0) + \frac{f'(x_0)}{1}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots = \sum_{n=1}^{\infty} \frac{f^{(n)}(x_0)}{n!} \]

To show this consider the polynomial

\[ f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + \cdots \]

1. To get \( a_0 \) \( \iff \) choose \( x = x_0 \) \( \Rightarrow \) \( a_0 = f(x_0) \)
2. To get \( a_1 \) \( \iff \) take derivative of the polynomial above to have

\[ \frac{df}{dx}(x) = a_1 + 2a_2(x - x_0) + 3a_3(x - x_0)^2 + 4a_4(x - x_0)^4 + \cdots \]

choose \( x = x_0 \) \( \Rightarrow \) \( a_1 = \frac{df(x)}{dx}|_{x=x_0} \) and so on ... \( a_k = \frac{1}{k!} \frac{d^k f(x)}{dx^k}|_{x=x_0} \)
Power Series - contd.

Consider

\[ f(x) = \sum_{n=0}^{\infty} a_n x^n \Rightarrow f'(x) = \sum_{n=1}^{\infty} n a_n x^{n-1} \quad \text{and} \quad \int_{0}^{x} f(t) dt = \sum_{n=0}^{\infty} \frac{a_n}{n+1} x^{n+1} \]

1. Exponential function, cosh, sinh, sin, cos, ...

\[ e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \quad \text{and} \quad e^{-x} = \sum_{n=0}^{\infty} (-1)^n \frac{x^n}{n!} \Rightarrow \frac{e^x - e^{-x}}{2} = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!} \]

2. other useful formulas

\[ \sum_{n=0}^{\infty} x^n = \frac{1}{1 - x} \Rightarrow \sum_{n=1}^{\infty} n x^{n-1} = \frac{1}{(1 - x)^2} \quad \text{and} \quad \frac{1}{1 + x^2} = \sum_{n=0}^{\infty} (-1)^n n x^2 \]

Integrate to obtain \( \arctan(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1} \). For \( x = 1 \) we have

\[ \frac{\pi}{4} = 1 = 1/3 + 1/5 - 1/7 + \cdots \]
Numerical Differentiation - examples

- \( f' = \frac{f(1) - f(0)}{h} \)
- \( f' = \frac{f(0) - f(1)}{h} \)
- \( f' = \frac{f(1) - 2f(0) + f(-1)}{2h} \)
- \( f'' = \frac{f(1) - 2f(0) + f(-1)}{h^2} \)
- \( f' = \frac{f(-2) - 8f(-1) + 8f(1) - f(2)}{12h} \)
- \( f'' = \frac{-f(-2) + 16f(-1) - 30f(0) + 16f(1) - f(2)}{12h^2} \)
Vectorial Scalar Function

\[ f(x = f(x_1, \ldots, x_N) \]

Gradient \( \nabla_x f(x) \) = 
\[
\begin{bmatrix}
\frac{\partial f(x)}{\partial x_1} \\
\frac{\partial f(x)}{\partial x_2} \\
\vdots \\
\frac{\partial f(x)}{\partial x_N}
\end{bmatrix}
\] = 0 and the Hessian matrix \( H_x > 0 \).

where the elements of the Hessian matrix are \( \{H_x\}_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \)

**Theorem:** If \( f(z, z^*) \) is a real-valued function of the complex vectors \( z \) and \( z^* \), the vector pointing in the direction of the maximum rate of change of \( f(z, z^*) \) is \( \nabla_z f(z, z^*) \), the derivative of \( f(z, z^*) \) wrt \( z^* \). [Hayes 1996].

Thus, the turning points of \( f(z, z^*) \) are solutions to \( \nabla_{z^*} f(z, z^*) = 0 \),

where \( \nabla_{z^*} = \frac{1}{2} \begin{bmatrix}
\frac{\partial}{\partial x_1} - J \frac{\partial}{\partial y_1} \\
\vdots \\
\frac{\partial}{\partial x_n} - J \frac{\partial}{\partial y_n}
\end{bmatrix} \)

\( , \nabla_z a^H z = a^* \), \( , \nabla_{z^*} a^H z = 0 \)
Summary

- Several background concepts in one place

- The course is self–contained, most of the background should be found in this Lecture

- For more detail about spectrum estimation, refer to Hayes’ book

- For more detail about adaptive filtering, refer to Haykin’s book

- We will frequently come back to this lecture, when using these techniques in spectral estimation and adaptive signal processing

- Towards the end of the course, we will combine spectrum estimation and adaptive signal processing towards an application in brain computer interface (BCI)

We will next “get our hands dirty” with real world spectrum estimation
Notes:

○