Numerical Linear Algebra and Optimisation

Notes for MATH3204, The University of Queensland

George Stuart

Semester 1, 2026

About these notes. These are my notes from MATH3204, written to go alongside the course's lecture slides and lecture notes rather than to replace them. Some sections only summarise the main points, a few stop part-way through, and many proofs are left to the slides. Section n covers Lecture n, so a reference such as "slide 15 of Lecture 6" points to the matching set of slides.

Contents

I Numerical Linear Algebra **2**
1 Vector Spaces . . . 2
2 Matrices . . . 6
3 Norms . . . 10
4 Eigenvalues and Eigenvectors . . . 13
5 Singular Value Decomposition (SVD) . . . 19
6 Special Matrices . . . 21
7 Matrix Factorisations . . . 27
8 Linear Systems (Direct Methods) . . . 28
9 Least Squares (Direct Methods) . . . 30
10 Stationary Methods . . . 34
11 Polynomial Acceleration . . . 36
12 Krylov Subspaces . . . 39
13 Krylov Basis . . . 41
14 Krylov Optimality . . . 41
15 CG and GMRES . . . 45

II Optimisation **45**
16 Introduction to Optimisation . . . 45
17 Topology . . . 46
18 Calculus . . . 49
19 Optimality . . . 51
20 Convexity . . . 55
21 Descent Methods . . . 57
22 Gradient Descent . . . 59
23 Newton's Method . . . 60
24 Quasi-Newton Methods . . . 61

Part I

Numerical Linear Algebra

1 Vector Spaces

1.1 Field

Formally a field is a set $\mathbb{F}$ together with two binary operations "$+$" and "$\times$" which satisfy the field axioms:

- Associativity
- Commutativity
- Additive and Multiplicative identity
- Additive inverses
- Multiplicative inverse (the additive identity doesn't need an inverse)
- Distributivity of multiplication over addition

1.2 Vector Spaces

Vector spaces are the fundamental objects for linear algebra.

A vector space $\mathcal{V}$ over a field $\mathbb{F}$ is a set of objects (called vectors) together with the operations of vector addition ($+$) and scalar multiplication ($\times$) that satisfies the following axioms:

- Closure of vector addition
- Commutativity of addition
- Associativity of addition
- Additive identity
- Additive inverse
- Closure of scalar multiplication
- Distributive laws
- Associative law
- Monoidal law

Less formally a vector space $\mathcal{V}$ over a field $\mathbb{F}$ is a set such that,

$$\forall \boldsymbol{u}, \boldsymbol{v} \in \mathcal{V} \text{ and } \forall a, b \in \mathbb{F} \qquad a\boldsymbol{u} + b\boldsymbol{v} \in \mathcal{V}$$

(and just checking this is usually sufficient for $\mathcal{V}$ to be a vector space).

1.3 Subspace

A subspace $\mathcal{W}$ of a vector space $\mathcal{V}$ is a subset $\mathcal{W} \subseteq \mathcal{V}$ that is by itself also a vector space over the same field as $\mathcal{V}$.

Key notes:

- The intersection of any collection of subspaces is also a subspace (proof is trivial)
- The union of two subspaces is not necessarily a subspace
  Proof. As a counter example, suppose $\mathcal{V} = \mathbb{R}^2$ and consider the subspaces,

$$\mathcal{U} = \operatorname{span}\left\{\begin{pmatrix}1\\0\end{pmatrix}\right\} \qquad \text{and} \qquad \mathcal{W} = \operatorname{span}\left\{\begin{pmatrix}0\\1\end{pmatrix}\right\}$$

These are both valid subspaces, however $\mathcal{U} \cup \mathcal{W}$ is not a subspace. To show this explicitly,

$$\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix} \in \mathcal{U} \cup \mathcal{W} \quad \text{but} \quad \begin{pmatrix}1\\0\end{pmatrix} + \begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}1\\1\end{pmatrix} \notin \mathcal{U} \cup \mathcal{W} \qquad \square$$

- The sets $\{\mathbf{0}\}$ and $\mathcal{V}$ are always subspaces of $\mathcal{V}$ and are called trivial subspaces
- A subspace $\mathcal{W} \subseteq \mathcal{V}$ is called a proper subspace if $\mathcal{W} \neq \mathcal{V}$

One other important property:

- Every subspace must contain the zero vector

1.4 Span

Let $\mathcal{V}$ be a vector space over a field $\mathbb{F}$ and let $\mathcal{S} \subseteq \mathcal{V}$. The span of $\mathcal{S}$ is the set of all finite linear combinations of the vectors in $\mathcal{S}$.

I.e.,

$$\text{span}(\mathcal{S}) = \left\{ \sum_{i=1}^{k} a_i \boldsymbol{v}_i \ : \ \forall i, \ \boldsymbol{v}_i \in \mathcal{S}, a_i \in \mathbb{F} \right\}$$

Note that for any $\mathcal{S} \subseteq \mathcal{V}$, $\text{span}(\mathcal{S})$ is a subspace of $\mathcal{V}$ (follows trivially).

1.5 Linear Dependence & Independence

A finite set $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} \subseteq \mathcal{V}$ is linearly dependent if there exist scalars $a_1, \ldots, a_n \in \mathbb{F}$ not all zero such that,

$$\sum_{i=1}^{n} a_i \boldsymbol{v}_i = \mathbf{0}$$

This means that at least one of the vectors in the set can be written as a linear combination of the other vectors.

If the only solution to the above equation is that $a_1 = \cdots = a_n = 0$ then the vectors are said to be linearly independent.

1.6 Basis Vectors

A (possibly countably infinite) set of vectors $\mathcal{B} \subseteq \mathcal{V}$ is a basis for $\mathcal{V}$ if,

- the vectors in $\mathcal{B}$ span $\mathcal{V}$
- the vectors in $\mathcal{B}$ are linearly independent

Given that $\mathcal{B}$ is a basis for $\mathcal{V}$, then any vector $\boldsymbol{w} \in \mathcal{V}$ can be represented by a **unique** linear combination of the vectors in $\mathcal{B}$.

Proof. Suppose $\boldsymbol{w} \in \mathcal{V}$ had two representations such that,

$$\boldsymbol{w} = \sum_{i=1}^{n} a_i \boldsymbol{v}_i \qquad \text{and} \qquad \boldsymbol{w} = \sum_{i=1}^{n} b_i \boldsymbol{v}_i$$

Subtracting these two expressions for $\boldsymbol{w}$ we get that,

$$\sum_{i=1}^{n} (a_i - b_i) \boldsymbol{v}_i = \mathbf{0}$$

However given that $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ are linearly independent, the only way that this can be the case is if,

$$a_i - b_i = 0 \quad \forall i \qquad \Longrightarrow \qquad a_i = b_i$$

And thus the representation of $\boldsymbol{w}$ is unique. □

Properties:

- The empty set $\{\}$ is a basis for the zero vector space $\{\mathbf{0}\}$

1.7 Dimension of a Vector Space

The dimension of a vector space $\mathcal{V}$ over $\mathbb{F}$ is the number of vectors in any basis of $\mathcal{V}$.

Key facts:

- If $\mathcal{W} \subseteq \mathcal{V}$ is a subspace then $\dim(\mathcal{W}) \leq \dim(\mathcal{V})$
- If $\mathcal{W} \subseteq \mathcal{V}$ and $\dim(\mathcal{W}) = \dim(\mathcal{V})$ then $\mathcal{W} = \mathcal{V}$
- If $\dim(\mathcal{V}) = d$ then
  - any linearly independent set in $\mathcal{V}$ has at most d vectors
  - every basis of $\mathcal{V}$ has exactly d vectors
- The only vector space of dimension 0 is $\{\mathbf{0}\}$

Important point:

- The dimension depends on the underlying field. For example,

$$\dim_{\mathbb{C}}(\mathbb{C}^n) = n \qquad \text{but} \qquad \dim_{\mathbb{R}}(\mathbb{C}^n) = 2n$$

(this is easy to verify).

1.8 Inner Product Space

The course doesn't cover inner products directly, so they are summarised briefly here.

Let $\mathcal{V}$ be a vector space over a field $\mathbb{F}$. An inner product is a map $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to \mathbb{F}$ that obeys the following axioms:

- Conjugate symmetry: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \overline{\langle \boldsymbol{y}, \boldsymbol{x} \rangle}$
- Linearity: $\langle a\boldsymbol{x} + b\boldsymbol{y}, \boldsymbol{z} \rangle = a \langle \boldsymbol{x}, \boldsymbol{z} \rangle + b \langle \boldsymbol{y}, \boldsymbol{z} \rangle$ (*)
- Positive-definiteness: $\langle \boldsymbol{x}, \boldsymbol{x} \rangle > 0$ for $\boldsymbol{x} \neq \mathbf{0}$

Note that inner product spaces are really only defined for $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$ (the axioms don't make much sense for other fields). There are generalisations, but these are almost always the fields that you work with anyway.

(*) Depending on how the inner product is defined, it is linear in either its first or its second argument (and conjugate linear in the other). So rather than relying on this axiom, consider the specific definition of the inner product you are working with. For example, on $\mathbb{C}^n$ both $\boldsymbol{x}^*\boldsymbol{y}$ and $\boldsymbol{x}^T\overline{\boldsymbol{y}}$ are inner products, and $\langle a\boldsymbol{x} + b\boldsymbol{y}, \boldsymbol{z} \rangle$ expands differently under each. This course uses the first, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^*\boldsymbol{y}$, which is linear in its second argument.

Example:

- On $\mathbb{C}^n$: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^*\boldsymbol{y}$ is a valid inner product

1.9 Orthogonal / Orthonormal Vectors

A set of vectors $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} \subseteq \mathbb{C}^n$ is orthogonal if,

$$\langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle = 0 \quad \forall i \neq j$$

If in addition $\|\boldsymbol{v}_i\| = \sqrt{\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle} = 1$ for all i then the set is said to be orthonormal.

It can be shown that every orthogonal set of non-zero vectors is linearly independent.

Proof. Suppose $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\} \subseteq \mathbb{C}^n \setminus \{\boldsymbol{0}\}$ are orthogonal and that there exist scalars $a_1, \ldots, a_n$ such that,

$$\begin{aligned}
\sum_{i=1}^{n} a_i \boldsymbol{v}_i &= \boldsymbol{0} \\
\left\langle \sum_{i=1}^{n} a_i \boldsymbol{v}_i, \sum_{j=1}^{n} a_j \boldsymbol{v}_j \right\rangle &= 0 \\
\sum_{i=1}^{n} \sum_{j=1}^{n} a_i^* a_j \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle &= 0 \\
\sum_{i=1}^{n} a_i^* a_i \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle &= 0 \quad \text{(orthogonality cancels cross terms)} \\
\sum_{i=1}^{n} |a_i|^2 \langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle &= 0
\end{aligned}$$

Given that $\langle \boldsymbol{v}_i, \boldsymbol{v}_i \rangle > 0$ the only way in which the above can be true is if $|a_i| = 0$ for all i which implies that $a_i = 0$. Therefore $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ are linearly independent. □

(note that the above theorem is often stated as any orthonormal set of vectors is linearly independent since it is common to work with orthonormal sets and this also excludes the possibility of there being a zero vector).

Also a consequence of the theorem above is that:

- any orthonormal set of vectors forms a basis for its span (trivial from the definition of a basis).

1.10 Linear Map

Let $\mathcal{U}$ and $\mathcal{V}$ be vector spaces over the same field $\mathbb{F}$. A map $\boldsymbol{f} : \mathcal{U} \to \mathcal{V}$ is **linear** if,

$$\boldsymbol{f}(\alpha \boldsymbol{x} + \beta \boldsymbol{y}) = \alpha \boldsymbol{f}(\boldsymbol{x}) + \beta \boldsymbol{f}(\boldsymbol{y}) \quad \forall \alpha, \beta \in \mathbb{F},\ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$$

Many "functions" in mathematics fit under the definition of a linear map. For example,

- Differentiation
- Integration
- Expectation

The key idea, shown in the next lecture, is that if $\mathcal{U}$ and $\mathcal{V}$ are finite dimensional then every linear map between them can be represented by a matrix.

2 Matrices

Suppose $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ is a matrix. Matrices come in 3 shapes:

- Square ($m = n$)
- Tall and skinny ($m > n$)
- Short and fat ($m < n$)

Two common matrix operations:

- Transpose, $\boldsymbol{A}^T$
- Conjugate/Hermitian transpose, $\boldsymbol{A}^* = \overline{\boldsymbol{A}}^T = \overline{\boldsymbol{A}^T}$

Note that the Hermitian transpose can just be seen as a generalisation of the regular transpose to complex matrices. That is if $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ then $\boldsymbol{A}^T = \boldsymbol{A}^*$.

Properties:

- $(\boldsymbol{AB})^* = \boldsymbol{B}^* \boldsymbol{A}^*$
- $(\boldsymbol{A} + \boldsymbol{B})^* = \boldsymbol{A}^* + \boldsymbol{B}^*$

A square matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is,

- symmetric if $\boldsymbol{A} = \boldsymbol{A}^T$
- skew-symmetric if $\boldsymbol{A} = -\boldsymbol{A}^T$
- Hermitian if $\boldsymbol{A} = \boldsymbol{A}^*$
- skew-Hermitian if $\boldsymbol{A} = -\boldsymbol{A}^*$
- orthogonal if $\boldsymbol{A}^{-1} = \boldsymbol{A}^T$
- unitary if $\boldsymbol{A}^{-1} = \boldsymbol{A}^*$
- normal if $\boldsymbol{A}\boldsymbol{A}^* = \boldsymbol{A}^* \boldsymbol{A}$

Properties of the above:

- Hermitian matrices have real main diagonals
- Skew-Hermitian matrices have pure imaginary main diagonals
- Skew-symmetric matrices have zeros on the main diagonals

(all the above follow trivially)

2.1 Matrix Operations

Matrix addition and scalar multiplication work elementwise. Matrix multiplication however is more interesting.

If $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ and $\boldsymbol{B} \in \mathbb{C}^{n \times p}$ then $\boldsymbol{AB} \in \mathbb{C}^{m \times p}$ where,

$$\begin{aligned}[\boldsymbol{AB}]_{ij} &= \sum_{k=1}^{n} a_{ik} b_{kj} \\ &= (\text{row } i \text{ of } \boldsymbol{A}) \cdot (\text{column } j \text{ of } \boldsymbol{B})\end{aligned}$$

Another equivalent definition is that,

$$\boldsymbol{AB} = \sum_{k=1}^{n} \boldsymbol{a}_k \boldsymbol{b}_k^T$$

where $\boldsymbol{a}_k$ is the kth column of $\boldsymbol{A}$ and $\boldsymbol{b}_k^T$ is the kth row of $\boldsymbol{B}$. In words, this is the sum of the outer products of the columns of $\boldsymbol{A}$ with the rows of $\boldsymbol{B}$.

One last very important (and equivalent) viewpoint is that,

$$\boldsymbol{AB} = (\boldsymbol{Ax}_1 \mid \ldots \mid \boldsymbol{Ax}_p)$$

where $\boldsymbol{B} = (\boldsymbol{x}_1 | \ldots | \boldsymbol{x}_p)$.

Knowing all of these viewpoints of matrix multiplication is useful since different viewpoints lend themselves more easily to different situations.

Some properties:

- Associativity
- Distributivity
- **Not** commutativity

2.2 Determinants

The course doesn't really give an intuitive interpretation of the determinant (at least not this far into the course).

For any square matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$,

- $\det(\boldsymbol{A}^T) = \det(\boldsymbol{A})$
- $\det(\boldsymbol{A}^*) = \det(\overline{\boldsymbol{A}}) = \overline{\det(\boldsymbol{A})}$
- $\det(\boldsymbol{AB}) = \det(\boldsymbol{A}) \det(\boldsymbol{B})$
- $\det(\boldsymbol{A}^{-1}) = \frac{1}{\det(\boldsymbol{A})}$
- $\det(\alpha \boldsymbol{A}) = \alpha^n \det(\boldsymbol{A}) \quad \forall \alpha \in \mathbb{C}$

Proposition: for any unitary matrix $\boldsymbol{U} \in \mathbb{C}^{n \times n}$, $|\det \boldsymbol{U}| = 1$.

Proof.

$$\begin{aligned} \boldsymbol{UU}^* &= \boldsymbol{I} \\ \det(\boldsymbol{UU}^*) &= \det(\boldsymbol{I}) \\ \det(\boldsymbol{U})\overline{\det(\boldsymbol{U})} &= 1 \\ |\det(\boldsymbol{U})|^2 &= 1 \end{aligned}$$

□

2.3 Trace

Another function defined on square matrices $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is the trace. It is defined by,

$$\operatorname{tr}(\boldsymbol{A}) = \sum_i a_{ii}$$

(i.e. the sum of the entries on the main diagonal).

Properties:

- $\operatorname{tr}(\boldsymbol{A}^T) = \operatorname{tr}(\boldsymbol{A})$
- $\operatorname{tr}(\boldsymbol{A}^*) = \operatorname{tr}(\overline{\boldsymbol{A}}) = \overline{\operatorname{tr}(\boldsymbol{A})}$
- $\operatorname{tr}(\boldsymbol{ABC}) = \operatorname{tr}(\boldsymbol{CAB})$ (cyclic properties of the trace)
- $\operatorname{tr}(\boldsymbol{A} + \boldsymbol{B}) = \operatorname{tr}(\boldsymbol{A}) + \operatorname{tr}(\boldsymbol{B})$
- $\operatorname{tr}(\alpha \boldsymbol{A}) = \alpha \operatorname{tr}(\boldsymbol{A}) \ \forall \alpha \in \mathbb{C}$

Notice that:

- $\operatorname{tr}(\boldsymbol{AB}) \neq \operatorname{tr}(\boldsymbol{A}) \operatorname{tr}(\boldsymbol{B})$

2.4 Matrices as Linear Maps

Let $\mathcal{U}$ and $\mathcal{V}$ be n and m dimensional vector spaces respectively. Let $\mathcal{B}_\mathcal{U}$ and $\mathcal{B}_\mathcal{V}$ be bases for the two spaces and suppose that $\boldsymbol{f} : \mathcal{U} \to \mathcal{V}$ is a linear map. Then there exists a unique matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ such that,

$$[\boldsymbol{f}(\boldsymbol{x})]_{\mathcal{B}_\mathcal{V}} = \boldsymbol{A}[\boldsymbol{x}]_{\mathcal{B}_\mathcal{U}} \quad \forall \boldsymbol{x} \in \mathcal{U}$$

Furthermore the matrix $\boldsymbol{A}$ can be constructed as follows:

- Set the i^{th} column of $\boldsymbol{A}$ to be $[\boldsymbol{f}(\boldsymbol{u}_i)]_{\mathcal{B}_\mathcal{V}}$ (where $\boldsymbol{u}_i$ is the i^{th} basis vector of $\mathcal{U}$)

Or in other words, the matrix $\boldsymbol{A}$ should be set to whatever the basis vectors of $\mathcal{U}$ map to under the transformation $\boldsymbol{f}$ represented using the basis vectors of $\mathcal{V}$.

2.5 Matrix Subspaces

Given that any linear map can be represented by a matrix, we often think of matrices as functions,

$$\boldsymbol{A} \in \mathbb{C}^{m \times n} : \quad \mathbb{C}^n \to \mathbb{C}^m$$

Thinking of matrices in this way is more natural when thinking about some of the common matrix subspaces.

- $\text{dom}(\boldsymbol{A}) = \mathbb{C}^n$
- $\text{range}(\boldsymbol{A}) = \text{colsp}(\boldsymbol{A}) = \{\boldsymbol{A}\boldsymbol{x} \,:\, \boldsymbol{x} \in \text{dom}(\boldsymbol{A})\}$
  - This will necessarily be a subspace of $\mathbb{C}^m$ (proof is trivial)
  - $\text{rank}(\boldsymbol{A}) = \dim(\text{range}\, \boldsymbol{A}) \leq \min\{m, n\}$
- $\text{null}(\boldsymbol{A}) = \ker(\boldsymbol{A}) = \{\boldsymbol{x} \in \text{dom}(\boldsymbol{A}) \,:\, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}\}$
  - This will necessarily be a subspace of $\mathbb{C}^n$
  - $\text{nullity}(\boldsymbol{A}) = \dim(\text{null}\, \boldsymbol{A})$

These are tied together in the very important rank-nullity theorem which states that for any matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$,

$$\text{rank}(\boldsymbol{A}) + \text{nullity}(\boldsymbol{A}) = n$$

This is not proved in the lectures, but intuitively it says that each of the n dimensions of the input space is mapped either into the output space, $\text{range}(\boldsymbol{A})$, or into the null space, $\text{null}(\boldsymbol{A})$.

2.6 Orthogonal Complement

Let $\mathcal{V}$ be an inner product space and suppose $\mathcal{W} \subseteq \mathcal{V}$ is a subspace. Then the orthogonal complement of $\mathcal{W}$ is defined as follows,

$$\mathcal{W}^\perp = \{\boldsymbol{v} \in \mathcal{V} \,:\, \langle \boldsymbol{v}, \boldsymbol{w} \rangle = 0,\ \forall \boldsymbol{w} \in \mathcal{W}\}$$

Note that if $\mathcal{V}$ is finite dimensional then,

$$\dim(\mathcal{W}) + \dim(\mathcal{W}^\perp) = \dim(\mathcal{V})$$

2.7 Four Fundamental Subspaces

If $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ then,

$$\text{null}(\boldsymbol{A}) = \text{range}(\boldsymbol{A}^*)^\perp \qquad \text{and} \qquad \text{range}(\boldsymbol{A}) = \text{null}(\boldsymbol{A}^*)^\perp$$

Proof. Looking at the first equality. Take $\boldsymbol{x} \in \text{null}(\boldsymbol{A})$ and $\boldsymbol{y} \in \text{range}(\boldsymbol{A}^*)$. This implies that $\boldsymbol{y} = \boldsymbol{A}^*\boldsymbol{z}$ for some $\boldsymbol{z}$. Then $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, \boldsymbol{A}^*\boldsymbol{z} \rangle = \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{z} \rangle = \langle \boldsymbol{0}, \boldsymbol{z} \rangle = 0$. Thus $\text{null}(\boldsymbol{A}) \subseteq \text{range}(\boldsymbol{A}^*)^\perp$. Conversely let $\boldsymbol{x} \in \text{range}(\boldsymbol{A}^*)^\perp$, which means that for any $\boldsymbol{y} \in \text{range}(\boldsymbol{A}^*)$ we have $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = 0$. Taking $\boldsymbol{y} = \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x}$,

$$0 = \langle \boldsymbol{x}, \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x} \rangle = \langle \boldsymbol{A}\boldsymbol{x}, \boldsymbol{A}\boldsymbol{x} \rangle = \|\boldsymbol{A}\boldsymbol{x}\|^2$$

so $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$, i.e. $\boldsymbol{x} \in \text{null}(\boldsymbol{A})$. Thus $\text{range}(\boldsymbol{A}^*)^\perp \subseteq \text{null}(\boldsymbol{A})$, which proves the first equality.

The second equality follows by applying the first to $\boldsymbol{A}^*$, which gives $\text{null}(\boldsymbol{A}^*) = \text{range}(\boldsymbol{A})^\perp$, and then taking the orthogonal complement of both sides (in finite dimensions $(\mathcal{W}^\perp)^\perp = \mathcal{W}$). □

2.8 Subspace Properties

- $\text{colsp}(\boldsymbol{A}) = \text{range}(\boldsymbol{A})$
- $\text{rowsp}(\boldsymbol{A}) = \text{range}(\boldsymbol{A}^T)$
- $\dim(\text{colsp}\,\boldsymbol{A}) = \dim(\text{rowsp}\,\boldsymbol{A}) = \text{rank}(\boldsymbol{A})$
- $\boldsymbol{A}$ is full rank if $\text{rank}(\boldsymbol{A}) = \min\{m, n\}$
- $\text{rank}(\boldsymbol{A}) = \text{rank}(\boldsymbol{A}^T) = \text{rank}(\boldsymbol{A}^*)$
- $\text{rank}(\boldsymbol{A}^*\boldsymbol{A}) = \text{rank}(\boldsymbol{A})$
- pre or post multiplication by non-singular matrices leaves the rank unchanged

2.9 Singular Matrices

A square matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is non-singular (invertible) if,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0} \implies \boldsymbol{x} = \boldsymbol{0}$$

(this is the definition of non-singularity used in this course).

For $\boldsymbol{A}$ to be non-singular is equivalent to:

- $\text{rank}(\boldsymbol{A}) = n$
- $\exists! \boldsymbol{A}^{-1} \in \mathbb{C}^{n \times n}$ such that $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$
- $\det(\boldsymbol{A}) \neq 0$
- $\text{nullity}(\boldsymbol{A}) = 0$
- $\text{null}(\boldsymbol{A}) = \{\boldsymbol{0}\}$
- Rows and columns of $\boldsymbol{A}$ are linearly independent
- The linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ has a unique solution for every $\boldsymbol{b} \in \mathbb{C}^n$

If $\boldsymbol{A}$ is non-singular then:

- $(\boldsymbol{A}^{-1})^T = \left(\boldsymbol{A}^T\right)^{-1} = \boldsymbol{A}^{-T}$
- $(\boldsymbol{A}^{-1})^* = (\boldsymbol{A}^*)^{-1} = \boldsymbol{A}^{-*}$

2.10 Moore–Penrose Pseudo-Inverse

For any matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ there exists a unique pseudo-inverse $\boldsymbol{A}^\dagger \in \mathbb{C}^{n \times m}$ that satisfies the following 4 conditions:

- $\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{A}$
- $\boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}^\dagger$
- $(\boldsymbol{A}\boldsymbol{A}^\dagger)^* = \boldsymbol{A}\boldsymbol{A}^\dagger$
- $(\boldsymbol{A}^\dagger\boldsymbol{A})^* = \boldsymbol{A}^\dagger\boldsymbol{A}$

In certain cases the pseudo-inverse can be found explicitly as follows:

- If $\boldsymbol{A}$ is invertible: $\boldsymbol{A}^\dagger = \boldsymbol{A}^{-1}$
- If $\boldsymbol{A}$ is full-column rank: $\boldsymbol{A}^\dagger = (\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*$ (and $\boldsymbol{A}^\dagger\boldsymbol{A} = \boldsymbol{I}$ acts as a left inverse)
- If $\boldsymbol{A}$ is full-row rank: $\boldsymbol{A}^\dagger = \boldsymbol{A}^*(\boldsymbol{A}\boldsymbol{A}^*)^{-1}$ (and $\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{I}$ acts as a right inverse)

(It is easy to verify that each of these satisfies the Moore–Penrose conditions.)

Properties:

- $(\boldsymbol{A}^\dagger)^\dagger = \boldsymbol{A}$
- $(\boldsymbol{A}^\dagger)^T = (\boldsymbol{A}^T)^\dagger$
- $(\boldsymbol{A}^\dagger)^* = (\boldsymbol{A}^*)^\dagger$
- In general $(\boldsymbol{A}\boldsymbol{B})^\dagger \neq \boldsymbol{B}^\dagger\boldsymbol{A}^\dagger$

3 Norms

3.1 Vector Norms

Norms capture the notions of size and distance in a vector space. Approximations and convergence in numerical linear algebra are measured using norms.

Let $\mathcal{V}$ be a vector space over $\mathbb{F}$ (where $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$). A norm is a function $\|\cdot\| : \mathcal{V} \to [0, \infty)$ satisfying,

- Triangle inequality: $\|\boldsymbol{u} + \boldsymbol{v}\| \leq \|\boldsymbol{u}\| + \|\boldsymbol{v}\|$
- Absolute homogeneity: $\|a\boldsymbol{u}\| = |a|\,\|\boldsymbol{u}\|$
- Positive definiteness: $\|\boldsymbol{u}\| = 0 \iff \boldsymbol{u} = \boldsymbol{0}$

A vector space with a norm is called a normed vector space.

Property:

- Reverse triangle inequality: $\Big|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|\Big| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$
  Proof.

$$\begin{aligned} \boldsymbol{x} = \boldsymbol{y} + \boldsymbol{x} - \boldsymbol{y} &\implies \|\boldsymbol{x}\| \leq \|\boldsymbol{y}\| + \|\boldsymbol{x} - \boldsymbol{y}\| \implies \|\boldsymbol{x} - \boldsymbol{y}\| \geq \|\boldsymbol{x}\| - \|\boldsymbol{y}\| \\ \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{y} - \boldsymbol{x} &\implies \|\boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y} - \boldsymbol{x}\| \implies \|\boldsymbol{x} - \boldsymbol{y}\| \geq -(\|\boldsymbol{x}\| - \|\boldsymbol{y}\|) \end{aligned}$$

□

$$\therefore \Big|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|\Big| \leq \|\boldsymbol{x} - \boldsymbol{y}\|$$

3.2 Vector p-norms

For vectors $\boldsymbol{x} \in \mathbb{C}^n$, the ℓ_p norm is typically denoted $\|\boldsymbol{x}\|_p$ and is given by,

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

for $p \in [1, \infty)$.

(notice that we require $p \geq 1$ since for $p < 1$ the triangle inequality would not hold).

Special p-norms:

- ℓ_1 is called the Manhattan norm
- ℓ_2 is called the Euclidean norm (the most common norm)
- ℓ_∞ is called the max norm

(not proved in the course, but you can show that $\|\boldsymbol{x}\|_\infty = \lim_{p\to\infty}\|\boldsymbol{x}\|_p = \max_i |x_i|$, i.e. the component of $\boldsymbol{x}$ that has the maximum absolute value).

3.3 Cauchy–Schwarz Inequality

For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{C}^d$,

$$|\langle \boldsymbol{x}, \boldsymbol{y}\rangle| \leq \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2$$

with equality achieved if $\boldsymbol{x}$ and $\boldsymbol{y}$ are linearly dependent (that is $\boldsymbol{x} = \alpha\boldsymbol{y}$ for some $\alpha \in \mathbb{C}$).

3.4 Hölder's Inequality

For $p, q \in [1, \infty]$ satisfying,

$$\frac{1}{p} + \frac{1}{q} = 1$$

We have that,

$$|\,\langle \boldsymbol{x}, \boldsymbol{y}\rangle\,| \leq \|\boldsymbol{x}\|_p \|\boldsymbol{y}\|_q$$

(for $p = 2$ this reduces to the Cauchy–Schwarz inequality). For $1 < p < \infty$, equality holds when the vectors $(|x_1|^p, \ldots, |x_n|^p)$ and $(|y_1|^q, \ldots, |y_n|^q)$ are linearly dependent and the products $\overline{x_i} y_i$ all have the same argument. (Unlike Cauchy–Schwarz, $\boldsymbol{x}$ and $\boldsymbol{y}$ being linearly dependent is not enough when $p \neq 2$.)

3.5 Unitary Invariance of the Euclidean Norm

Let $\boldsymbol{U} \in \mathbb{C}^{n\times n}$ be a unitary matrix. Then,

$$\|\boldsymbol{U}\boldsymbol{x}\| = \|\boldsymbol{x}\|$$

Proof.

$$\|\boldsymbol{U}\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{U}\boldsymbol{x}, \boldsymbol{U}\boldsymbol{x}\rangle} = \sqrt{(\boldsymbol{U}\boldsymbol{x})^*(\boldsymbol{U}\boldsymbol{x})} = \sqrt{\boldsymbol{x}^*\boldsymbol{U}^*\boldsymbol{U}\boldsymbol{x}} = \sqrt{\boldsymbol{x}^*\boldsymbol{x}} = \|\boldsymbol{x}\| \qquad \square$$

(technically $\boldsymbol{U}$ doesn't strictly need to be unitary but it just needs to have orthonormal columns so this will also hold for non-square matrices).

3.6 Entry-Wise Matrix Norm

The entry-wise matrix norm is the norm that you would get if you flatten out a matrix into a vector and then just take a regular ℓ_p norm.

$$\|\boldsymbol{A}\|_p = \|\mathrm{vec}(\boldsymbol{A})\|_p = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |\boldsymbol{A}_{ij}|^p\right)^{\frac{1}{p}}$$

The associated ℓ_2 norm is called the Frobenius norm,

$$\begin{aligned}\|\boldsymbol{A}\|_F &= \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |\boldsymbol{A}_{ij}|^2\right)^{\frac{1}{2}} \\ &= \sqrt{\mathrm{tr}(\boldsymbol{A}^*\boldsymbol{A})}\end{aligned}$$

And in a similar vein the Frobenius inner product is defined to be,

$$\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \mathrm{tr}(\boldsymbol{A}^*\boldsymbol{B})$$

3.7 Unitary Invariance of the Frobenius Norm

Given any matrix $\boldsymbol{U} \in \mathbb{C}^{m\times n}$ with orthonormal columns, we have,

$$\|\boldsymbol{UA}\|_F = \|\boldsymbol{A}\|_F$$

Proof.

$$\|\boldsymbol{UA}\|_F = \sqrt{\operatorname{tr}\left((\boldsymbol{UA})^*(\boldsymbol{UA})\right)} = \sqrt{\operatorname{tr}\left(\boldsymbol{A}^*\boldsymbol{U}^*\boldsymbol{UA}\right)} = \sqrt{\operatorname{tr}(\boldsymbol{A}^*\boldsymbol{A})} = \|\boldsymbol{A}\|_F \qquad \square$$

3.8 Induced Matrix Norm

An alternative matrix norm is the matrix norm induced by vector norms.

Consider a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Given any vector norm $\|\cdot\|_p$ for some $p \geq 1$, the corresponding induced matrix norm is defined as,

$$\|\boldsymbol{A}\|_p \coloneqq \max_{\substack{\boldsymbol{x}\in\mathbb{C}^n \\ \|\boldsymbol{x}\|_p=1}} \|\boldsymbol{Ax}\|_p = \max_{\boldsymbol{x}\in\mathbb{C}^n\setminus\{\boldsymbol{0}\}} \frac{\|\boldsymbol{Ax}\|_p}{\|\boldsymbol{x}\|_p}$$

Essentially this measures the maximum factor by which the matrix $\boldsymbol{A}$ can stretch a vector from the unit p-ball.

Properties:

- $\|\boldsymbol{Ax}\|_p \leq \|\boldsymbol{A}\|_p\|\boldsymbol{x}\|_p$ (follows trivially)
- $\|\boldsymbol{AB}\|_p \leq \|\boldsymbol{A}\|_p\|\boldsymbol{B}\|_p$ (known as the sub-multiplicativity property)

When $p = 2$ the induced 2-norm is often called the **spectral** norm. In this course the spectral norm is the "default" matrix norm, so if you see $\|\boldsymbol{A}\|$ it probably means the spectral norm.

3.9 Unitary Invariance of the Spectral Norm

Given any matrix $\boldsymbol{U} \in \mathbb{C}^{m\times n}$ with orthonormal columns,

$$\|\boldsymbol{UA}\|_2 = \|\boldsymbol{A}\|_2$$

Proof.

$$\|\boldsymbol{UA}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{UAx}\|_2 = \max_{\|\boldsymbol{x}\|_2=1} \|\boldsymbol{Ax}\|_2 = \|\boldsymbol{A}\|_2 \qquad \square$$

3.10 Condition Number

Given a linear system,

$$\boldsymbol{Ax} = \boldsymbol{b}$$

An important question is to ask how sensitive $\boldsymbol{x}$ is to small changes in $\boldsymbol{b}$. Suppose $\boldsymbol{b}$ is perturbed by $\delta\boldsymbol{b}$ such that

$$\widehat{\boldsymbol{b}} = \boldsymbol{b} + \delta\boldsymbol{b}$$

Let $\widehat{\boldsymbol{x}}$ be the corresponding vector such that $\boldsymbol{A}\widehat{\boldsymbol{x}} = \widehat{\boldsymbol{b}}$. Then,

$$\left.\begin{aligned}\boldsymbol{Ax} &= \boldsymbol{b} \\ \boldsymbol{A}\widehat{\boldsymbol{x}} &= \widehat{\boldsymbol{b}}\end{aligned}\right\} \implies \boldsymbol{A}(\widehat{\boldsymbol{x}} - \boldsymbol{x}) = (\widehat{\boldsymbol{b}} - \boldsymbol{b}) \implies (\widehat{\boldsymbol{x}} - \boldsymbol{x}) = \boldsymbol{A}^{-1}(\widehat{\boldsymbol{b}} - \boldsymbol{b})$$

Using the induced matrix norm this implies that,

$$\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_p \leq \|\boldsymbol{A}^{-1}\|_p \, \|\widehat{\boldsymbol{b}} - \boldsymbol{b}\|_p$$

We also have that,

$$\|\boldsymbol{b}\|_p \leq \|\boldsymbol{A}\|_p \, \|\boldsymbol{x}\|_p$$

And putting these two together it implies that,

$$\frac{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_p}{\|\boldsymbol{x}\|_p} \leq \underbrace{\|\boldsymbol{A}\|_p \, \|\boldsymbol{A}^{-1}\|_p}_{\kappa_p(\boldsymbol{A})} \frac{\|\widehat{\boldsymbol{b}} - \boldsymbol{b}\|_p}{\|\boldsymbol{b}\|_p}$$

So essentially this is saying that the normalised deviations of $\boldsymbol{x}$ are bounded by the normalised deviations in $\boldsymbol{b}$ times a constant that is known as the condition number, $\kappa_p(\boldsymbol{A})$, of the matrix $\boldsymbol{A}$.

A matrix $\boldsymbol{A}$ is said to be:

- Well conditioned if $\kappa_p(\boldsymbol{A})$ is "small"
- Ill conditioned if $\kappa_p(\boldsymbol{A})$ is "large"

Property: $\kappa_p(\boldsymbol{A}) \geq 1$,

Proof.

$$1 = \|\boldsymbol{I}\|_p = \|\boldsymbol{A}\boldsymbol{A}^{-1}\|_p \leq \|\boldsymbol{A}\|_p \, \|\boldsymbol{A}^{-1}\|_p$$

□

Property: if $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ is unitary then $\kappa(\boldsymbol{A}) = 1$

Proof.

$$\begin{aligned}\kappa(\boldsymbol{A}) &= \|\boldsymbol{A}\|\|\boldsymbol{A}^{-1}\| = \|\boldsymbol{A}\|\|\boldsymbol{A}^*\| = \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\| \times \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{A}^*\boldsymbol{x}\| \\ &= \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{x}\| \times \max_{\|\boldsymbol{x}\|=1} \|\boldsymbol{x}\| = 1 \times 1 = 1\end{aligned}$$

□

4 Eigenvalues and Eigenvectors

Let $\boldsymbol{A} \in \mathbb{C}^{n \times n}$. Suppose there exists $\lambda \in \mathbb{C}$ and $\boldsymbol{v} \in \mathbb{C}^n \backslash \{\boldsymbol{0}\}$ such that,

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

Then λ is called an eigenvalue of $\boldsymbol{A}$ and $\boldsymbol{v}$ is called an eigenvector of $\boldsymbol{A}$. The pair $(\lambda, \boldsymbol{v})$ is called an eigenpair of $\boldsymbol{A}$.

Notes:

- Eigenpairs are *only* defined for square matrices
  - otherwise $\boldsymbol{A}\boldsymbol{v}$ and $\lambda\boldsymbol{v}$ would not even be the same size
- The eigenvector $\boldsymbol{v}$ can never be the zero vector
  - otherwise this would hold trivially for all λ

Finding eigenpairs:

$$\boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v} \qquad \Longrightarrow \qquad (\boldsymbol{A} - \lambda \boldsymbol{I})\boldsymbol{v} = \boldsymbol{0}$$

Given that $\boldsymbol{v} \neq \boldsymbol{0}$ the only solution to this system is if $\boldsymbol{A} - \lambda \boldsymbol{I}$ is singular. This happens when the expression has determinant zero. Notice that the determinant of $\boldsymbol{A} - \lambda \boldsymbol{I}$ is a polynomial in λ and its roots will be the eigenvalues.

$$\det(\boldsymbol{A} - \lambda \boldsymbol{I}) = p_n(\lambda) = \sum_{k=0}^{n} c_k \lambda^k$$

By the fundamental theorem of algebra, there will be n solutions to the characteristic polynomial in the complex numbers and so $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ will have exactly n eigenvalues (not necessarily all unique).

(if you are working over the real numbers only, though, there may be fewer than n eigenvalues).

4.1 Characteristic Polynomial

We define the characteristic polynomial $p_n(t)$ of an n-by-n matrix $\boldsymbol{A}$ by,

$$p_n(t) = \det(\boldsymbol{A} - t\boldsymbol{I})$$

As was described above, we know that when $t = \lambda$ the characteristic polynomial is zero (i.e. its roots are at the eigenvalues of $\boldsymbol{A}$).

$$p_n(t) = 0 \iff t \text{ is an eigenvalue of } \boldsymbol{A}$$

Since p_n is a polynomial of degree n, we know by the fundamental theorem of algebra that there will be n solutions $(\lambda_1, \ldots, \lambda_n)$ and so the characteristic polynomial can be factorised as follows,

$$p_n(t) = \prod_{k=1}^{n} (\lambda_k - t) = (-1)^n \prod_{k=1}^{n} (t - \lambda_k)$$

Setting $t = 0$ gives a very nice result:

- $\det(\boldsymbol{A}) = p_n(0) = \prod_{k=1}^{n} \lambda_k$

(following a similar logic you can also prove that the trace is equal to the sum of the eigenvalues, but that proof is a bit harder to follow so it is left out here).

4.2 Spectrum & Spectral Radius

The spectrum of $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is the set of all eigenvalues of $\boldsymbol{A}$,

$$\begin{aligned}\operatorname{spec}(\boldsymbol{A}) &= \{\lambda \in \mathbb{C} : \exists \boldsymbol{v} \neq \boldsymbol{0},\ \boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}\} \\ &= \{\lambda \in \mathbb{C} : p_n(\lambda) = 0\}\end{aligned}$$

The spectral radius of $\boldsymbol{A}$ is the largest absolute value of its eigenvalues,

$$\rho(\boldsymbol{A}) \coloneqq \max_{\lambda \in \operatorname{spec}(\boldsymbol{A})} |\lambda|$$

4.3 Properties

First some relatively trivial properties (proofs not included):

1. If $(\lambda, \boldsymbol{v})$ is an eigenpair of $\boldsymbol{A}$ then so is $(\lambda, c\boldsymbol{v})$ for any $c \in \mathbb{C} \setminus \{0\}$
2. If $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ and $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$ then $\overline{\boldsymbol{A}}\overline{\boldsymbol{v}} = \overline{\lambda}\overline{\boldsymbol{v}}$
    - For real matrices this implies that if $(\lambda, \boldsymbol{v})$ is an eigenpair then so is $(\overline{\lambda}, \overline{\boldsymbol{v}})$.
3. If $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ and λ is an eigenvalue of $\boldsymbol{A}$ then $\overline{\lambda}$ is an eigenvalue of $\boldsymbol{A}^*$

Some other properties:

1. $\operatorname{spec}(\boldsymbol{A}) = \operatorname{spec}(\boldsymbol{A}^T)$
2. $\operatorname{spec}(\overline{\boldsymbol{A}}) = \operatorname{spec}(\boldsymbol{A}^*)$
3. In general $\operatorname{spec}(\boldsymbol{A}) \neq \operatorname{spec}(\boldsymbol{A}^*)$ but we do have that $\rho(\boldsymbol{A}) = \rho(\boldsymbol{A}^*)$

Property 1 follows trivially from the properties of the determinant. Then property 2 follows from property 1. The last property follows from property 3 in the first list: the eigenvalues of $\boldsymbol{A}^*$ are the conjugates of the eigenvalues of $\boldsymbol{A}$, and $|\overline{\lambda}| = |\lambda|$.

4.4 Matrix Polynomials

Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$. A matrix polynomial of degree k is defined as,

$$p_k(\boldsymbol{A}) = \sum_{i=0}^{k} a_i \boldsymbol{A}^i \qquad \text{for } a_i \in \mathbb{C}$$

(also note that by definition we let $\boldsymbol{A}^0 = \boldsymbol{I}$).

Note that polynomial factorisation also applies to matrix polynomials whereby,

$$p_k(\boldsymbol{A}) = a_k \prod_{i=1}^{k} (\boldsymbol{A} - \beta_i \boldsymbol{I})$$

where $\beta_1, \ldots, \beta_k$ are the roots of $p_k(t) = \sum_{i=0}^{k} a_i t^i$.

4.5 Spectral Mapping Theorem for Matrix Polynomials

Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ and let,

$$p_k(t) = \sum_{i=0}^{k} a_i t^i$$

Then,

- If $(\lambda, \boldsymbol{v})$ is an eigenpair of $\boldsymbol{A}$ then $(p_k(\lambda), \boldsymbol{v})$ is an eigenpair of $p_k(\boldsymbol{A})$
- Conversely, every eigenvalue of $p_k(\boldsymbol{A})$ is of the form $p_k(\lambda)$ for some $\lambda \in \text{spec}(\boldsymbol{A})$. So $\text{spec}(p_k(\boldsymbol{A})) = \{p_k(\lambda) : \lambda \in \text{spec}(\boldsymbol{A})\}$ (but an eigenvector of $p_k(\boldsymbol{A})$ need not be an eigenvector of $\boldsymbol{A}$)

Proof of the first statement:

Proof. Suppose $(\lambda, \boldsymbol{v})$ is an eigenpair of $\boldsymbol{A}$,

$$\implies \boldsymbol{A}\boldsymbol{v} = \lambda \boldsymbol{v}$$

We want to show that $p_k(\boldsymbol{A})\boldsymbol{v} = p_k(\lambda)\boldsymbol{v}$.

$$\begin{aligned} p_k(\boldsymbol{A})\boldsymbol{v} &= \left(\sum_{i=0}^{k} a_i \boldsymbol{A}^i \right) \boldsymbol{v} \\ &= \sum_{i=0}^{k} a_i \boldsymbol{A}^i \boldsymbol{v} \\ &= \sum_{i=0}^{k} a_i \lambda^i \boldsymbol{v} \\ &= \left(\sum_{i=0}^{k} a_i \lambda^i \right) \boldsymbol{v} \\ &= p_k(\lambda)\boldsymbol{v} \end{aligned}$$

In the proof I use the fact that,

$$\boldsymbol{A}^i \boldsymbol{v} = \boldsymbol{A}^{i-1}\boldsymbol{A}\boldsymbol{v} = \boldsymbol{A}^{i-1}\lambda \boldsymbol{v} = \cdots = \lambda^i \boldsymbol{v}$$

□

4.6 Theorem: $\boldsymbol{A}$ is singular $\iff 0 \in \text{spec}(\boldsymbol{A})$

Proof.

$$\begin{aligned} \boldsymbol{A} \text{ is singular} &\iff \exists \boldsymbol{v} \neq \boldsymbol{0},\ \boldsymbol{A}\boldsymbol{v} = \boldsymbol{0} \\ &\iff \exists \boldsymbol{v} \neq \boldsymbol{0},\ \boldsymbol{A}\boldsymbol{v} = 0\boldsymbol{v} \\ &\iff 0 \in \text{spec}(\boldsymbol{A}) \end{aligned}$$

(an alternative proof uses the fact that the determinant is equal to the product of the eigenvalues, so a zero determinant is only achieved if $0 \in \text{spec}(\boldsymbol{A})$). □

4.7 Theorem

For $\lambda, \mu \in \mathbb{C}$,

$$\lambda \in \text{spec}(\boldsymbol{A}) \iff \lambda + \mu \in \text{spec}(\boldsymbol{A} + \mu\boldsymbol{I})$$

Proof.

$$\begin{aligned} \lambda \in \text{spec}(\boldsymbol{A}) &\iff \det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0 \\ &\iff \det(\boldsymbol{A} + \mu\boldsymbol{I} - \mu\boldsymbol{I} - \lambda\boldsymbol{I}) = 0 \\ &\iff \det(\boldsymbol{A} + \mu\boldsymbol{I} - (\mu + \lambda)\boldsymbol{I}) = 0 \\ &\iff \lambda + \mu \in \text{spec}(\boldsymbol{A} + \mu\boldsymbol{I}) \end{aligned}$$

(the proof in the lectures was slightly different but the same idea). □

4.8 Theorem: Hermitian $\implies$ eigenvalues are real

Proof. This follows from the property that if λ is an eigenvalue of $\boldsymbol{A}$ then $\overline{\lambda}$ will be an eigenvalue of $\boldsymbol{A}^*$, which is shown explicitly below.

Recall that $\boldsymbol{A}$ is Hermitian if $\boldsymbol{A} = \boldsymbol{A}^*$. Suppose λ is an eigenvalue of $\boldsymbol{A}$ then,

$$\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0$$

We can now use the property that $\det(\boldsymbol{B}^*) = \overline{\det(\boldsymbol{B})}$,

$$\begin{aligned} \det((\boldsymbol{A} - \lambda\boldsymbol{I})^*) &= \overline{\det(\boldsymbol{A} - \lambda\boldsymbol{I})} \\ \det(\boldsymbol{A}^* - \overline{\lambda}\boldsymbol{I}) &= \overline{0} \\ &= 0 \end{aligned}$$

And so $\overline{\lambda}$ is an eigenvalue of $\boldsymbol{A}^*$.

Since $\boldsymbol{A} = \boldsymbol{A}^*$ this implies that,

$$\lambda = \overline{\lambda} \qquad \implies \qquad \lambda \in \mathbb{R}$$ □

4.9 Eigenvalues of Triangular Matrices

For triangular matrices, computing the eigenvalues is easy since they will just be equal to the diagonal entries,

$$\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = \prod_{i=1}^{n}(a_{ii} - \lambda) \quad \implies \quad \text{spec}(\boldsymbol{A}) = \{a_{11}, \ldots, a_{nn}\}$$

(note that diagonal matrices are also included in this category, since they are both upper triangular and lower triangular).

4.10 Similarity Transformations

Since finding the eigenvalues of diagonal or triangular matrices is much easier, a natural question is:

- Is there a way to transform a matrix so that it looks diagonal or triangular *without* changing its eigenvalues?

The answer to this is the concept of similarity transformations.

A matrix $\boldsymbol{A}$ is said to be **similar** to $\boldsymbol{B}$ if there exists a non-singular matrix $\boldsymbol{S}$ such that,

$$\boldsymbol{A} = \boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{S}$$

An important property is that:

- $\text{spec}(\boldsymbol{A}) = \text{spec}(\boldsymbol{B})$

Proof.

$$\begin{aligned}
\det(\boldsymbol{A} - \lambda\boldsymbol{I}) &= \det(\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{S} - \lambda\boldsymbol{I}) \\
&= \det(\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{S} - \lambda\boldsymbol{S}^{-1}\boldsymbol{I}\boldsymbol{S}) \\
&= \det\left(\boldsymbol{S}^{-1}(\boldsymbol{B} - \lambda\boldsymbol{I})\boldsymbol{S}\right) \\
&= \det(\boldsymbol{S}^{-1})\det(\boldsymbol{B} - \lambda\boldsymbol{I})\det(\boldsymbol{S}) \\
&= \frac{1}{\det(\boldsymbol{S})}\det(\boldsymbol{B} - \lambda\boldsymbol{I})\det(\boldsymbol{S}) \\
&= \det(\boldsymbol{B} - \lambda\boldsymbol{I})
\end{aligned}$$

Which implies that $\boldsymbol{A}$ and $\boldsymbol{B}$ have the same characteristic polynomials. □

Also note that this notion of similarity is only defined for square matrices.

(Don't worry too much about whether it should be $\boldsymbol{S}^{-1}\boldsymbol{B}\boldsymbol{S}$ or $\boldsymbol{S}\boldsymbol{B}\boldsymbol{S}^{-1}$: either way the two matrices are similar. Different matrix decompositions put the inverse on the first or on the last factor, so it is more important to remember what each matrix represents in a particular decomposition and which one is inverted. Thinking back to how the decomposition is constructed usually makes this clear.)

4.11 Diagonalisable Matrix

Since diagonal matrices are the simplest, we may be interested in knowing which matrices are similar to diagonal matrices.

$\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is said to be diagonalisable if it is similar to a diagonal matrix. I.e. if there exists a diagonal matrix $\boldsymbol{D}$ such that,

$$\boldsymbol{A} = \boldsymbol{S}^{-1}\boldsymbol{D}\boldsymbol{S}, \qquad \text{for some non-singular matrix } \boldsymbol{S}$$

Theorem: a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is diagonalisable if and only if it has n linearly independent eigenvectors.

Proof. You can prove this using a constructive argument. Suppose $\boldsymbol{A}$ has n linearly independent eigenvectors $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ with $\boldsymbol{A}\boldsymbol{v}_i = \lambda_i\boldsymbol{v}_i$, and let

$$\boldsymbol{V} = (\boldsymbol{v}_1|\ldots|\boldsymbol{v}_n) \qquad \text{and} \qquad \boldsymbol{D} = \text{diag}(\lambda_1, \ldots, \lambda_n)$$

Then, looking at the matrix product one column at a time,

$$\boldsymbol{AV} = (\boldsymbol{Av}_1|\dots|\boldsymbol{Av}_n) = (\lambda_1\boldsymbol{v}_1|\dots|\lambda_n\boldsymbol{v}_n) = \boldsymbol{VD}$$

Since the columns of $\boldsymbol{V}$ are linearly independent, $\boldsymbol{V}$ is non-singular and so $\boldsymbol{A} = \boldsymbol{VDV}^{-1}$. That is, $\boldsymbol{A}$ is similar to the diagonal matrix $\boldsymbol{D}$ (with $\boldsymbol{S} = \boldsymbol{V}^{-1}$).

(this is not technically a full proof since it only shows one way of the biconditional) □

(looking at this construction also explains why matrix diagonalisation is typically written as $\boldsymbol{A} = \boldsymbol{VDV}^{-1}$ instead of $\boldsymbol{A} = \boldsymbol{V}^{-1}\boldsymbol{DV}$. It's just about what the matrix $\boldsymbol{V}$ represents here).

4.12 Eigendecomposition

A very important matrix decomposition is the eigendecomposition also known as

- eigenvalue decomposition (EVD)
- spectral decomposition

It essentially follows from the proof of the theorem above, and it states that any matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ with n linearly independent eigenvectors can be diagonalised as follows,

$$\boldsymbol{A} = \boldsymbol{VDV}^{-1}$$

where,

$$\begin{aligned} \boldsymbol{V} &= (\boldsymbol{v}_1|\dots|\boldsymbol{v}_n) && \text{is a matrix of the eigenvectors of } \boldsymbol{A} \\ \boldsymbol{D} &= \operatorname{diag}(\lambda_1,\dots,\lambda_n) && \text{is a diagonal matrix of the eigenvalues of } \boldsymbol{A} \end{aligned}$$

Pros:

- Almost every matrix in $\mathbb{C}^{n\times n}$ is diagonalisable (apart from a set of Lebesgue measure zero)

Cons:

- Only works on square matrices
- Computing $\boldsymbol{V}^{-1}$ can be computationally heavy and error prone if $\boldsymbol{V}$ is ill conditioned

4.13 Unitary Diagonalisation

We can address one of the cons mentioned above by looking for matrices that are unitarily similar to a diagonal matrix. In other words a matrix is unitarily diagonalisable if it can be decomposed as follows,

$$\boldsymbol{A} = \boldsymbol{UDU}^*$$

Theorem: a matrix $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is unitarily diagonalisable if and only if it is normal (i.e. $\boldsymbol{AA}^* = \boldsymbol{A}^*\boldsymbol{A}$)

(proof not covered).

Pros:

- Unitary matrices are nice to work with

Cons:

- Only works on matrices that are normal (which is a subset of square matrices)

5 Singular Value Decomposition (SVD)

Any matrix of any size can be "diagonalised" through appropriate pre and post multiplication by unitary matrices.

That is for any $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ it can be decomposed as,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

where,

- $\boldsymbol{U}$ is an $m \times m$ unitary matrix
- $\boldsymbol{\Sigma}$ is an $m \times n$ rectangular diagonal matrix
- $\boldsymbol{V}^*$ is an $n \times n$ unitary matrix

Some wording to know:

- The diagonal entries of $\boldsymbol{\Sigma}$ are called the singular values of $\boldsymbol{A}$
- Columns of $\boldsymbol{U}$ are called the left-singular vectors of $\boldsymbol{A}$
- Columns of $\boldsymbol{V}$ are called the right-singular vectors of $\boldsymbol{A}$

Properties:

- To make $\boldsymbol{\Sigma}$ unique the convention is to arrange the singular values in decreasing order from largest to smallest.
- If $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ then all 3 matrices in the decomposition will also be real

5.1 Constructing the SVD

The numerical approach to constructing the SVD is actually quite straightforward. Suppose we have some matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ that we want to decompose into,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

- the left-singular vectors are a set of orthonormal eigenvectors of $\boldsymbol{A}\boldsymbol{A}^*$
- the right-singular vectors are a set of orthonormal eigenvectors of $\boldsymbol{A}^*\boldsymbol{A}$
- the singular values are the square roots of the eigenvalues of both $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$

Notes:

- $\boldsymbol{A}\boldsymbol{A}^*$ is a normal matrix $\implies$ unitarily diagonalisable $\implies$ $\boldsymbol{U}$ exists
- $\boldsymbol{A}^*\boldsymbol{A}$ is a normal matrix $\implies$ unitarily diagonalisable $\implies$ $\boldsymbol{V}$ exists
- $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$ are both Hermitian $\implies$ have real eigenvalues
- $\boldsymbol{A}\boldsymbol{A}^*$ and $\boldsymbol{A}^*\boldsymbol{A}$ will have the same non-zero eigenvalues

Proof of that last property:

Proof. Suppose $\lambda \neq 0$ is an eigenvalue of $\boldsymbol{A}\boldsymbol{A}^*$,

$$\begin{aligned} \exists \boldsymbol{v} \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{A}^*\boldsymbol{v} &= \lambda \boldsymbol{v} \\ \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{A}^*\boldsymbol{v} &= \lambda \boldsymbol{A}^*\boldsymbol{v} \\ \boldsymbol{A}^*\boldsymbol{A}(\boldsymbol{A}^*\boldsymbol{v}) &= \lambda(\boldsymbol{A}^*\boldsymbol{v}) \end{aligned}$$

Here $\boldsymbol{A}^*\boldsymbol{v} \neq \boldsymbol{0}$, since otherwise $\lambda\boldsymbol{v} = \boldsymbol{A}\boldsymbol{A}^*\boldsymbol{v} = \boldsymbol{0}$ would force $\lambda = 0$. This implies that λ is an eigenvalue of $\boldsymbol{A}^*\boldsymbol{A}$ (with eigenvector $\boldsymbol{A}^*\boldsymbol{v}$). The same argument with the roles of $\boldsymbol{A}$ and $\boldsymbol{A}^*$ swapped gives the other direction. □

5.2 Properties of SVD

Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$ be the SVD of $\boldsymbol{A} \in \mathbb{C}^{m\times n}$. Suppose that $\boldsymbol{A}$ has r non-zero singular values. Then,

- $\text{rank}(\boldsymbol{A}) = r$
- $\|\boldsymbol{A}\| = \sigma_1$ (i.e. the spectral norm is equal to the largest singular value)

Compact SVD: Suppose $r \leq \min\{m, n\}$. We can discard the zero-entries in $\boldsymbol{\Sigma}$ to get $\boldsymbol{A} = \boldsymbol{U}_r\boldsymbol{\Sigma}_r\boldsymbol{V}_r^*$ where,

- $\boldsymbol{U}_r \in \mathbb{C}^{m\times r}$
- $\boldsymbol{\Sigma}_r \in \mathbb{C}^{r\times r}$
- $\boldsymbol{V}_r^* \in \mathbb{C}^{r\times n}$

Whilst the compact SVD saves space and still preserves $\boldsymbol{A}$ exactly, it does lose some information that may be useful. The full SVD gives orthonormal bases for the fundamental subspaces, for example:

- $\text{range}(\boldsymbol{A}) = \text{span}\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r\}$
- $\text{null}(\boldsymbol{A}) = \text{span}\{\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_n\}$

and the compact SVD throws away the vectors $\boldsymbol{v}_{r+1}, \ldots, \boldsymbol{v}_n$ (and $\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m$) needed for bases like the second. Application to finding the pseudoinverse:

- $\boldsymbol{A}^\dagger = \boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^*$
- $\|\boldsymbol{A}^\dagger\| = \frac{1}{\sigma_r}$

5.3 Schatten Norm

A third type of matrix norm is the Schatten norm. For $p \geq 1$ the Schatten norm of a matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ is defined by applying the vector p-norm to the singular values of $\boldsymbol{A}$. I.e.,

$$\|\boldsymbol{A}\|_p = \left(\sum_{i=1}^{\text{rank}(\boldsymbol{A})} \sigma_i^p\right)^{\frac{1}{p}}$$

Properties:

- All Schatten norms are sub-multiplicative
- All Schatten norms are unitarily invariant
- The most famous cases are $p = 1, 2, \infty$
  - $p = 1$ is known as the "nuclear norm" or the "trace norm"
  - $p = 2$ coincides with the Frobenius norm
  - $p = \infty$ coincides with the spectral norm (i.e. the ℓ_2 induced norm)

5.4 Eckart–Young–Mirsky Theorem

Let $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ with an SVD given by,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

Notice that one way to expand out this matrix multiplication is as follows,

$$\boldsymbol{A} = \sum_{i=1}^{\text{rank}(\boldsymbol{A})} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$$

The Eckart–Young–Mirsky theorem tells us that for any $k \in \{1, \ldots, \operatorname{rank}(\boldsymbol{A})\}$ the matrix,

$$\boldsymbol{A}_k \coloneqq \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^*$$

is the best rank-k approximation to $\boldsymbol{A}$ in terms of both the spectral norm and the Frobenius norm.

5.5 Application to Condition Numbers

Recall that the condition number κ of a matrix $\boldsymbol{A}$ is given by,

$$\kappa(\boldsymbol{A}) = \|\boldsymbol{A}\|\|\boldsymbol{A}^\dagger\|$$

If the spectral norm is used then it turns out that,

$$\kappa(\boldsymbol{A}) = \frac{\sigma_1}{\sigma_r}$$

where σ_r is the smallest non-zero singular value (since $\|\boldsymbol{A}^\dagger\| = 1/\sigma_r$). For a square non-singular $\boldsymbol{A}$ we have $r = n$, so this is the familiar $\kappa(\boldsymbol{A}) = \sigma_1/\sigma_n$.

6 Special Matrices

6.1 Diagonal Matrices

Properties:

- $\operatorname{spec}(\boldsymbol{D}) = \{d_{11}, d_{22}, \ldots, d_{nn}\}$
- $\det(\boldsymbol{D}) = \prod_{i=1}^{n} d_{ii}$
- $\boldsymbol{D}$ is non-singular iff $d_{ii} \neq 0$ for all i.
- $\operatorname{rank}(\boldsymbol{D})$ = number of non-zero diagonal entries

Left multiplication by a diagonal matrix will scale the rows. Right multiplication will scale the columns.

Inverse of a diagonal matrix $\operatorname{diag}(d_{11}, d_{22}, \ldots, d_{nn})$ is $\operatorname{diag}(\frac{1}{d_{11}}, \frac{1}{d_{22}}, \ldots, \frac{1}{d_{nn}})$.

6.2 Block-Diagonal

Similar idea to diagonal but the diagonals can be "blocks".

$$\boldsymbol{D} = \begin{pmatrix} \boldsymbol{D}_{11} & & & \\ & \boldsymbol{D}_{22} & & \\ & & \ddots & \\ & & & \boldsymbol{D}_{kk} \end{pmatrix}$$

Properties:

- $\operatorname{spec}(\boldsymbol{D}) = \bigcup_{i=1}^{k} \operatorname{spec}(\boldsymbol{D}_{ii})$
- $\det(\boldsymbol{D}) = \prod_{i=1}^{k} \det(\boldsymbol{D}_{ii})$

- $\boldsymbol{D}$ is non-singular iff each $\boldsymbol{D}_{ii}$ is non-singular
- $\text{rank}(\boldsymbol{D}) = \sum_{i=1}^{k} \text{rank}(\boldsymbol{D}_{ii})$

Like with above:

- Left multiplication by a block-diagonal scales the rows as blocks
- Right multiplication by a block-diagonal scales the columns as blocks

Inverse matrix,

$$\boldsymbol{D}^{-1} = \begin{pmatrix} \boldsymbol{D}_{11}^{-1} & & & \\ & \boldsymbol{D}_{22}^{-1} & & \\ & & \ddots & \\ & & & \boldsymbol{D}_{kk}^{-1} \end{pmatrix}$$

6.3 Triangular

Two types:

- Upper triangular: everything strictly below the main diagonal is zero
- Lower triangular: everything strictly above the main diagonal is zero

Properties:

- $\text{spec}(\boldsymbol{T}) = \{t_{11}, t_{22}, \ldots, t_{nn}\}$
- $\det(\boldsymbol{T}) = \prod_{i=1}^{n} t_{ii}$
- $\boldsymbol{T}$ is non-singular iff $t_{ii} \neq 0$ for all i
- $\text{rank}(\boldsymbol{T}) \geq$ the number of non-zero diagonal entries

Sparsity patterns:

- The inverse of a non-singular upper (lower) triangular matrix is upper (lower) triangular
- Upper triangular $\times$ upper triangular = upper triangular (and the same for lower triangular)

6.4 Block-Triangular

Same idea as the block-diagonal. Here the matrix is triangular if you partition it into blocks. So you can think of the matrix as being made up of several blocks and these blocks as a whole should make a triangular matrix.

Properties:

- $\text{spec}(\boldsymbol{T}) = \bigcup_{i=1}^{k} \text{spec}(\boldsymbol{T}_{ii})$
- $\det(\boldsymbol{T}) = \prod_{i=1}^{k} \det(\boldsymbol{T}_{ii})$
- $\boldsymbol{T}$ is non-singular iff each $\boldsymbol{T}_{ii}$ is non-singular
- $\text{rank}(\boldsymbol{T}) \geq \sum_{i=1}^{k} \text{rank}(\boldsymbol{T}_{ii})$

(sparsity patterns are preserved in the same way as with the triangular matrices but with respect to the blocks).

6.5 Upper/Lower Hessenberg

A Hessenberg matrix is "almost" triangular: it has one extra diagonal, making it not quite triangular. An upper Hessenberg matrix $\boldsymbol{H}$ has $h_{ij} = 0$ for $i > j + 1$ (everything below the first sub-diagonal is zero). For example, when $n = 4$,

$$\boldsymbol{H} = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{pmatrix}$$

A lower Hessenberg matrix is the transpose of this pattern.

Properties:

- Said to be "unreduced" if *all* the sub-diagonal entries are non-zero
- If an upper Hessenberg matrix is unreduced then its rank is $\geq n - 1$, since the first $n - 1$ columns are clearly independent

6.6 Permutation Matrices

These are matrices where exactly one entry in each row and column is equal to 1 and all other entries are 0.

Properties (suppose $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is a permutation matrix):

- Left multiplication: $\boldsymbol{PA}$ permutes the rows of $\boldsymbol{A}$
- Right multiplication: $\boldsymbol{AP}$ permutes the columns of $\boldsymbol{A}$
- $\boldsymbol{PP}^T = \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$ (i.e. $\boldsymbol{P}$ is orthogonal)
- $\det(\boldsymbol{P}) = \pm 1$ (follows from being real orthogonal)

6.7 Projection Matrices

A matrix $\boldsymbol{P} \in \mathbb{C}^{n \times n}$ is a projection/idempotent matrix if

$$\boldsymbol{P} = \boldsymbol{P}^2$$

Properties:

- If $\boldsymbol{v} \in \mathcal{R}(\boldsymbol{P})$ then $\boldsymbol{Pv} = \boldsymbol{v}$
- If $\boldsymbol{P}$ is a projection then so is $\boldsymbol{I} - \boldsymbol{P}$
- All eigenvalues are either 0 or 1

Proofs of the properties above:

Proof. 1. Suppose $\boldsymbol{v} \in \mathcal{R}(\boldsymbol{P})$. Then $\exists \boldsymbol{w}$ such that $\boldsymbol{Pw} = \boldsymbol{v}$.

$$\boldsymbol{Pv} = \boldsymbol{PPw} = \boldsymbol{P}^2\boldsymbol{w} = \boldsymbol{Pw} = \boldsymbol{v}$$

2. Suppose $\boldsymbol{P}$ is a projection,

$$(\boldsymbol{I} - \boldsymbol{P})^2 = \boldsymbol{I}^2 - 2\boldsymbol{IP} + \boldsymbol{P}^2 = \boldsymbol{I} - 2\boldsymbol{P} + \boldsymbol{P} = \boldsymbol{I} - \boldsymbol{P}$$

3. Suppose $(\lambda, \boldsymbol{v})$ is an eigenpair,

$$\boldsymbol{Pv} = \lambda\boldsymbol{v} \implies \boldsymbol{P}^2\boldsymbol{v} = \lambda\boldsymbol{Pv} \implies \boldsymbol{Pv} = \lambda^2\boldsymbol{v} \implies \lambda = \lambda^2 \implies \lambda \in \{0, 1\}$$

□

6.7.1 Orthogonal Projection Matrices

A matrix $\boldsymbol{P} \in \mathbb{C}^{n\times n}$ is an orthogonal projection if:

- $\boldsymbol{P} = \boldsymbol{P}^2$
- $\boldsymbol{P} = \boldsymbol{P}^*$

(i.e. it is a projection matrix and it is Hermitian).

Important: despite the name, an orthogonal projection matrix is *not* (in general) an orthogonal matrix. The term orthogonal here comes from the fact that $\mathcal{R}(\boldsymbol{P}) \perp \mathcal{R}(\boldsymbol{I}-\boldsymbol{P})$.

Properties:

- $\mathcal{R}(\boldsymbol{P}) \perp \mathcal{R}(\boldsymbol{I}-\boldsymbol{P})$
- Pythagoras' theorem: $\|\boldsymbol{v}\|^2 = \|\boldsymbol{P}\boldsymbol{v}\|^2 + \|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}\|^2$

Proofs of the above:

Proof. 1. Take $\boldsymbol{x} \in \mathcal{R}(\boldsymbol{P})$ and $\boldsymbol{y} \in \mathcal{R}(\boldsymbol{I}-\boldsymbol{P})$. Then $\exists \boldsymbol{u}, \boldsymbol{v}$ such that $\boldsymbol{P}\boldsymbol{u} = \boldsymbol{x}$ and $(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v} = \boldsymbol{y}$.

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \langle \boldsymbol{P}\boldsymbol{u}, (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{P}^*(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}\rangle = \langle \boldsymbol{u}, (\boldsymbol{P}^*-\boldsymbol{P}^*\boldsymbol{P})\boldsymbol{v}\rangle = \langle \boldsymbol{u}, \boldsymbol{0}\rangle = 0$$

since $\boldsymbol{P}^* - \boldsymbol{P}^*\boldsymbol{P} = \boldsymbol{P} - \boldsymbol{P}^2 = \boldsymbol{0}$.

2. This follows from the standard Pythagoras' theorem since,

$$\|\boldsymbol{v}\|^2 = \|\boldsymbol{P}\boldsymbol{v} + (\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}\|^2 = \|\boldsymbol{P}\boldsymbol{v}\|^2 + \|(\boldsymbol{I}-\boldsymbol{P})\boldsymbol{v}\|^2 \quad \text{(by orthogonality)}$$

□

As the name suggests, an orthogonal projection matrix will orthogonally project vectors into the range of $\boldsymbol{P}$ (which is going to be a lower dimensional space if $\boldsymbol{P}$ is not full rank). The reason we call it an orthogonal projection is because the error vector $\boldsymbol{x} - \boldsymbol{P}\boldsymbol{x}$ is perpendicular to every vector in the range of $\boldsymbol{P}$. This follows from the fact that $\mathcal{R}(\boldsymbol{P}) \perp \mathcal{R}(\boldsymbol{I}-\boldsymbol{P})$.

6.7.2 Constructing an Orthogonal Projection Matrix

Let $\boldsymbol{v} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$. Then,

$$\boldsymbol{P} = \frac{\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^2}$$

is the orthogonal projection matrix onto (the line spanned by) the vector $\boldsymbol{v}$.

Proof. This formula arises quite naturally if you make a drawing. Draw some vector $\boldsymbol{v}$ and then another vector $\boldsymbol{w}$ pointing in a different direction and try to visualise what the orthogonal projection of $\boldsymbol{w}$ onto $\boldsymbol{v}$ is. You should be able to come up with the following formula,

$$\text{projection} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\|\boldsymbol{w}\|\cos(\theta) = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\|\boldsymbol{w}\|\frac{\langle \boldsymbol{v}, \boldsymbol{w}\rangle}{\|\boldsymbol{v}\|\|\boldsymbol{w}\|} = \frac{\boldsymbol{v}\boldsymbol{v}^*\boldsymbol{w}}{\|\boldsymbol{v}\|^2} = \boldsymbol{P}\boldsymbol{w}$$

For completeness, you can also verify that $\boldsymbol{P}$ as defined is an orthogonal projection matrix since,

$$\boldsymbol{P}^2 = \frac{\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^2}\frac{\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v}(\boldsymbol{v}^*\boldsymbol{v})\boldsymbol{v}^*}{\|\boldsymbol{v}\|^4} = \frac{\|\boldsymbol{v}\|^2\,\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^4} = \frac{\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^2} = \boldsymbol{P}$$

$$\boldsymbol{P}^* = \frac{(\boldsymbol{v}\boldsymbol{v}^*)^*}{\|\boldsymbol{v}\|^2} = \frac{\boldsymbol{v}\boldsymbol{v}^*}{\|\boldsymbol{v}\|^2} = \boldsymbol{P}$$

□

More generally, given any matrix $\boldsymbol{Q} \in \mathbb{C}^{m\times n}$ with orthonormal columns, $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^*$ is the unique orthogonal projection onto $\mathcal{R}(\boldsymbol{Q})$.

Proof.

- First prove that $\boldsymbol{P}$ is a projection matrix

$$\boldsymbol{P}^2 = \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{Q}\boldsymbol{Q}^* = \boldsymbol{Q}\boldsymbol{I}\boldsymbol{Q}^* = \boldsymbol{Q}\boldsymbol{Q}^* = \boldsymbol{P}$$

- Next prove that $\boldsymbol{P}$ is Hermitian

$$\boldsymbol{P}^* = (\boldsymbol{Q}\boldsymbol{Q}^*)^* = \boldsymbol{Q}\boldsymbol{Q}^* = \boldsymbol{P}$$

- Prove that $\boldsymbol{P}$ maps onto $\mathcal{R}(\boldsymbol{Q})$ (trivial)
- Prove that $\boldsymbol{P}$ is unique: suppose $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are both orthogonal projections onto the same subspace $\mathcal{S}$. For any $\boldsymbol{x}$ we can write $\boldsymbol{x} = \boldsymbol{P}_i\boldsymbol{x} + (\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{x}$ where $\boldsymbol{P}_i\boldsymbol{x} \in \mathcal{S}$ and $(\boldsymbol{I} - \boldsymbol{P}_i)\boldsymbol{x} \in \mathcal{S}^\perp$ (since $\mathcal{R}(\boldsymbol{I} - \boldsymbol{P}_i) \perp \mathcal{R}(\boldsymbol{P}_i) = \mathcal{S}$). Splitting $\boldsymbol{x}$ into a part in $\mathcal{S}$ and a part in $\mathcal{S}^\perp$ can only be done in one way (if $\boldsymbol{s}_1 + \boldsymbol{t}_1 = \boldsymbol{s}_2 + \boldsymbol{t}_2$ then $\boldsymbol{s}_1 - \boldsymbol{s}_2 = \boldsymbol{t}_2 - \boldsymbol{t}_1 \in \mathcal{S} \cap \mathcal{S}^\perp = \{\boldsymbol{0}\}$), so $\boldsymbol{P}_1\boldsymbol{x} = \boldsymbol{P}_2\boldsymbol{x}$ for every $\boldsymbol{x}$.

□

Even more generally, for an arbitrary matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, $\boldsymbol{P} = \boldsymbol{A}\boldsymbol{A}^\dagger$ is the unique orthogonal projection onto $\mathcal{R}(\boldsymbol{A})$.

Proof. These follow easily from the pseudo-inverse properties.

- First prove that $\boldsymbol{P}$ is a projection matrix

$$\boldsymbol{P}^2 = \boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{P}$$

- Next prove that $\boldsymbol{P}$ is Hermitian

$$\boldsymbol{P}^* = (\boldsymbol{A}\boldsymbol{A}^\dagger)^* = \boldsymbol{A}\boldsymbol{A}^\dagger = \boldsymbol{P}$$

- Prove that $\boldsymbol{P}$ maps onto $\mathcal{R}(\boldsymbol{A})$ (trivial)

□

6.8 Positive Definite Matrices

Positive-definiteness is a generalisation of the concept of positive numbers in matrices (more formally it provides a partial ordering to matrices).

Let $\boldsymbol{A} \in \mathbb{C}^{n\times n}$:

- PD: $\boldsymbol{A} \succ \boldsymbol{0} \iff \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle > 0$ for all $\boldsymbol{x} \in \mathbb{C}^n \setminus \{\boldsymbol{0}\}$
- PSD: $\boldsymbol{A} \succeq \boldsymbol{0} \iff \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle \geq 0$ for all $\boldsymbol{x} \in \mathbb{C}^n$

(negative definite and negative semi-definite matrices are defined similarly, but are less commonly used).

Properties:

- PSD $\implies$ Hermitian

(the proof is not that hard but it is not very nice, so it is omitted here; see slide 15 of Lecture 6).

- For any matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$, we have $\boldsymbol{A}^*\boldsymbol{A} \succeq \boldsymbol{0}$ and $\boldsymbol{A}\boldsymbol{A}^* \succeq \boldsymbol{0}$

(this follows since $\langle \boldsymbol{x}, \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x}\rangle = \|\boldsymbol{A}\boldsymbol{x}\|^2 \geq 0$, and similarly for $\boldsymbol{A}\boldsymbol{A}^*$).

Furthermore:

- If $\text{rank}(\boldsymbol{A}) = n$ then $\boldsymbol{A}^*\boldsymbol{A} \succ \boldsymbol{0}$
- If $\text{rank}(\boldsymbol{A}) = m$ then $\boldsymbol{A}\boldsymbol{A}^* \succ \boldsymbol{0}$

(if $\text{rank}(\boldsymbol{A}) = n$ then $\boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{0}$ for all $\boldsymbol{x} \neq \boldsymbol{0}$, so $\langle \boldsymbol{x}, \boldsymbol{A}^*\boldsymbol{A}\boldsymbol{x}\rangle = \|\boldsymbol{A}\boldsymbol{x}\|^2 > 0$. The second statement is the same argument applied to $\boldsymbol{A}^*$).

More properties:

- $\boldsymbol{A} \succ \boldsymbol{0} \implies \lambda_i(\boldsymbol{A}) > 0 \; \forall i$
- $\boldsymbol{A} \succeq \boldsymbol{0} \implies \lambda_i(\boldsymbol{A}) \geq 0 \; \forall i$
- $\boldsymbol{A} \prec \boldsymbol{0} \implies \lambda_i(\boldsymbol{A}) < 0 \; \forall i$
- $\boldsymbol{A} \preceq \boldsymbol{0} \implies \lambda_i(\boldsymbol{A}) \leq 0 \; \forall i$
- For $\boldsymbol{A} \succ \boldsymbol{0}$ the EVD and SVD coincide
- Consequently the eigenvalues and singular values coincide
- Every PD matrix is invertible and its inverse is also PD
- For $\boldsymbol{A}, \boldsymbol{B} \succ \boldsymbol{0}$ and $\alpha > 0$ we have $\alpha\boldsymbol{A} \succ \boldsymbol{0}$ and $\boldsymbol{A} + \boldsymbol{B} \succ \boldsymbol{0}$

(there are even more properties on slide 17 of Lecture 6).

Matrix square root:

- $\boldsymbol{A} \succeq \boldsymbol{0} \implies \exists! \boldsymbol{B} \succeq \boldsymbol{0}$ s.t. $\boldsymbol{B}^2 = \boldsymbol{A}$

6.8.1 Matrix Partial Ordering

This notion of positive-definiteness induces a matrix partial ordering where,

$$\boldsymbol{A} \succ \boldsymbol{B} \iff \boldsymbol{A} - \boldsymbol{B} \succ \boldsymbol{0}$$

$$\boldsymbol{A} \succeq \boldsymbol{B} \iff \boldsymbol{A} - \boldsymbol{B} \succeq \boldsymbol{0}$$

A few interesting properties (for Hermitian $\boldsymbol{A}$ and $\boldsymbol{B}$, with the eigenvalues of each sorted in the same order):

- If $\boldsymbol{A} \succeq \boldsymbol{B}$ then $\lambda_i(\boldsymbol{A}) \geq \lambda_i(\boldsymbol{B})$
- If $\boldsymbol{A} \succeq \boldsymbol{B}$ and $\boldsymbol{A} \neq \boldsymbol{B}$ then $\exists i$ such that $\lambda_i(\boldsymbol{A}) > \lambda_i(\boldsymbol{B})$
- If $\boldsymbol{A} \succ \boldsymbol{B}$ then $\lambda_i(\boldsymbol{A}) > \lambda_i(\boldsymbol{B})$

6.9 Diagonally Dominant Matrices

A matrix is (row-wise) diagonally dominant (DD) if:

- in every row, the magnitude of the diagonal entry is greater than or equal to the sum of the magnitudes of all the other entries in that row, i.e. $|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|$ for all i

A matrix is strictly diagonally dominant (SDD) if this inequality is strict.

Properties:

- All SDD matrices are non-singular (Lévy–Desplanques theorem)
- If $\boldsymbol{A}$ is Hermitian and diagonally dominant with non-negative diagonals then $\boldsymbol{A}$ is positive semi-definite
- If $\boldsymbol{A}$ is Hermitian and strictly diagonally dominant with positive diagonals then $\boldsymbol{A}$ is positive definite

(the converse of the last two statements however is not true).

7 Matrix Factorisations

So far we have already seen two:

- EVD
- SVD

However there are many other factorisations.

7.1 LU Factorisation

Given a square matrix $\boldsymbol{A}$ we can decompose it as,

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$

where $\boldsymbol{L}$ is lower triangular and $\boldsymbol{U}$ is upper triangular.

There are some problems with LU:

- It only works for square matrices
- Even then it may not always exist
- It can be numerically unstable

How to find an LU factorisation:

- The key is Gaussian elimination, which algorithmically gets you the answer (see Section 8).

7.2 LDU Factorisation

Suppose $\boldsymbol{A}$ admits an $\boldsymbol{LU}$ decomposition. We can convert the $\boldsymbol{LU}$ decomposition into an $\boldsymbol{LDU}$ where $\boldsymbol{D}$ is a diagonal matrix.

This is not complicated (see slide 5 of Lecture 7 to see it visually).

Advantages:

- $\boldsymbol{L}$, $\boldsymbol{D}$ and $\boldsymbol{U}$ are now unique (when $\boldsymbol{A}$ is non-singular and $\boldsymbol{L}$ and $\boldsymbol{U}$ have ones on their diagonals)

7.3 PLU Factorisation

Recall the problems with the LU factorisation mentioned above. It turns out that for any square matrix $\boldsymbol{A}$ you can find a specific permutation of rows such that the LU factorisation exists and is numerically stable.

Finding this specific permutation is beyond the scope of this course, but suppose that $\boldsymbol{P}$ is a permutation matrix that achieves this desired permutation. This implies that any square matrix $\boldsymbol{A}$ can be decomposed as,

$$\boldsymbol{P}\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$$

7.4 Cholesky Factorisation

For square matrices $\boldsymbol{A} \succ \boldsymbol{0}$, the LU factorisation always exists and is referred to as Cholesky Factorisation.

The Cholesky Factorisation theorem states that for any matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$,

- $\boldsymbol{A}$ is positive semidefinite if and only if there is a lower triangular matrix $\boldsymbol{L}$ with non-negative diagonal entries such that $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^*$
- Similarly $\boldsymbol{A}$ is positive definite if and only if there is a lower triangular matrix $\boldsymbol{L}$ with positive diagonal entries such that $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^*$ and $\boldsymbol{L}$ is unique
- Furthermore, if $\boldsymbol{A}$ is real then there exists an $\boldsymbol{L}$ that is also real

This decomposition is similar to LU decomposition but it only works for positive definite or semidefinite matrices and decomposes them into,

$$\boldsymbol{A} = \boldsymbol{L}\boldsymbol{L}^*$$

Things to note:

- All the stability issues vanish for Cholesky Factorisation.

7.5 QR Factorisation

Applicable to any matrix $\boldsymbol{A} \in \mathbb{C}^{m\times n}$ where we can decompose $\boldsymbol{A}$ as,

$$\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$$

where:

- $\boldsymbol{Q} \in \mathbb{C}^{m\times m}$ is a unitary matrix
- $\boldsymbol{R} \in \mathbb{C}^{m\times n}$ is a non-square upper triangular matrix

How to find a QR factorisation:

- This can be done using the Gram–Schmidt process

When $m > n$, the last $m - n$ rows of $\boldsymbol{R}$ are zero, so they (and the last $m - n$ columns of $\boldsymbol{Q}$) can be dropped to form a "reduced" or "thin" QR factorisation.

See the lecture slides for more details (starting from slide 9). It is worth looking into this because there are some nice properties of the QR factorisation. For example it lets you find basis vectors for $\mathcal{R}(\boldsymbol{A})$ and $\mathcal{N}(\boldsymbol{A}^*)$.

8 Linear Systems (Direct Methods)

8.1 Direct vs Indirect Methods

Generally speaking in numerical linear algebra algorithms can be categorised into two main groups.

Direct Methods:

- The procedure takes a finite number of steps
- On completion one has the exact solution (up to rounding errors)
- No useful approximation if one stops early
- Not possible to "warm-start"

Indirect Methods:

- The solution is updated iteratively
- Some methods may take an infinite number of steps
- Solution can often be approximated if one stops early

- Possible to "warm-start"

This section is going to focus on the problem of linear systems from the perspective of direct methods.

8.2 Linear Systems

The most fundamental problem in linear algebra is solving linear systems and these sorts of systems arise very frequently in all areas of mathematics and science.

Suppose you have some system,

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \quad \boldsymbol{A} \in \mathbb{C}^{n\times n},\ \boldsymbol{b} \in \mathbb{C}^n$$

The system may have:

- Unique solution (if $\boldsymbol{A}$ is non-singular)
- Infinitely many solutions (if $\boldsymbol{A}$ is singular and $\boldsymbol{b} \in \mathcal{R}(\boldsymbol{A})$)
- No solutions (if $\boldsymbol{A}$ is singular and $\boldsymbol{b} \notin \mathcal{R}(\boldsymbol{A})$)

Here we are going to focus on when $\boldsymbol{A}$ is non-singular. In the cases where $\boldsymbol{A}$ is singular you may instead want to look into least-squares methods.

8.3 When $\boldsymbol{A}$ is non-singular

The main method here is to perform Gaussian elimination (essentially an LU factorisation).

Note that it was mentioned previously that LU factorisations can be numerically unstable. Therefore a more robust method is to do a PLU factorisation.

Algorithm for solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ using PLU factorisation:

1. Perform partial pivoting: $\tilde{\boldsymbol{A}} = \boldsymbol{P}\boldsymbol{A}$
2. Permute the right-hand side: $\tilde{\boldsymbol{b}} = \boldsymbol{P}\boldsymbol{b}$
3. Calculate LU factorisation: $\tilde{\boldsymbol{A}} = \boldsymbol{L}\boldsymbol{U}$
4. Forward solve: $\boldsymbol{L}\boldsymbol{y} = \tilde{\boldsymbol{b}}$
5. Backward solve: $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$

To give a bit more detail on the last two steps (forward/backward solve): after step 3 we have the linear system $\boldsymbol{L}\boldsymbol{U}\boldsymbol{x} = \tilde{\boldsymbol{b}}$. To solve this we can first solve $\boldsymbol{L}\boldsymbol{y} = \tilde{\boldsymbol{b}}$ and then use that result to solve $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$. Solving $\boldsymbol{L}\boldsymbol{y} = \tilde{\boldsymbol{b}}$ is called forward solving: since $\boldsymbol{L}$ is lower triangular, the first equation only involves y_1, the second only y_1 and y_2, and so on, so it is very easy to do by substituting values in from top to bottom. Similarly, solving $\boldsymbol{U}\boldsymbol{x} = \boldsymbol{y}$ is just substituting values in from bottom to top, and is called "backward" solving.

Note on algorithm:

- we only need to perform the PLU factorisation once and can then use it multiple times for solving systems with different right-hand sides $\boldsymbol{b}$

8.3.1 If $\boldsymbol{A}$ is non-singular, why not just compute $\boldsymbol{A}^{-1}$?

Notice that an alternative way to solve the system would be to just find $\boldsymbol{A}^{-1}$.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \quad \Longrightarrow \quad \boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}$$

Whilst this may be simpler mathematically, computationally it is much more work. The standard approach to find $\boldsymbol{A}^{-1}$ is to solve the system,

$$\boldsymbol{AX} = \boldsymbol{I}$$

To do this you need to:

- First perform a PLU factorisation on $\boldsymbol{A}$
- Then use the factorisation to solve the n linear systems $\boldsymbol{Ax}_i = \boldsymbol{e}_i$ for $i = 1, \ldots, n$
- Then multiply $\boldsymbol{b}$ by the resulting inverse matrix

This is many times slower than just solving the $\boldsymbol{Ax} = \boldsymbol{b}$ system directly as described above. On top of this there are even more problems such as the fact that:

- If $\boldsymbol{A}$ is sparse (which happens often in applications) then $\boldsymbol{A}^{-1}$ is typically dense, requiring a huge amount of storage
- Numerical round-off errors tend to be magnified when computing $\boldsymbol{A}^{-1}$

This is why computing $\boldsymbol{A}^{-1}$ is the first cardinal sin of numerical linear algebra.

9 Least Squares (Direct Methods)

9.1 Ordinary Least Squares

What happens when you have a linear system $\boldsymbol{Ax} = \boldsymbol{b}$ but $\boldsymbol{b} \notin \mathcal{R}(\boldsymbol{A})$?

Then there is no solution to the system but we may still want to find the value $\boldsymbol{x}$ that makes the difference between $\boldsymbol{Ax}$ and $\boldsymbol{b}$ as small as possible. This is the Ordinary Least Squares (OLS) problem.

Given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and vector $\boldsymbol{b} \in \mathbb{R}^m$ then the OLS solution is the vector $\widehat{\boldsymbol{x}} \in \mathbb{R}^n$ such that,

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x} \in \mathbb{R}^n}{\arg\min} \|\boldsymbol{Ax} - \boldsymbol{b}\|^2$$

(notice that for simplicity we are focussing on $\mathbb{R}$ but everything here also works for complex numbers).

9.2 Application: Polynomial Fitting

In general OLS can be used to perform any sort of "linear regression". Be careful with the word "linear" here though. This does not mean that you are fitting a straight line to the data but rather that the model is linear in its parameters. So for example you can use OLS to fit polynomials to data.

Suppose you have m data points (t_i, b_i) and put the b_i into a vector $\boldsymbol{b} = (b_1, \ldots, b_m)^T$. To fit a polynomial of degree n we need to solve the following linear system,

$$\underbrace{\begin{pmatrix} 1 & t_1 & \ldots & t_1^n \\ 1 & t_2 & \ldots & t_2^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_m & \ldots & t_m^n \end{pmatrix}}_{\boldsymbol{A}\text{: Vandermonde Matrix}} \underbrace{\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}}_{\boldsymbol{x}} = \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}}_{\boldsymbol{b}}$$

Note here that $\boldsymbol{A}$ depends on the choice of basis vectors for the polynomials. The Vandermonde matrix uses the standard monomial basis but in practice it can be numerically unstable and so there are many alternatives that are more frequently used.

9.3 Data Fitting in Other Norms

Also note that there are situations where you may want to minimise the residuals with respect to other norms (other than the standard ℓ_2 norm).

Minimising with ℓ_1 norm:

- called the least absolute deviation
- more robust to outliers
- (related to the median while the ℓ_2 norm is related to the mean)

Minimising with ℓ_∞ norm:

- when you want to minimise the worst case error
- used in min-max problems

These two formulations can be solved using linear programming (covered in MATH3202).

9.4 Finding the OLS

To solve the OLS we need to find the value $\boldsymbol{x}$ that minimises $f(\boldsymbol{x})$ where,

$$f(\boldsymbol{x}) \coloneqq \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$$

(note that we added the $\frac{1}{2}$ for convenience. This has no effect on the minimiser).

It will be shown later that $f(\boldsymbol{x})$ is convex and so the minimiser $\widehat{\boldsymbol{x}}$ is the value such that,

$$\boldsymbol{\nabla} f(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$$

(f is convex because its Hessian is $\nabla^2 f(\boldsymbol{x}) = \boldsymbol{A}^T\boldsymbol{A} \succeq \boldsymbol{0}$; see the second-order condition for convexity in Section 20).

To solve this first we can manipulate the expression for $f(\boldsymbol{x})$,

$$\begin{aligned}
f(\boldsymbol{x}) &= \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2 \\
&= \frac{1}{2}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\
&= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}^T - \boldsymbol{b}^T)(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) \\
&= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b}) \\
&= \frac{1}{2}(\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}^T\boldsymbol{b}) \\
&= \frac{1}{2}\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x} + \frac{1}{2}\boldsymbol{b}^T\boldsymbol{b} \\
&= \frac{1}{2}\left\langle \boldsymbol{x}, \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x}\right\rangle - \left\langle \boldsymbol{A}^T\boldsymbol{b}, \boldsymbol{x}\right\rangle + \frac{1}{2}\boldsymbol{b}^T\boldsymbol{b}
\end{aligned}$$

(notice in the working that I combine $\boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{b}$ and $\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{x}$ since they are both scalars and a scalar is equal to its own transpose)

Now to compute the gradient vector we use two useful facts in matrix calculus:

- If $f(\boldsymbol{x}) = \langle \boldsymbol{h}, \boldsymbol{x}\rangle$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = \boldsymbol{h}$
- If $f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{H}\boldsymbol{x}\rangle$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = (\boldsymbol{H} + \boldsymbol{H}^T)\boldsymbol{x}$ (assuming $\boldsymbol{H}$ is square)

(Notice the parallels with the univariate case. You can prove these by computing all of the partial derivatives explicitly and then matching the result. The first fact is quite easy; the second is a fair bit more tedious, but not hard. As an extra note, the second rule implies that if $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{x}$ then $\boldsymbol{\nabla} f(\boldsymbol{x}) = 2\boldsymbol{x}$, just like the univariate derivative of x^2).

Using those rules above we can compute $\boldsymbol{\nabla} f(\boldsymbol{x})$,

$$\begin{aligned}\boldsymbol{\nabla} f(\boldsymbol{x}) &= \frac{1}{2}\Big(\boldsymbol{A}^T\boldsymbol{A} + (\boldsymbol{A}^T\boldsymbol{A})^T\Big)\boldsymbol{x} - \boldsymbol{A}^T\boldsymbol{b} \\ &= \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^T\boldsymbol{b}\end{aligned}$$

Hence we have to find $\widehat{\boldsymbol{x}}$ such that $\boldsymbol{\nabla} f(\widehat{\boldsymbol{x}}) = \boldsymbol{0}$. I.e.,

$$\boldsymbol{A}^T\boldsymbol{A}\widehat{\boldsymbol{x}} = \boldsymbol{A}^T\boldsymbol{b}$$

(this is called the normal equation).

9.5 Comments on the Normal Equation

You should be able to see a few intuitive results in the normal equation.

Firstly if $\boldsymbol{A}$ is non-singular then $\boldsymbol{A}^{-1}\boldsymbol{b}$ is clearly a solution,

$$\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{A}^{-1}\boldsymbol{b} = \boldsymbol{A}^T\boldsymbol{b}$$

This should make sense since if $\boldsymbol{A}$ is non-singular then the vector that minimises $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|$ is the vector $\boldsymbol{A}^{-1}\boldsymbol{b}$.

Next notice that if $\boldsymbol{A}^T\boldsymbol{A}$ is non-singular then a solution for $\widehat{\boldsymbol{x}}$ is to take,

$$\widehat{\boldsymbol{x}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$$

This is a formula that I have seen many times before, for example in MATH2001.

The cool thing is that even if $\boldsymbol{A}^T\boldsymbol{A}$ is singular the system will still have a solution. In fact there will always be a solution to this system, since $\boldsymbol{A}^T\boldsymbol{b} \in \mathcal{R}(\boldsymbol{A}^T) = \mathcal{R}(\boldsymbol{A}^T\boldsymbol{A})$ (the two ranges are equal because $\text{rank}(\boldsymbol{A}^T\boldsymbol{A}) = \text{rank}(\boldsymbol{A}^T)$ and one contains the other).

9.6 Theorem: Solution to the Normal Equations

A vector $\boldsymbol{x}$ solves the OLS problem (and thus the normal equations) if and only if it can be written as,

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{b} + (\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})\boldsymbol{y} \qquad \text{where } \boldsymbol{y} \in \mathbb{R}^n$$

Here:

- $\boldsymbol{A}^\dagger\boldsymbol{b}$ is referred to as the least-norm solution
- $(\boldsymbol{I} - \boldsymbol{A}^\dagger\boldsymbol{A})$ is an orthogonal projection onto $\mathcal{N}(\boldsymbol{A})$

Proof. First verify that $\widehat{\boldsymbol{x}} = \boldsymbol{A}^\dagger\boldsymbol{b}$ is a valid solution,

$$\begin{aligned}\boldsymbol{A}^T\boldsymbol{A}\widehat{\boldsymbol{x}} &= \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{b} \\ &= \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^\dagger)^T\boldsymbol{b} \quad \text{(one of the pseudo-inverse properties)} \\ &= (\boldsymbol{A}\boldsymbol{A}^\dagger\boldsymbol{A})^T\boldsymbol{b} \\ &= \boldsymbol{A}^T\boldsymbol{b}\end{aligned}$$

Next, adding the $(\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{y}$ component doesn't change $\boldsymbol{A}\boldsymbol{x}$, since

$$\boldsymbol{A}(\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{y} = (\boldsymbol{A} - \boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{y} = \boldsymbol{0}$$

so every vector of this form also solves the normal equations. Conversely, if $\boldsymbol{x}$ solves the normal equations then $\boldsymbol{A}^T\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{A}^\dagger \boldsymbol{b}) = \boldsymbol{0}$, so $\boldsymbol{x} - \boldsymbol{A}^\dagger \boldsymbol{b} \in \mathcal{N}(\boldsymbol{A}^T\boldsymbol{A}) = \mathcal{N}(\boldsymbol{A})$ (if $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0}$ then $\|\boldsymbol{A}\boldsymbol{z}\|^2 = \boldsymbol{z}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{z} = 0$). Since $\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A}$ is the orthogonal projection onto $\mathcal{N}(\boldsymbol{A})$, taking $\boldsymbol{y} = \boldsymbol{x} - \boldsymbol{A}^\dagger \boldsymbol{b}$ gives $(\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{y} = \boldsymbol{y}$, and so $\boldsymbol{x} = \boldsymbol{A}^\dagger \boldsymbol{b} + (\boldsymbol{I} - \boldsymbol{A}^\dagger \boldsymbol{A})\boldsymbol{y}$. (See slide 11 for the lecture's version.) □

Important property of the solution:

$$\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} \perp \mathcal{R}(\boldsymbol{A})$$

(this should make sense intuitively. The value $\boldsymbol{x}$ that minimises the residuals should give you a vector that is orthogonal to the range of $\boldsymbol{A}$. Try to think about it in just 2 dimensions where you have a line and a point and you need to find what vector from the line to the point has the smallest distance. It turns out that it will be when the vector is at right angles to the line. In fact if you draw this you can even see why this is a consequence of Euclidean geometry and not necessarily the case if you use different norms).

One other thing to note is that from this framework you can view $\boldsymbol{A}\boldsymbol{A}^\dagger \boldsymbol{b} = \boldsymbol{A}\widehat{\boldsymbol{x}}$ as the orthogonal projection of $\boldsymbol{b}$ onto $\mathcal{R}(\boldsymbol{A})$.

9.7 Solving the Normal Equation

In practice finding $\boldsymbol{A}^\dagger$ is slow and so in numerical linear algebra one typically uses other methods to solve the normal equation directly (linking to what was said previously about how computing $\boldsymbol{A}^{-1}$ is almost never a good idea).

There are 3 main direct methods that one can use:

Solve the Normal Equation:

- There are clever techniques to solve this directly.

QR Factorisation:

- This is the "standard" direct method
- More detail below

SVD Factorisation:

- This can be used to compute $\boldsymbol{A}^\dagger$
- It is more expensive but can be more robust

9.8 Why not use Cholesky?

Above when I listed the 3 main direct methods I did not include Cholesky but note that Cholesky could also be used. Recall that if $\boldsymbol{A}$ is full-column rank then $\boldsymbol{A}^T\boldsymbol{A} \succ \boldsymbol{0}$ and so Cholesky can be applied. However in practice it is not used due to numerical instability.

The algorithm would look something like this,

Algorithm:

- Let $\boldsymbol{M} = \boldsymbol{A}^T\boldsymbol{A}$ and use Cholesky to decompose it into $\boldsymbol{M} = \boldsymbol{L}\boldsymbol{L}^T$
- Normal equation is now $\boldsymbol{L}\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$

- Solve $\boldsymbol{L}\boldsymbol{y} = \boldsymbol{A}^T\boldsymbol{b}$ using forward substitution
- Solve $\boldsymbol{L}^T\boldsymbol{x} = \boldsymbol{y}$ using backward substitution

(these triangular decomposition techniques really just all use the fact that solving systems involving triangular matrices is really easy and can be done using forward/backward substitution).

The reason for the instability is because $\kappa(\boldsymbol{A}^T\boldsymbol{A}) = \kappa^2(\boldsymbol{A})$. This is proved explicitly on slide 17.

9.9 Solving Using QR Factorisation

Suppose $\boldsymbol{A}$ has full column rank and take its thin QR factorisation $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$ (so $\boldsymbol{Q} \in \mathbb{R}^{m\times n}$ has orthonormal columns and $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is upper triangular and non-singular). It then turns out that the OLS solution is given by,

$$\boldsymbol{x} = \boldsymbol{R}^{-1}\boldsymbol{Q}^T\boldsymbol{b}$$

(details on slide 17).

To perform this technique, never form $\boldsymbol{R}^{-1}$ explicitly. Instead:

- Solve $\boldsymbol{R}\boldsymbol{x} = \boldsymbol{Q}^T\boldsymbol{b}$ using backward substitution

(recall that $\boldsymbol{R}$ is upper triangular).

9.10 Solving Using SVD Factorisation

Using SVD you can form the pseudoinverse explicitly,

$$\boldsymbol{x} = \boldsymbol{A}^\dagger\boldsymbol{b} = \boldsymbol{V}_r\boldsymbol{\Sigma}_r^{-1}\boldsymbol{U}_r^T\boldsymbol{b}$$

9.11 Computing the QR Factorisation

The rest of the lecture (from slide 21) goes into methods to compute the QR factorisation, which aren't covered in detail here.

The main idea is that you can use Gram–Schmidt to compute the QR factorisation. The lecture then talks about the difference between the classical Gram–Schmidt and modified Gram–Schmidt algorithms.

10 Stationary Methods

Now we are moving on to looking at iterative algorithms.

10.1 Motivation

Suppose we need to solve $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ where $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$.

One method could be to find a function $\boldsymbol{g}(\boldsymbol{x})$ such that,

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0} \iff \boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x}$$

Then given an initial guess $\boldsymbol{x}_0$ you iterate using the following rule,

$$\boldsymbol{x}_{k+1} = \boldsymbol{g}(\boldsymbol{x}_k)$$

How do you pick $\boldsymbol{g}(\boldsymbol{x})$?

- Naïve choice: $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x})$

- Newton's method: $g(x) = x - \frac{f(x)}{f'(x)}$ (univariate case)

Here specifically we are trying to apply these techniques to problems of the form,

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$$

10.2 General Stationary Iterations

For the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ the main technique is to split up $\boldsymbol{A} = \boldsymbol{M} - \boldsymbol{N}$ in a way such that $\boldsymbol{M}$ is invertible. Hence,

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ (\boldsymbol{M} - \boldsymbol{N})\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{x} &= \boldsymbol{M}^{-1}(\boldsymbol{N}\boldsymbol{x} + \boldsymbol{b}) \\ &= \boldsymbol{M}^{-1}(\boldsymbol{M}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}) \\ &= \boldsymbol{x} + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) \\ \implies \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) &= \boldsymbol{0} \end{aligned}$$

Therefore solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is equivalent to solving $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}) = \boldsymbol{0}$. Then take $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{f}(\boldsymbol{x})$ and so the iterative algorithm is,

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{g}(\boldsymbol{x}_k) \\ &= \boldsymbol{x}_k + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \end{aligned}$$

10.3 How to Choose M?

There are two opposing goals in the choice of $\boldsymbol{M}$:

1. Choose $\boldsymbol{M}$ such that inverting is easy
2. Choose $\boldsymbol{M}$ such that $\boldsymbol{M}^{-1}$ is close to $\boldsymbol{A}^{-1}$

(the reason for the second goal is because if $\boldsymbol{M}^{-1}$ is close to $\boldsymbol{A}^{-1}$ then the convergence will generally be faster. For example if $\boldsymbol{M}^{-1} = \boldsymbol{A}^{-1}$ then it will only take one iteration to converge).

NB: We don't actually form $\boldsymbol{M}^{-1}$ explicitly. Rather we solve $\boldsymbol{M}\boldsymbol{y} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ and then use this result.

10.4 Basic Relaxation Methods

The basic idea is to break up a matrix $\boldsymbol{A}$ as,

$$\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{E} + \boldsymbol{F}$$

where,

- $\boldsymbol{D}$ is just a diagonal matrix with the diagonal entries coming from $\boldsymbol{A}$
- $\boldsymbol{E}$ is the part of $\boldsymbol{A}$ below the main diagonal
- $\boldsymbol{F}$ is the part of $\boldsymbol{A}$ above the main diagonal

This simple method actually leads to 3 important methods:

1. Jacobi Method
   - when $\boldsymbol{M} = \boldsymbol{D}$
2. Gauss–Seidel (GS) method
   - when $\boldsymbol{M} = \boldsymbol{D} + \boldsymbol{E}$
3. Successive Over-Relaxation (SOR)
   - when $\boldsymbol{M} = \omega^{-1}\boldsymbol{D} + \boldsymbol{E}$

10.5 Convergence of Basic Relaxation Methods

Let $\boldsymbol{x}^\star$ denote the true solution such that $\boldsymbol{A}\boldsymbol{x}^\star = \boldsymbol{b}$. From the recurrence relation we have,

$$\begin{aligned}
\boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k) \\
\boldsymbol{x}_{k+1} - \boldsymbol{x}^\star &= \boldsymbol{x}_k - \boldsymbol{x}^\star + \boldsymbol{M}^{-1}(\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{A}\boldsymbol{x}_k) \\
&= \boldsymbol{x}_k - \boldsymbol{x}^\star - \boldsymbol{M}^{-1}\boldsymbol{A}(\boldsymbol{x}_k - \boldsymbol{x}^\star) \\
&= (\boldsymbol{I} - \boldsymbol{M}^{-1}\boldsymbol{A})(\boldsymbol{x}_k - \boldsymbol{x}^\star) \\
&= (\boldsymbol{I} - \boldsymbol{M}^{-1}\boldsymbol{A})^{k+1}(\boldsymbol{x}_0 - \boldsymbol{x}^\star)
\end{aligned}$$

Let $\boldsymbol{e}_k \coloneqq \boldsymbol{x}_k - \boldsymbol{x}^\star$ be the error vector. Then the above relationship can be written as $\boldsymbol{e}_{k+1} = \boldsymbol{T}^{k+1}\boldsymbol{e}_0$, or equivalently,

$$\boldsymbol{e}_k = \boldsymbol{T}^k \boldsymbol{e}_0$$

where $\boldsymbol{T} = \boldsymbol{I} - \boldsymbol{M}^{-1}\boldsymbol{A}$ is called the iteration matrix.

It can be shown that,

$$\lim_{k\to\infty} \boldsymbol{e}_k = \boldsymbol{0} \text{ for every initial guess } \boldsymbol{x}_0 \iff \lim_{k\to\infty} \boldsymbol{T}^k = \boldsymbol{0}$$

(the $\impliedby$ direction follows directly from $\boldsymbol{e}_k = \boldsymbol{T}^k\boldsymbol{e}_0$. For the $\implies$ direction, taking $\boldsymbol{e}_0$ to be each standard basis vector in turn shows that every column of $\boldsymbol{T}^k$ tends to $\boldsymbol{0}$).

10.5.1 Sufficient Condition for Convergence

$$\|\boldsymbol{T}\| < 1 \implies \lim_{k\to\infty} \boldsymbol{e}_k = \boldsymbol{0}$$

where $\|\cdot\|$ is any induced norm.

Proof.

$$\begin{aligned}
\|\boldsymbol{e}_k\| &= \|\boldsymbol{T}^k\boldsymbol{e}_0\| \\
&\leq \|\boldsymbol{T}^k\|\|\boldsymbol{e}_0\| \\
&\leq \|\boldsymbol{T}\|^k\|\boldsymbol{e}_0\|
\end{aligned}$$

And so as $k \to \infty$ we have $\|\boldsymbol{T}\|^k \to 0$ (since $\|\boldsymbol{T}\| < 1$), and hence $\boldsymbol{e}_k \to \boldsymbol{0}$. □

10.5.2 Necessary and Sufficient Condition for Convergence

The previous condition above is still useful since it is very easy to check but it is not a necessary condition for convergence and so failing to meet it does not mean the algorithm will not converge. However here we give a necessary and sufficient condition,

$$\rho(\boldsymbol{T}) < 1 \iff \lim_{k\to\infty} \boldsymbol{e}_k = \boldsymbol{0} \text{ for every initial guess } \boldsymbol{x}_0$$

(proof not given in the course).

10.6 Non-Asymptotic Rate of Convergence

The end of the lecture (slides 16 to 18) covers a non-asymptotic rate of convergence, which isn't covered in these notes.

11 Polynomial Acceleration

For the whole of this section suppose $\boldsymbol{A} \succ \boldsymbol{0}$ and define,

$$M \coloneqq \lambda_{\max}(\boldsymbol{A}) \qquad m \coloneqq \lambda_{\min}(\boldsymbol{A})$$

11.1 Stationary Richardson Iteration

Recall that previously we talked about stationary methods where,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{M}^{-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k)$$

Perhaps the simplest stationary iteration method you could try is one of the form,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k)$$

where α is some constant (i.e. this is like taking $\boldsymbol{M}^{-1} = \alpha\boldsymbol{I}$).

On the assignment we prove that the optimal step size α is,

$$\alpha = \frac{2}{M + m}$$

And in such cases the convergence is given by,

$$\|\boldsymbol{e}_k\| \leq \left(\frac{\kappa - 1}{\kappa + 1}\right) \|\boldsymbol{e}_{k-1}\|$$

(recall that for positive definite $\boldsymbol{A}$ the eigenvalues coincide with the singular values and so $\kappa = \frac{M}{m}$).

11.2 Non-Stationary Richardson

The idea behind non-stationary Richardson is to see if we can find some sequence of values α_k such that,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k)$$

converges faster. To do this we can use what is known as polynomial acceleration.

As before, suppose $\boldsymbol{x}^\star$ is the solution to the problem such that $\boldsymbol{A}\boldsymbol{x}^\star = \boldsymbol{b}$ and let $\boldsymbol{e}_k = \boldsymbol{x}_k - \boldsymbol{x}^\star$. Then,

$$\begin{aligned}
\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + \alpha_{k-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k-1}) \\
\boldsymbol{x}_k - \boldsymbol{x}^\star &= \boldsymbol{x}_{k-1} - \boldsymbol{x}^\star + \alpha_{k-1}\boldsymbol{A}(\boldsymbol{x}^\star - \boldsymbol{x}_{k-1}) \\
\boldsymbol{e}_k &= (\boldsymbol{I} - \alpha_{k-1}\boldsymbol{A})(\boldsymbol{x}_{k-1} - \boldsymbol{x}^\star) \\
&= (\boldsymbol{I} - \alpha_{k-1}\boldsymbol{A})\boldsymbol{e}_{k-1} \\
&= \left(\prod_{i=0}^{k-1}(\boldsymbol{I} - \alpha_i\boldsymbol{A})\right)\boldsymbol{e}_0
\end{aligned}$$

(this is essentially exactly the same proof that was used in the last lecture for the convergence of stationary methods, except this time the iteration is not stationary and so we get a product of matrices that doesn't simplify to a simple matrix power as before).

So,

$$\boldsymbol{e}_k = p_k(\boldsymbol{A})\boldsymbol{e}_0 \qquad \text{where} \qquad p_k(\boldsymbol{A}) \coloneqq \prod_{i=0}^{k-1}(\boldsymbol{I} - \alpha_i\boldsymbol{A})$$

Given this, what is the best choice of $\{\alpha_0, \ldots, \alpha_{k-1}\}$? One way to answer this question is to say that the best choice of constants will be those that minimise the error vector $\boldsymbol{e}_k$. I.e.,

$$\begin{aligned}
\|\boldsymbol{e}_k\| &= \|p_k(\boldsymbol{A})\boldsymbol{e}_0\| \\
&\leq \|p_k(\boldsymbol{A})\|\|\boldsymbol{e}_0\|
\end{aligned}$$

And so a reasonable choice for the constants will be those that make $\|p_k(\boldsymbol{A})\|$ as small as possible. This is the idea behind polynomial acceleration.

11.3 Polynomial Acceleration

For a given choice $\{\alpha_0, \ldots, \alpha_{k-1}\}$ what is $\|p_k(\boldsymbol{A})\|$?

We previously assumed that $\boldsymbol{A} \succ \boldsymbol{0}$ which implies that $\boldsymbol{A}$ is Hermitian. If you then look at the definition of $p_k(\boldsymbol{A})$ it should be clear that it is also Hermitian (since each α_i is taken to be real). It was also mentioned previously in the course that if $\boldsymbol{X}$ is Hermitian then $\|\boldsymbol{X}\| = \rho(\boldsymbol{X})$. Therefore,

$$\|p_k(\boldsymbol{A})\| = \rho(p_k(\boldsymbol{A})) = \max_{j=1,\ldots,n} |\lambda_j(p_k(\boldsymbol{A}))|$$

Recall now from the spectral mapping theorem for matrix polynomials that $\lambda(p_k(\boldsymbol{A})) = p_k(\lambda(\boldsymbol{A}))$. Therefore,

$$\begin{aligned}\|p_k(\boldsymbol{A})\| &= \max_{j=1,\ldots,n} |p_k(\lambda_j(\boldsymbol{A}))| \\ &= \max_{j=1,\ldots,n} \left| \prod_{i=0}^{k-1} \Big(1 - \alpha_i \lambda_j(\boldsymbol{A})\Big) \right|\end{aligned}$$

Thus,

$$\min_{\{\alpha_0,\ldots,\alpha_{k-1}\}} \|p_k(\boldsymbol{A})\| = \min_{\{\alpha_0,\ldots,\alpha_{k-1}\}} \max_{j=1,\ldots,n} \left| \prod_{i=0}^{k-1} \Big(1 - \alpha_i \lambda_j(\boldsymbol{A})\Big) \right|$$

(and so we are dealing with a sort of min/max problem).

11.4 Chebyshev Polynomials

A lot of details are skipped here (slides 8 to 13), but the short story is that by loosening the min/max problem slightly we find that the solution for the optimal value of the coefficients is related to the roots of the Chebyshev polynomials.

This gives us an expression for the optimal choice for α_i,

$$\frac{1}{\alpha_i} = \frac{1}{2}\left(m + M + (M - m)\cos\left(\frac{\pi(2(i+1)-1)}{2k}\right)\right) \quad \text{for } i = 0, \ldots, k-1$$

(notice how this is similar to the optimal value in the stationary Richardson iteration where we had $\frac{1}{\alpha} = \frac{m+M}{2}$. In this optimal non-stationary value we just have an extra correction term).

We also get a bound on the error vector whereby,

$$\|\boldsymbol{e}_k\| \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|\boldsymbol{e}_0\|$$

(compare this with the stationary Richardson iteration: $\sqrt{\kappa}$ has replaced κ, which means that polynomial acceleration drastically improves the convergence speed).

11.5 Drawbacks of Naïve Chebyshev Acceleration

1. There can be large round-off errors
2. One has to choose the number of iterations k in advance
3. You need to have a good estimate of the interval that contains all the eigenvalues of $\boldsymbol{A}$. Furthermore you also need to know what m and M are.

The first two problems can be alleviated using a three-term recurrence relation in the implementation of the Chebyshev acceleration. The lecture notes go into a bit more detail about how this is implemented.

12 Krylov Subspaces

12.1 Background

Krylov subspace methods are perhaps the most important class of iterative methods for solving systems of linear equations.

- No explicit form of $\boldsymbol{A}$ is needed. They only require some "black-box" procedure that takes $\boldsymbol{v}$ and returns $\boldsymbol{Av}$ (or perhaps $\boldsymbol{A}^*\boldsymbol{v}$ as well)
- This makes them particularly well-suited for large sparse linear systems
- Many different types of Krylov subspace methods exist that are tailored to specific properties of the matrix $\boldsymbol{A}$
- Mathematically Krylov subspace methods will converge after a finite number of iterations (although in practice there can often be rounding errors).
- When iteration is stopped early the estimates are usually very good

12.2 Motivation

Recall that for the accelerated (non-stationary) Richardson iteration we have,

$$\begin{aligned}\boldsymbol{x}_k &= \boldsymbol{x}_{k-1} + \alpha_{k-1}(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k-1}) \\ &= \boldsymbol{x}_{k-1} + \alpha_{k-1}\boldsymbol{r}_{k-1}\end{aligned}$$

where $\boldsymbol{r}_{k-1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k-1}$ is the residual vector.

(which is different from the error vector. The error vector is the difference between $\boldsymbol{x}$ and $\boldsymbol{x}^\star$ whereas the residual is the difference between $\boldsymbol{Ax}$ and $\boldsymbol{Ax}^\star = \boldsymbol{b}$).

We can expand out the expression above to represent $\boldsymbol{x}_k$ in terms of $\boldsymbol{x}_0$ and the residuals such that,

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \alpha_i \boldsymbol{r}_i$$

So $\boldsymbol{x}_k$ can be written as $\boldsymbol{x}_0$ plus a linear function of the residual vectors $\boldsymbol{r}_i$. Recall that we previously found that the error vector $\boldsymbol{e}_i$ is given by,

$$\boldsymbol{e}_i = \prod_{j=0}^{i-1} (\boldsymbol{I} - \alpha_j \boldsymbol{A})\boldsymbol{e}_0$$

We can use this to express $\boldsymbol{r}_i$ since,

$$\boldsymbol{A}\boldsymbol{e}_i = \boldsymbol{A}(\boldsymbol{x}_i - \boldsymbol{x}^\star) = \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{A}\boldsymbol{x}^\star = \boldsymbol{A}\boldsymbol{x}_i - \boldsymbol{b} = -\boldsymbol{r}_i$$

Using this relationship we get that,

$$\begin{aligned}\boldsymbol{r}_i = -\boldsymbol{A}\boldsymbol{e}_i &= -\boldsymbol{A}\left(\prod_{j=0}^{i-1}(\boldsymbol{I} - \alpha_j \boldsymbol{A})\right)\boldsymbol{e}_0 \\ &= \left(\prod_{j=0}^{i-1}(\boldsymbol{I} - \alpha_j \boldsymbol{A})\right)(-\boldsymbol{A}\boldsymbol{e}_0) \\ &= \left(\prod_{j=0}^{i-1}(\boldsymbol{I} - \alpha_j \boldsymbol{A})\right)\boldsymbol{r}_0\end{aligned}$$

(this relies on the fact that $\boldsymbol{A}(\boldsymbol{I} - c\boldsymbol{A}) = (\boldsymbol{I} - c\boldsymbol{A})\boldsymbol{A}$, which holds for any square matrix $\boldsymbol{A}$).

From this we can express $\boldsymbol{x}_k$ as,

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \sum_{i=0}^{k-1} \alpha_i \prod_{j=0}^{i-1} (\boldsymbol{I} - \alpha_j \boldsymbol{A}) \boldsymbol{r}_0$$

Or in other words,

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \text{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$$

It turns out that, once k is large enough (any $k \geq t$ for the t defined below, and in particular $k = n$ always works),

$$\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \text{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$$

(proof requires the Cayley–Hamilton theorem).

12.3 Krylov Subspace Methods

Krylov subspace methods are built around the idea that for each value k we exploit the structure of,

$$\boldsymbol{x}_0 + \text{span}\{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$$

to choose the best approximation to $\boldsymbol{x}^\star$ within this subspace.

In general, the Krylov subspace of order k, generated by the matrix $\boldsymbol{A} \in \mathbb{C}^{n \times n}$ and the vector $\boldsymbol{v} \in \mathbb{C}^n$ is defined as,

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \text{span}\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \boldsymbol{A}^2\boldsymbol{v}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v}\}$$

12.4 Dimension of the Krylov Subspace

Notice that,

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) \subseteq \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v}) \qquad \Longrightarrow \qquad \dim\Big(\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})\Big) \leq \dim\Big(\mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{v})\Big)$$

Intuitively speaking though it should make sense that as k grows the dimension of the Krylov subspace cannot grow indefinitely. This intuition is correct and it is described by the notion of the grade of $\boldsymbol{v}$ with respect to $\boldsymbol{A}$.

Theorem: there is a positive integer $t(\boldsymbol{v}, \boldsymbol{A})$ called the grade of $\boldsymbol{v}$ with respect to $\boldsymbol{A}$ such that,

$$\dim(\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v})) = \begin{cases} k, & k \leq t \\ t, & k \geq t \end{cases}$$

Or in other words this theorem tells us two very interesting pieces of information:

1. for $k \leq t$ the dimension of the Krylov subspace is equal to the number of vectors spanning it. This implies that the vectors $\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{v}$ are linearly independent.
2. After the cut-off at $k = t$ the dimension stops increasing. I.e. all new vectors are linearly dependent on the previous ones.

It turns out that it is only when $k \geq t$ that we get that $\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$.

(the proof of this also follows from Cayley–Hamilton).

12.5 Standard Form of Krylov Subspace Methods

While specific Krylov subspace methods can vary greatly, the general idea is typically along these lines:

- Start from an initial guess $\boldsymbol{x}_0$ and set $k = 0$
- Iteratively form a new guess $\boldsymbol{x}_k$ where $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$
- This process will find the solution $\boldsymbol{x}^\star$ after t iterations

Ideally we want the method to give estimates such that $\boldsymbol{x}_k \approx \boldsymbol{x}^\star$ well before that, allowing the algorithm to be stopped early.

13 Krylov Basis

Given the importance of Krylov subspaces, we want a way to be able to construct vectors in the subspace. To do this we need a basis.

The most natural choice would be to take,

$$\mathcal{B} = \{\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0\}$$

We know that these vectors are linearly independent for $k \leq t$ and so they do form a valid basis.

However these are not a good choice because they become increasingly close to being linearly dependent as k grows. In fact for large k adjacent vectors in the sequence are nearly parallel. A better basis would be an orthonormal basis.

13.1 Finding an Orthonormal Basis

To do this we use Gram–Schmidt. Or specifically a slightly modified version of Gram–Schmidt referred to as the Arnoldi process.

See the algorithm on slide 10 of the lecture. The lecture notes also go into much more detail about this.

13.2 Theorem

The Arnoldi process breaks down at step $j \iff$ the grade of $\boldsymbol{r}_0$ with respect to $\boldsymbol{A}$ is j

(by breaks down here we mean that it reaches a division by zero)

14 Krylov Optimality

To refresh what we have established so far, we know that the exact solution to the linear system lies in an affine Krylov subspace,

$$\boldsymbol{x}^\star \in \boldsymbol{x}_0 + \mathcal{K}_t(\boldsymbol{A}, \boldsymbol{r}_0)$$

where t (the dimension of the Krylov subspace) is the grade of $\boldsymbol{r}_0$ with respect to $\boldsymbol{A}$.

Hence we can search for "best" approximations in a sequence of increasing subspaces,

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0), \quad k = 1, 2, \ldots$$

Previously we established that through techniques such as the Arnoldi process we can:

- Construct a basis for $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$

- Therefore we can generate candidates $\boldsymbol{x}_k$ in these spaces

But there are still some major questions that we have left unanswered:

- How do we pick the "best" $\boldsymbol{x}_k$?
- What does it even mean for a particular vector to be the "best"?
- How can we compute this efficiently?

The focus of this lecture is on answering the second point. I.e.,

what do we mean by the "best" iterate?

14.1 Some Optimality Criteria

There are of course different ways to define "best" and here are some.

1. Minimise the true error

$$\boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k}{\arg\min} \|\boldsymbol{x}-\boldsymbol{x}^\star\|_2 \quad \implies \quad \text{not computable}$$

The issue with this is that the optimality condition is not computable. This is because you don't know what $\boldsymbol{x}^\star$ is, so you have no way of knowing how close you are to $\boldsymbol{x}^\star$.

2. Energy norm minimisation (if $\boldsymbol{A} \succ \boldsymbol{0}$)

$$\boldsymbol{x}_k = \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k}{\arg\min} \|\boldsymbol{x}-\boldsymbol{x}^\star\|_{\boldsymbol{A}}$$

Now at first glance this looks like it is also non-computable and so it is no better than the first suggestion. However this is actually not the case.

$$\text{recall:} \quad \|\boldsymbol{v}\|_{\boldsymbol{A}}^2 = \langle \boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}\rangle = \boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}$$

Notice how you require $\boldsymbol{A}$ to be positive definite for this to be a norm (since you need the output to be non-negative and only equal to zero when $\boldsymbol{v}=\boldsymbol{0}$).

Given this definition we can expand out the expression above,

$$\begin{aligned}
\|\boldsymbol{x}-\boldsymbol{x}^\star\|_{\boldsymbol{A}}^2 &= \langle \boldsymbol{x}-\boldsymbol{x}^\star, \boldsymbol{A}(\boldsymbol{x}-\boldsymbol{x}^\star)\rangle \\
&= \langle \boldsymbol{x}-\boldsymbol{x}^\star, \boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}\rangle \\
&= \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle - \langle \boldsymbol{x}, \boldsymbol{b}\rangle - \langle \boldsymbol{x}^\star, \boldsymbol{A}\boldsymbol{x}\rangle + \langle \boldsymbol{x}^\star, \boldsymbol{b}\rangle \\
&= \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle - \langle \boldsymbol{x}, \boldsymbol{b}\rangle - \langle \boldsymbol{A}\boldsymbol{x}^\star, \boldsymbol{x}\rangle + \langle \boldsymbol{x}^\star, \boldsymbol{b}\rangle \quad (\text{since } \boldsymbol{A}\succ\boldsymbol{0} \implies \boldsymbol{A}=\boldsymbol{A}^T) \\
&= \langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle - 2\langle \boldsymbol{b}, \boldsymbol{x}\rangle + \langle \boldsymbol{x}^\star, \boldsymbol{b}\rangle
\end{aligned}$$

Given that the last term does not depend on $\boldsymbol{x}$ this means that,

$$\begin{aligned}
\boldsymbol{x}_k &= \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k}{\arg\min} \|\boldsymbol{x}-\boldsymbol{x}^\star\|_{\boldsymbol{A}} \\
&= \underset{\boldsymbol{x}\in\boldsymbol{x}_0+\mathcal{K}_k}{\arg\min} \frac{1}{2}\langle \boldsymbol{x}, \boldsymbol{A}\boldsymbol{x}\rangle - \langle \boldsymbol{b}, \boldsymbol{x}\rangle
\end{aligned}$$

(I am also using the fact here that minimising $\|\cdot\|_{\boldsymbol{A}}$ is the same as minimising $\frac{1}{2}\|\cdot\|_{\boldsymbol{A}}^2$.)

The point of the above is that using the energy norm makes this minimisation computable, since the expression that you need to minimise no longer depends on the unknown value of $\boldsymbol{x}^\star$.

3. Residual minimisation

The idea with residual minimisation is to take an alternative approach where,

$$\boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k}{\arg\min} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2$$

4. Galerkin / projection condition

For some k dimensional subspace $\mathcal{W}_k$,

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0) \quad \text{is selected such that,} \quad \boldsymbol{r}_k \perp \mathcal{W}_k$$

where recall that $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$.

Or in other words this is saying to pick the vector $\boldsymbol{x}_k$ to be such that its residual is orthogonal to the k-dimensional subspace $\mathcal{W}_k$.

Unifying Principle: It turns out that the second and third criteria above arise as special cases of the Galerkin projection condition for appropriate choices of $\mathcal{W}_k$.

14.2 Intuition Behind Projection Methods

We are searching for solutions within a k-dimensional affine space,

$$\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}_0)$$

Therefore there are k degrees of freedom, so in order to choose a unique iterate $\boldsymbol{x}_k$ we need k independent conditions.

Rather than optimising explicitly we impose orthogonality conditions on the residual. That is we require that,

$$\boldsymbol{r}_k \perp \mathcal{W}_k$$

for some k-dimensional constraint/test space $\mathcal{W}_k$. This provides exactly k independent conditions, thus leading to a unique solution.

And so we have the following general "ingredients" of a projection method:

- A search subspace: $\mathcal{K}_k$ with $\dim(\mathcal{K}_k) = k$
- A test/constraint subspace: $\mathcal{W}_k$ with $\dim(\mathcal{W}_k) = k$
- Petrov–Galerkin condition: find $\boldsymbol{x}_k \in \boldsymbol{x}_0 + \mathcal{K}_k$ such that $\boldsymbol{r}_k \perp \mathcal{W}_k$

If:

- $\mathcal{W}_k = \mathcal{K}_k$ then we have an orthogonal projection
- $\mathcal{W}_k \neq \mathcal{K}_k$ then we have an oblique projection (e.g. $\mathcal{W}_k = \boldsymbol{A}\mathcal{K}_k$)
- Different choices of $\mathcal{W}_k$ lead to different Krylov methods (and different notions of optimality)

It turns out (shown later in the course) that,

- $\mathcal{W}_k = \mathcal{K}_k \implies$ energy norm minimisation
- $\mathcal{W}_k = \boldsymbol{A}\mathcal{K}_k \implies$ residual minimisation

In general, different choices of $\mathcal{W}_k$ lead to different Krylov methods.

This is called the Petrov–Galerkin method / projection method. Essentially the idea behind the method boils down to:

- Iterate $\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{y}_k$
- where $\boldsymbol{y}_k$ is a vector such that $\boldsymbol{y}_k \in \mathcal{K}_k$ and $(\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{y}_k) \perp \mathcal{W}_k$

14.3 Finding $\boldsymbol{y}_k$

To enforce the restrictions for $\boldsymbol{y}_k$ described above we need a basis for $\mathcal{K}_k$ and $\mathcal{W}_k$.

Suppose,

$$\mathcal{K}_k = \text{span}\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k\} \qquad \mathcal{W}_k = \text{span}\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_k\}$$

and define the following matrices containing the basis vectors,

$$\boldsymbol{K}_k \coloneqq [\boldsymbol{v}_1 | \ldots | \boldsymbol{v}_k] \qquad \boldsymbol{W}_k \coloneqq [\boldsymbol{w}_1 | \ldots | \boldsymbol{w}_k]$$

Therefore the conditions on $\boldsymbol{y}_k$ can be expressed as follows,

$$\begin{aligned} \boldsymbol{y}_k \in \mathcal{K}_k \quad &\implies \quad \boldsymbol{y}_k = \boldsymbol{K}_k \boldsymbol{z}_k \quad \text{for some} \quad \boldsymbol{z}_k \in \mathbb{R}^k \\ (\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{y}_k) \perp \mathcal{W}_k \quad &\implies \quad \boldsymbol{W}_k^T(\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{y}_k) = \boldsymbol{0} \end{aligned}$$

(the second condition is because for two subspaces to be orthogonal we only need to check that all of the basis vectors are orthogonal).

We can solve this system for $\boldsymbol{y}_k$ as follows,

$$\begin{aligned} \boldsymbol{W}_k^T(\boldsymbol{r}_0 - \boldsymbol{A}\boldsymbol{y}_k) &= \boldsymbol{0} \\ \boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{y}_k &= \boldsymbol{W}_k^T \boldsymbol{r}_0 \\ \boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k \boldsymbol{z}_k &= \boldsymbol{W}_k^T \boldsymbol{r}_0 && \text{(substituting the first condition)} \\ \boldsymbol{z}_k &= (\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k)^{-1} \boldsymbol{W}_k^T \boldsymbol{r}_0 && \text{(assuming non-singular)} \end{aligned}$$

Thus,

$$\boldsymbol{y}_k = \boldsymbol{K}_k (\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k)^{-1} \boldsymbol{W}_k^T \boldsymbol{r}_0$$

And so a basic prototype for a projection method is to take,

$$\boldsymbol{x}_k = \boldsymbol{x}_0 + \boldsymbol{K}_k (\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k)^{-1} \boldsymbol{W}_k^T \boldsymbol{r}_0$$

Here is a pseudo-algorithm,

Algorithm 1 Prototype projection method

Input: initial guess $\boldsymbol{x}_0$

1: $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$
2: **for** $k = 1, 2, \ldots$ until convergence **do**
3: Form bases $\boldsymbol{K}_k$ for $\mathcal{K}_k$ and $\boldsymbol{W}_k$ for $\mathcal{W}_k$
4: Solve the $k \times k$ system $(\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k)\, \boldsymbol{z}_k = \boldsymbol{W}_k^T \boldsymbol{r}_0$ for $\boldsymbol{z}_k$
5: $\boldsymbol{x}_k \leftarrow \boldsymbol{x}_0 + \boldsymbol{K}_k \boldsymbol{z}_k$
6: **end for**

Notes:

- As always, notice how we don't actually compute the inverse explicitly.
- Also note that in practice Krylov subspace methods perform these steps in very clever ways, and so the actual procedure may look very different from the pseudo-code above.
- One other thing to notice is that this does rely on $\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k$ being invertible, and one natural question may be to ask under what conditions we can guarantee its non-singularity.

The matrix $\boldsymbol{W}_k^T \boldsymbol{A}\boldsymbol{K}_k$ is non-singular if either of these conditions hold:

- Orthogonal projection case: when $\boldsymbol{A} \succ \boldsymbol{0}$ and $\mathcal{W} = \mathcal{K}$
- Oblique projection case: $\boldsymbol{A}$ is non-singular and $\mathcal{W} = \boldsymbol{A}\mathcal{K}$

14.4 Specialised Algorithms

14.4.1 Conjugate Gradient (CG) Method

Defined only for $\boldsymbol{A} \succ \boldsymbol{0}$ where,

$$\boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k}{\arg\min} \|\boldsymbol{x} - \boldsymbol{x}^\star\|_{\boldsymbol{A}}$$

i.e. it is an energy norm minimisation.

Notes:

- Only works for $\boldsymbol{A} \succ \boldsymbol{0}$
- (CG is usually stated for symmetric (or Hermitian) positive definite matrices. With the definition of positive definiteness used in this course, $\boldsymbol{A} \succ \boldsymbol{0}$ already implies that $\boldsymbol{A}$ is Hermitian; see Section 6.)

14.4.2 Minimum Residual (MINRES)

Defined for Hermitian $\boldsymbol{A}$ where,

$$\boldsymbol{x}_k = \underset{\boldsymbol{x} \in \boldsymbol{x}_0 + \mathcal{K}_k}{\arg\min} \|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}\|_2$$

14.4.3 Generalised Minimum Residual (GMRES)

Same as MINRES, but generalised to matrices that are not necessarily Hermitian.

(After this the lecture goes into a fair bit of detail on the methods themselves, including how they work and their convergence properties. Most of this is covered in lecture note 15.)

15 CG and GMRES

The important points from this lecture are included in Section 14.4, under Krylov optimality. Lecture note 15 goes into a lot more detail than what is included there.

Part II

Optimisation

16 Introduction to Optimisation

16.1 What is Optimisation?

An optimisation problem typically has 3 components:

- Variables (these are the things that we are free to choose and need to optimise over)
- Constraints (rules that a valid choice of the variables must satisfy)
- Objective function (some function that we want to maximise/minimise)

In the general case the problem is to solve,

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$$

where here $\mathcal{X}$ is known as the feasible region. The feasible region can often be split up into inequality constraints and equality constraints such that,

$$\mathcal{X} = \{\boldsymbol{x} : \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0},\ \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}\}$$

(although technically $\mathcal{X}$ could be some arbitrary subset of the space in which $\boldsymbol{x}$ comes from).

Optimisation problems generally can be classified into a few groups:

- Continuous vs Discrete
- Unconstrained vs Constrained
- Global vs Local
- Stochastic vs Deterministic

17 Topology

Here we develop some background necessary for the future results.

17.1 Line Segment

The line segment joining $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^d$ is the set of points on the straight line joining them.

$$\{\boldsymbol{y} + \alpha(\boldsymbol{x} - \boldsymbol{y}) : 0 \leq \alpha \leq 1\}$$

17.2 Hyperplane

For $\boldsymbol{a} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, a hyperplane is,

$$\{\boldsymbol{x} \in \mathbb{R}^d : \langle \boldsymbol{a}, \boldsymbol{x} \rangle = b\}$$

Note that:

- This is not a subspace of $\mathbb{R}^d$ unless $b = 0$
- For $d = 2$ the hyperplane is a line
- For $d = 3$ the hyperplane is just called a plane
- For any $\boldsymbol{x}_0$ on the hyperplane, $\boldsymbol{a}^T(\boldsymbol{x} - \boldsymbol{x}_0) = 0$ for every $\boldsymbol{x}$ on the hyperplane (which implies that $\boldsymbol{a}$ is the normal vector)
- We also define a half-space as the set where $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \leq b$ or $\langle \boldsymbol{a}, \boldsymbol{x} \rangle \geq b$ (i.e. the region above or below a hyperplane).

17.3 Polyhedron

A polyhedron is any set that can be written as the solution of finitely many linear inequalities or equalities,

$$\mathcal{P} = \{\boldsymbol{x} : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \boldsymbol{C}\boldsymbol{x} = \boldsymbol{d}\}$$

Geometrically, it is the intersection of finitely many half-spaces (from the inequalities) and hyperplanes (from equalities).

A polyhedron that is bounded is called a polytope.

17.3.1 Examples

Non-negative orthant:

$$\mathbb{R}^d_+ = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x} \geq \mathbf{0}\}$$

Unit simplex:

$$\varepsilon_d = \{\boldsymbol{x} : \boldsymbol{x} \geq \mathbf{0},\ \langle \mathbf{1}, \boldsymbol{x} \rangle \leq 1\}$$

I.e. the unit simplex contains all points in the non-negative orthant that are below the hyperplane $\langle \mathbf{1}, \boldsymbol{x} \rangle = 1$. It is quite easy to confirm that this hyperplane crosses each axis at the corresponding standard unit vector (see Figure 1).

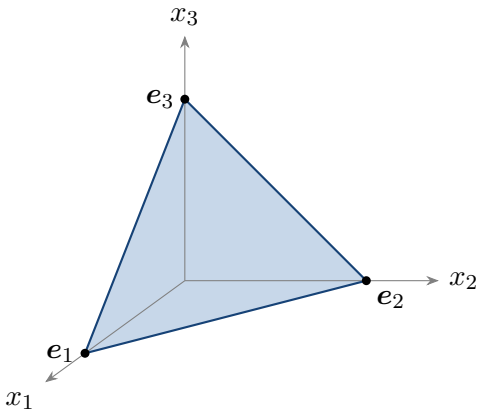


Figure 1: The unit simplex ε_3 is the solid tetrahedron with corners $\mathbf{0}$, $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$. Its shaded face, where $\langle \mathbf{1}, \boldsymbol{x} \rangle = 1$, is the probability simplex Δ_3.

Probability simplex:

$$\Delta_d = \{\boldsymbol{x} : \boldsymbol{x} \geq \mathbf{0},\ \langle \mathbf{1}, \boldsymbol{x} \rangle = 1\}$$

i.e. all the points on the face of the unit simplex where $\langle \mathbf{1}, \boldsymbol{x} \rangle = 1$. These are all the points whose entries sum to 1, and so each element in Δ_d defines a valid discrete probability distribution.

17.4 Convex Set

A set $\mathcal{C}$ is convex if for any two points $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$ the line segment joining $\boldsymbol{x}$ and $\boldsymbol{y}$ is entirely contained within the set $\mathcal{C}$. More formally for all $\alpha \in [0, 1]$,

$$\boldsymbol{y} + \alpha(\boldsymbol{x} - \boldsymbol{y}) \in \mathcal{C}$$

(or equivalently $\alpha\boldsymbol{x} + (1 - \alpha)\boldsymbol{y} \in \mathcal{C}$; if you expand it out they are both the same thing).

17.4.1 Operations Preserving Convexity

- Scaling: $\alpha\mathcal{C}$ is convex
- Minkowski sum: $\mathcal{C}_1 + \mathcal{C}_2$ is convex
  - $A + B = \{a + b : a \in A, b \in B\}$
- Affine mapping: $\boldsymbol{A}\mathcal{C} + \boldsymbol{b}$ is convex
- Cartesian product: $\mathcal{C}_1 \times \mathcal{C}_2$ is convex
- Intersection: $\mathcal{C}_1 \cap \mathcal{C}_2$ is convex

17.5 Local Neighbourhood

For any $\varepsilon > 0$, the ε-neighbourhood of a point $\boldsymbol{x} \in \mathbb{R}^d$ is,

$$\mathcal{B}(\boldsymbol{x}, \varepsilon) \coloneqq \{\boldsymbol{y} \in \mathbb{R}^d : \|\boldsymbol{y} - \boldsymbol{x}\| \leq \varepsilon\}$$

Different norms produce neighbourhoods of different shapes.

17.5.1 Interior Point

A point $\boldsymbol{x} \in \mathcal{C}$ is an interior point if,

$$\exists \varepsilon > 0 \quad \text{s.t.} \quad \mathcal{B}(\boldsymbol{x}, \varepsilon) \subseteq \mathcal{C}$$

17.5.2 Boundary Point

A point $\boldsymbol{x} \in \mathbb{R}^d$ is a boundary point of $\mathcal{C}$ if every neighbourhood of $\boldsymbol{x}$ intersects both $\mathcal{C}$ and its complement (so a point in $\mathcal{C}$ is a boundary point exactly when it is not an interior point).

17.5.3 Open Set

A set $\mathcal{C}$ is open if every point in $\mathcal{C}$ is an interior point.

17.5.4 Closed Set

A set $\mathcal{C}$ is closed if it contains all of its boundary points. Equivalently this implies that its complement must be open.

17.5.5 Bounded Set

A set $\mathcal{C}$ is bounded if there exists some $M > 0$ such that $\mathcal{C} \subseteq \mathcal{B}(\mathbf{0}, M)$.

17.5.6 Compact Set

A set in $\mathbb{R}^d$ is compact if and only if it is closed and bounded (this is the Heine–Borel Theorem).

17.6 Level Sets

For a function $f : \mathcal{C} \to \mathbb{R}$ the sub-level set at height α is,

$$\mathcal{S}(f, \alpha) = \{\boldsymbol{x} \in \mathcal{C} : f(\boldsymbol{x}) \leq \alpha\}$$

If f is continuous and $\mathcal{C}$ is closed, then every sub-level set $\mathcal{S}(f, \alpha)$ is also closed.

Similarly, the level set $\mathcal{L}(f, \alpha) = \{\boldsymbol{x} \in \mathcal{C} : f(\boldsymbol{x}) = \alpha\}$ is the set of all points at exactly the height α.

17.7 Weierstrass Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a continuous function and let $\mathcal{C} \subseteq \mathbb{R}^d$. Suppose any of the following hold,

- $\mathcal{C}$ is compact
- For some α, the sub-level set $\mathcal{S}(f, \alpha)$ is non-empty and compact
- $\mathcal{C}$ is closed and f is coercive. I.e. $\lim_{\|\boldsymbol{x}\| \to \infty} f(\boldsymbol{x}) = +\infty$

Then, there exists $\boldsymbol{x}^\star \in \mathcal{C}$ such that,

$$f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}) \qquad \forall \boldsymbol{x} \in \mathcal{C}$$

(in other words f attains its minimum on $\mathcal{C}$).

18 Calculus

18.1 Directional Derivative

Let $f : \mathcal{U} \to \mathbb{R}$ where $\mathcal{U} \subseteq \mathbb{R}^d$. Suppose $\boldsymbol{x} \in \text{int}(\mathcal{U})$ and $\boldsymbol{p} \in \mathbb{R}^d \setminus \{\boldsymbol{0}\}$ is some direction vector. Then if the limit,

$$\lim_{t \to 0} \frac{f(\boldsymbol{x} + t\boldsymbol{p}) - f(\boldsymbol{x})}{t}$$

exists, it is called the directional derivative of f at $\boldsymbol{x}$ in direction $\boldsymbol{p}$.

It is typically denoted as, $f'(\boldsymbol{x}; \boldsymbol{p})$.

18.2 Partial Derivative

The directional derivative along the coordinate direction $\boldsymbol{e}_i$ is called the partial derivative,

$$\frac{\partial f}{\partial x_i}(\boldsymbol{x}) \coloneqq \lim_{t \to 0} \frac{f(\boldsymbol{x} + t\boldsymbol{e}_i) - f(\boldsymbol{x})}{t}$$

This measures how f is changing in the ith variable keeping all other variables fixed.

18.3 Gradient Vector

If all partial derivatives exist at $\boldsymbol{x} \in \mathbb{R}^d$ then the gradient vector of f at $\boldsymbol{x}$ is given by,

$$\nabla f(\boldsymbol{x}) \coloneqq \left(\frac{\partial f}{\partial x_1}(\boldsymbol{x}), \ldots, \frac{\partial f}{\partial x_d}(\boldsymbol{x}) \right)^T$$

18.4 Differentiable Function (Fréchet Differentiable)

For a multivariate function to be differentiable it is not sufficient for just the gradient vector to be defined. That is because the existence of the gradient vector only tells you how the function behaves along the coordinate axes but not in arbitrary directions. So in general when one says that a multivariate function is differentiable they mean more than that the gradient vector exists (and this is encapsulated in the idea of Fréchet Differentiability).

Intuition: A function is differentiable at a point $\boldsymbol{x}$ if very close to $\boldsymbol{x}$ it behaves like a hyperplane. I.e.,

$$f(\boldsymbol{x} + \boldsymbol{p}) \approx f(\boldsymbol{x}) + \ell_{\boldsymbol{x}}(\boldsymbol{p})$$

Here the map $f(\boldsymbol{x}) + \ell_{\boldsymbol{x}}(\boldsymbol{p})$ is the best linear approximation of f near $\boldsymbol{x}$.

Formally, a function f is differentiable at $\boldsymbol{x}$ if there exists a linear map $\ell_{\boldsymbol{x}}(\boldsymbol{p})$ such that,

$$\lim_{\boldsymbol{p} \to \boldsymbol{0}} \frac{f(\boldsymbol{x} + \boldsymbol{p}) - f(\boldsymbol{x}) - \ell_{\boldsymbol{x}}(\boldsymbol{p})}{\|\boldsymbol{p}\|} = 0$$

The linear map is unique and has the form,

$$\ell_{\boldsymbol{x}}(\boldsymbol{p}) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{p} \rangle$$

If f is differentiable at $\boldsymbol{x}$ then all directional derivatives exist and,

$$f'(\boldsymbol{x}; \boldsymbol{q}) = \langle \nabla f(\boldsymbol{x}), \boldsymbol{q} \rangle$$

18.5 Proposition

Let f be differentiable. For any smooth curve γ on the level set,

$$\mathcal{L}(f, f(\boldsymbol{x}_0)) = \{\boldsymbol{x} : f(\boldsymbol{x}) = f(\boldsymbol{x}_0)\}$$

we have that the gradient $\nabla f(\boldsymbol{x}_0)$ is orthogonal to the tangent vector $\gamma'(t_0)$ at the point where $\boldsymbol{x}_0 = \gamma(t_0)$.

(essentially the gradient of a differentiable function is orthogonal to its level curves).

18.6 Gradient as Direction of Steepest Ascent

Let f be differentiable. The gradient points in the direction of maximum increase of f at $\boldsymbol{x}_0$. Formally,

$$\frac{\nabla f(\boldsymbol{x}_0)}{\|\nabla f(\boldsymbol{x}_0)\|} = \arg\max_{\|\boldsymbol{p}\|=1} \langle \nabla f(\boldsymbol{x}_0), \boldsymbol{p} \rangle$$

Proof. Let $\boldsymbol{p}$ be any unit vector. By the Cauchy–Schwarz inequality,

$$\langle \nabla f(\boldsymbol{x}_0), \boldsymbol{p} \rangle \leq \|\nabla f(\boldsymbol{x}_0)\|$$

Equality is achieved when,

$$\boldsymbol{p} = \frac{\nabla f(\boldsymbol{x}_0)}{\|\nabla f(\boldsymbol{x}_0)\|}$$

□

18.7 Continuously Differentiable

Let $\mathcal{U}$ be an open set. Then $f \in \mathcal{C}^1(\mathcal{U})$ if all partial derivatives exist and are continuous on $\mathcal{U}$.

Important conditions to remember:

- **Necessary**: If a function is discontinuous at a point then it cannot be differentiable there
- **Necessary**: If partial derivatives do not exist at a point then the function cannot be differentiable there
- **Sufficient**: If all partial derivatives exist and are continuous in a neighbourhood then the function is continuously differentiable (i.e. in $\mathcal{C}^1$)
- **Inconclusive**: If partial derivatives exist but are not continuous

18.8 Hessian

If all second-order partial derivatives of f exist then the Hessian is,

$$\nabla^2 f(\boldsymbol{x}) \coloneqq \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_1 \partial x_2}(\boldsymbol{x}) & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_d}(\boldsymbol{x}) \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_2^2}(\boldsymbol{x}) & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_d}(\boldsymbol{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1}(\boldsymbol{x}) & \frac{\partial^2 f}{\partial x_d \partial x_2}(\boldsymbol{x}) & \dots & \frac{\partial^2 f}{\partial x_d^2}(\boldsymbol{x}) \end{pmatrix}$$

Intuition:

- The Hessian captures the curvature of f

18.9 Twice Continuously Differentiable

Let $\mathcal{U}$ be an open set. Then $f \in \mathcal{C}^2(\mathcal{U})$ if all partial derivatives are continuously differentiable.

18.10 Useful Matrix Calculus Formulae

$$f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle \implies \begin{cases} \nabla f(\boldsymbol{x}) = \boldsymbol{a} \\ \nabla^2 f(\boldsymbol{x}) = \boldsymbol{0} \end{cases}$$

$$f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{B}\boldsymbol{x} \rangle \implies \begin{cases} \nabla f(\boldsymbol{x}) = (\boldsymbol{B} + \boldsymbol{B}^T)\boldsymbol{x} \\ \nabla^2 f(\boldsymbol{x}) = \boldsymbol{B} + \boldsymbol{B}^T \end{cases}$$

18.11 Taylor's Theorem

If f is twice differentiable,

$$f(\boldsymbol{x} + \boldsymbol{p}) \approx f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{p} \rangle + \frac{1}{2} \left\langle \boldsymbol{p}, \nabla^2 f(\boldsymbol{x}) \boldsymbol{p} \right\rangle$$

The lecture also includes several other formulae (see slides 18 and 19).

18.12 Lipschitz Continuity

A mapping $\boldsymbol{h} : \mathcal{U} \to \mathcal{V}$ is called L-Lipschitz continuous if there exists a constant $L \in [0, \infty)$ such that,

$$\|\boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{h}(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{U}$$

You can think of L here as the maximum slope of the function.

Notes:

- A function with an L_g-Lipschitz continuous gradient is denoted as $\mathcal{C}^{1,1}_{L_g}$
- A function with an L_H-Lipschitz continuous Hessian is denoted as $\mathcal{C}^{2,2}_{L_H}$

(note that when applying the definition of Lipschitz continuity to a vector here we use the Euclidean norm. When applying it to a matrix we use the spectral norm).

(There are a few more results at the end of the lecture slides, which aren't included here.)

19 Optimality

19.1 Generic Optimisation Problem

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

where:

- f is the objective function
- $\mathcal{C}$ is the feasible set

A point $\boldsymbol{x}^\star \in \mathcal{C}$ is a global minimiser if,

$$f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \mathcal{C}$$

A strict global minimiser is one where the inequality is strict for all $\boldsymbol{x} \in \mathcal{C} \setminus \{\boldsymbol{x}^\star\}$.

A point $\boldsymbol{x}^\star \in \mathcal{C}$ is a local minimiser if,

$$f(\boldsymbol{x}^\star) \leq f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \text{ near } \boldsymbol{x}^\star$$

where "near" here means within some small ε neighbourhood.

19.2 Unconstrained Problem

The generic unconstrained problem is,

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$$

19.2.1 First-Order Necessary Condition

If $\boldsymbol{x}^\star$ is a local minimiser of f and f is differentiable near $\boldsymbol{x}^\star$ then,

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$$

19.2.2 Stationary Points

A point $\boldsymbol{x}^\star \in \mathbb{R}^d$ is called a stationary point if,

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$$

Notes:

- Every local minimiser is a stationary point
- Not every stationary point is a local minimiser
- To distinguish between stationary points we need second-order information (i.e. the Hessian).

19.2.3 Second-Order Necessary Condition

If $\boldsymbol{x}^\star$ is an unconstrained local minimiser and f is twice differentiable near $\boldsymbol{x}^\star$ then,

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0} \quad \text{and} \quad \nabla^2 f(\boldsymbol{x}^\star) \succeq \mathbf{0}$$

19.2.4 Second-Order Sufficient Condition

If f is twice differentiable near $\boldsymbol{x}^\star$ and,

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0} \quad \text{and} \quad \nabla^2 f(\boldsymbol{x}^\star) \succ \mathbf{0}$$

Then $\boldsymbol{x}^\star$ is a strict local minimiser of f.

19.2.5 Definition: Saddle Point

A stationary point $\boldsymbol{x}^\star$ is a saddle point if there exist directions $\boldsymbol{p}$ and $\boldsymbol{q}$ such that for small $\alpha > 0$,

$$f(\boldsymbol{x}^\star) < f(\boldsymbol{x}^\star + \alpha\boldsymbol{p})$$
$$f(\boldsymbol{x}^\star) > f(\boldsymbol{x}^\star + \alpha\boldsymbol{q})$$

(i.e. it is a stationary point that is neither a local minimum nor a local maximum).

19.2.6 Sufficient Condition for a Saddle Point

If f is twice-differentiable near $\boldsymbol{x}^\star$,

$$\nabla f(\boldsymbol{x}^\star) = \mathbf{0} \quad \text{and} \quad \nabla^2 f(\boldsymbol{x}^\star) \text{ is indefinite}$$

then $\boldsymbol{x}^\star$ is a saddle point.

19.2.7 Sufficient Condition for Global Optimality

If f is twice differentiable and,

$$\nabla^2 f(\boldsymbol{x}) \succeq \mathbf{0} \quad \forall \boldsymbol{x} \in \mathbb{R}^d$$

then every stationary point of f is a global minimiser.

(this condition on the Hessian means that f is convex; convexity is covered in the next lecture, Section 20).

19.2.8 Summary

Condition	Requirement	Conclusion
First-order necessary	$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$	holds at every local minimiser
Second-order necessary	$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\boldsymbol{x}^\star) \succeq \mathbf{0}$	holds at every local minimiser
Second-order sufficient	$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\boldsymbol{x}^\star) \succ \mathbf{0}$	$\boldsymbol{x}^\star$ is a strict local minimiser
Saddle point	$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\boldsymbol{x}^\star)$ is indefinite	$\boldsymbol{x}^\star$ is a saddle point
Global optimality	$\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ and $\nabla^2 f(\boldsymbol{x}) \succeq \mathbf{0}$ for all $\boldsymbol{x}$	$\boldsymbol{x}^\star$ is a global minimiser

Table 1: Optimality conditions for the unconstrained problem $\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x})$.

19.3 Constrained Optimisation

The generic constrained problem is,

$$\min_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$$

If:

- $\boldsymbol{x}^\star \in \text{int}(\mathcal{C})$ then the optimality conditions are the same as those of unconstrained optimisation
- $\boldsymbol{x}^\star$ is on the boundary then the constraints restrict the feasible directions.

19.3.1 Feasible Directions

A vector $\boldsymbol{p} \in \mathbb{R}^d$ is a feasible direction at $\boldsymbol{x}^\star \in \mathcal{C}$ if $\exists \delta > 0$ such that,

$$\boldsymbol{x}^\star + \alpha \boldsymbol{p} \in \mathcal{C} \quad \forall \alpha \in [0, \delta]$$

(i.e. it is any direction that you can move in without leaving the feasible set).

We denote the set of all feasible directions from a given point $\boldsymbol{x}^\star \in \mathcal{C}$ with,

$$\mathcal{F}(\boldsymbol{x}^\star)$$

Note that if $\mathcal{C}$ is convex then $\mathcal{F}(\boldsymbol{x}^\star) = \{\alpha(\boldsymbol{x} - \boldsymbol{x}^\star) : \boldsymbol{x} \in \mathcal{C},\ \alpha > 0\}$. The reason for this is because the direction connecting any two points within a convex set is a feasible direction (and any positive multiple of a feasible direction is also feasible).

19.3.2 First-Order Necessary Condition

If $\boldsymbol{x}^\star \in \mathcal{C}$ is a local minimiser of f over $\mathcal{C}$ then,

$$\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle \geq 0 \quad \forall \boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star)$$

(i.e. the directional derivative in any feasible direction is non-negative and so it can only ever increase the value of the objective function).

Notice that the necessary condition $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$ for the unconstrained problem is just a special case of this. This is because in the unconstrained problem $\mathcal{F}(\boldsymbol{x}^\star) = \mathbb{R}^d$ and the only way that $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle \geq 0$ can hold for all $\boldsymbol{p} \in \mathbb{R}^d$ is if $\nabla f(\boldsymbol{x}^\star) = \mathbf{0}$.

A point that satisfies $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle \geq 0 \ \forall \boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star)$ is called a stationary point.

19.3.3 Second-Order Necessary Condition

If $\boldsymbol{x}^\star \in \mathcal{C}$ is a local minimiser of f over $\mathcal{C}$ then,

$$\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle \geq 0 \quad \forall \boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star)$$

and

$$\boldsymbol{p}^T \nabla^2 f(\boldsymbol{x}^\star) \boldsymbol{p} \geq 0 \quad \forall \boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star) \quad \text{with} \quad \langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle = 0$$

It should be clear how these are really just extensions of the second-order necessary conditions for unconstrained problems but now we are restricted to feasible directions in $\mathcal{F}(\boldsymbol{x}^\star)$.

19.3.4 Second-Order Sufficient Condition

If $\boldsymbol{x}^\star \in \mathcal{C}$ is stationary and,

$$\left\langle \boldsymbol{p}, \nabla^2 f(\boldsymbol{x}^\star) \boldsymbol{p} \right\rangle > 0 \quad \text{for all } \boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star) \setminus \{\mathbf{0}\} \quad \text{with} \quad \langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle = 0$$

then $\boldsymbol{x}^\star$ is a strict local minimiser.

19.3.5 Second-Order Sufficient Condition (Convex Feasible Set)

Suppose $\mathcal{C}$ is convex and $\boldsymbol{x}^\star \in \mathcal{C}$ is stationary. If for some $\gamma > 0$ we have,

$$\left\langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla^2 f(\boldsymbol{x}^\star)(\boldsymbol{x} - \boldsymbol{x}^\star) \right\rangle \geq \gamma \|\boldsymbol{x} - \boldsymbol{x}^\star\|^2, \quad \forall \boldsymbol{x} \in \mathcal{C}$$

then $\boldsymbol{x}^\star$ is a local minimiser.

19.3.6 Summary

Condition	Requirement	Conclusion
First-order necessary	$\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle \geq 0$ for all $\boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star)$	holds at every local minimiser
Second-order necessary	first-order condition, and $\boldsymbol{p}^T \nabla^2 f(\boldsymbol{x}^\star) \boldsymbol{p} \geq 0$ for all $\boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star)$ with $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle = 0$	holds at every local minimiser
Second-order sufficient	$\boldsymbol{x}^\star$ stationary, and $\boldsymbol{p}^T \nabla^2 f(\boldsymbol{x}^\star) \boldsymbol{p} > 0$ for all $\boldsymbol{p} \in \mathcal{F}(\boldsymbol{x}^\star) \setminus \{\mathbf{0}\}$ with $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{p} \rangle = 0$	$\boldsymbol{x}^\star$ is a strict local minimiser
Second-order sufficient ($\mathcal{C}$ convex)	$\boldsymbol{x}^\star$ stationary, and for some $\gamma > 0$, $\langle \boldsymbol{x} - \boldsymbol{x}^\star, \nabla^2 f(\boldsymbol{x}^\star)(\boldsymbol{x} - \boldsymbol{x}^\star) \rangle \geq \gamma \|\boldsymbol{x} - \boldsymbol{x}^\star\|^2$ for all $\boldsymbol{x} \in \mathcal{C}$	$\boldsymbol{x}^\star$ is a local minimiser

Table 2: Optimality conditions for the constrained problem $\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$.

(note that I left off all of the proofs from this section. See the lecture slides for the details).

20 Convexity

A function $f : \mathcal{C} \to \mathbb{R}$ defined on a convex set $\mathcal{C} \subseteq \mathbb{R}^d$ is convex if,

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}) \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C} \text{ and } \alpha \in [0,1]$$

Notes:

- In other words the straight line connecting any two points on the graph of f must lie on or above the graph.
- The requirement that f is defined on a convex set $\mathcal{C}$ is very important
- Geometrically such a function is "bowl-shaped"
- A function f is said to be concave if $-f$ is convex

Examples:

- Affine functions $f(\boldsymbol{x}) = \langle \boldsymbol{a}, \boldsymbol{x} \rangle + b$ are both convex and concave (and conversely, any function that is both convex and concave is affine)
- Any norm is convex
- Quadratic functions $f(\boldsymbol{x}) = c + \boldsymbol{b}^T \boldsymbol{x} + \frac{1}{2} \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ where $\boldsymbol{A} \succeq \boldsymbol{0}$ are convex

20.1 First-Order Condition for Convexity

If $f : \mathcal{C} \to \mathbb{R}$ is differentiable on an open set containing the convex set $\mathcal{C}$ then f is convex if and only if,

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$$

(i.e. at any point $\boldsymbol{x}$ the tangent plane at $\boldsymbol{x}$ is always an underestimate of f).

20.2 Second-Order Sufficient Condition for Convexity

Suppose $f : \mathcal{C} \to \mathbb{R}$ is twice differentiable on an open set containing the convex set $\mathcal{C}$. Then,

$$\text{if } \nabla^2 f(\boldsymbol{x}) \succeq \boldsymbol{0}, \quad \forall \boldsymbol{x} \in \mathcal{C} \quad \text{then } f \text{ is convex over } \mathcal{C}$$

(so a p.s.d. Hessian implies the function is convex, although this is not a necessary condition).

20.3 Optimality Problems

Consider the optimisation problem,

$$\min_{\boldsymbol{x} \in \mathcal{C}} f(\boldsymbol{x})$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is convex and differentiable on an open set containing the convex set $\mathcal{C}$.

1. **Necessary Condition**: Any locally optimal point is stationary
2. **Sufficient Condition**: Any stationary point is a global optimum

20.4 Strictly Convex Function

A function $f : \mathcal{C} \to \mathbb{R}$ over a convex set $\mathcal{C} \subseteq \mathbb{R}^d$ is strictly convex if for all $\boldsymbol{x} \neq \boldsymbol{y} \in \mathcal{C}$ and $\alpha \in (0,1)$,

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) < \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y})$$

(i.e. it is just the definition of convexity but with a strict inequality).

We also get first and second order sufficient conditions for strict convexity that mirror the ones for normal convexity but where the non-strict inequality becomes strict and the p.s.d. becomes p.d.

20.4.1 Uniqueness of Minimiser

Consider,

$$\min_{\boldsymbol{x}\in\mathcal{C}} f(\boldsymbol{x})$$

where f is strictly convex and $\mathcal{C}$ is convex. Then the minimiser (if one exists) is unique.

Proof. This follows quite directly from the definition of a strictly convex function. Suppose $\boldsymbol{x}_1 \neq \boldsymbol{x}_2$ were both minimisers, so that $f(\boldsymbol{x}_1) = f(\boldsymbol{x}_2) = f^\star$ is the minimum value. Since $\mathcal{C}$ is convex, $\frac{1}{2}\boldsymbol{x}_1 + \frac{1}{2}\boldsymbol{x}_2 \in \mathcal{C}$, and by strict convexity,

$$f\left(\tfrac{1}{2}\boldsymbol{x}_1 + \tfrac{1}{2}\boldsymbol{x}_2\right) < \tfrac{1}{2}f(\boldsymbol{x}_1) + \tfrac{1}{2}f(\boldsymbol{x}_2) = f^\star$$

which contradicts $f^\star$ being the minimum value. □

20.5 Strongly Convex Function

An even stronger requirement is this concept of a "strongly convex" function.

A function $f : \mathcal{C} \to \mathbb{R}$ is strongly convex with parameter $\mu > 0$ if,

$$f(\boldsymbol{x}) - \frac{\mu}{2}\|\boldsymbol{x}\|^2 \quad \text{is convex over } \mathcal{C}$$

In other words even when we subtract off a term that is quadratic in $\boldsymbol{x}$ the function is still convex.

In these cases we say that f is μ-strongly convex.

20.5.1 First-Order Condition for Strong Convexity

If f is differentiable on an open set containing $\mathcal{C}$, then f is μ-strongly convex if and only if,

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x}\rangle + \frac{\mu}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{C}$$

I.e. at every point the function lies above a quadratic function with curvature μ.

20.5.2 Second-Order Condition for Strong Convexity

Suppose $f : \mathcal{C} \to \mathbb{R}$ is twice differentiable on an open set containing the convex set $\mathcal{C} \subseteq \mathbb{R}^d$. Then,

$$\text{if } \nabla^2 f(\boldsymbol{x}) \succeq \mu \boldsymbol{I}, \quad \forall \boldsymbol{x} \in \mathcal{C} \qquad \text{then } f \text{ is } \mu\text{-strongly convex over } \mathcal{C}$$

Notice how this is just an extension of the second-order condition for convexity now requiring that the Hessian is still p.s.d. after μ is subtracted from each of its diagonal entries.

20.6 Hierarchy

Each type of convexity implies the next, and the same goes for the second-order conditions on the Hessian (for all $\boldsymbol{x} \in \mathcal{C}$):

$$\begin{array}{ccccc} \mu\text{-strongly convex} & \Longrightarrow & \text{strictly convex} & \Longrightarrow & \text{convex} \\ \Uparrow & & \Uparrow & & \Uparrow \\ \nabla^2 f(\boldsymbol{x}) \succeq \mu \boldsymbol{I} & \Longrightarrow & \nabla^2 f(\boldsymbol{x}) \succ \boldsymbol{0} & \Longrightarrow & \nabla^2 f(\boldsymbol{x}) \succeq \boldsymbol{0} \end{array}$$

None of these implications reverse in general. For example $f(x) = x^4$ is strictly convex, but it is not strongly convex and $f''(0) = 0$.

21 Descent Methods

We now focus on the unconstrained optimisation problem on smooth functions.

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) \quad \text{where } f \text{ is smooth enough}$$

21.1 Descent Direction

A direction $\boldsymbol{p} \in \mathbb{R}^d$ is a descent direction of f at $\boldsymbol{x}$ if for some $\alpha_0 > 0$ we have,

$$f(\boldsymbol{x} + \alpha\boldsymbol{p}) < f(\boldsymbol{x}) \quad \forall \alpha \in (0, \alpha_0]$$

I.e. moving in that direction (even if only a very small distance) will decrease the value of the function. Geometrically you can think of it as all directions that go "downhill" from where you are standing.

Sufficient condition for descent direction: If $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable and $\boldsymbol{p} \in \mathbb{R}^d$ such that,

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{p} \rangle < 0$$

then $\boldsymbol{p}$ is a descent direction for f at $\boldsymbol{x}$.

This should make intuitive sense since the directional derivative should be negative for a descent direction. This is not a necessary condition though: for example, at a strict local maximum $\nabla f(\boldsymbol{x}) = \boldsymbol{0}$, but every direction is a descent direction.

21.2 Basic Iterative Descent Method

Starting from an initial guess $\boldsymbol{x}_0 \in \mathbb{R}^d$ we do,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad k = 0, 1, \ldots$$

where,

- $\boldsymbol{p}_k \in \mathbb{R}^d$ is a descent direction at $\boldsymbol{x}_k$
- $\alpha_k > 0$ is an appropriate step-size at the kth iteration (known as the learning rate in ML contexts).

This leaves two fundamental questions that need to be answered when designing an iterative descent method:

1. How to choose the descent direction $\boldsymbol{p}_k$
2. How to choose the step size α_k

21.3 How to Choose the Descent Direction?

Recall that $\nabla f(\boldsymbol{x})$ points in the direction of steepest ascent. Hence the simplest descent direction is to take $\boldsymbol{p} = -\nabla f(\boldsymbol{x})$. More generally let $\boldsymbol{S}(\boldsymbol{x}) \succ \boldsymbol{0}$ and take the descent direction,

$$\boldsymbol{p} = -\boldsymbol{S}(\boldsymbol{x})\nabla f(\boldsymbol{x})$$

You can confirm that this is a valid descent direction since,

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{p} \rangle = -\langle \nabla f(\boldsymbol{x}), \boldsymbol{S}(\boldsymbol{x})\nabla f(\boldsymbol{x}) \rangle < 0 \quad \text{(by p.d., assuming } \nabla f(\boldsymbol{x}) \neq \boldsymbol{0})$$

Examples:

- Taking $\boldsymbol{S}(\boldsymbol{x}) = \boldsymbol{I}$ gives you "Gradient Descent"
- If f is strongly convex then $\boldsymbol{S}(\boldsymbol{x}) = [\nabla^2 f(\boldsymbol{x})]^{-1}$ gives you Newton's Method
- and many other methods

The matrix $\boldsymbol{S}(\boldsymbol{x})$ can effectively rescale the geometry of the space producing a descent direction better aligned with the local curvature.

21.4 How to Choose the Step-Size?

Choosing the appropriate step-size is critical. Too low and it takes many updates to converge, too large and you may have divergent behaviour.

Perhaps the ideal choice is to do what is known as an exact line search where we take,

$$\alpha_k \in \arg\min_{\alpha \geq 0} f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)$$

In general computing the exact line search is often too expensive.

Alternatively we can take the "naïve" approach and that is to pick $\alpha > 0$ such that,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) < f(\boldsymbol{x}_k)$$

However this criterion is too vague since very small values of α would satisfy this but may be too small to be useful.

More practical strategies are based on inexact line search which aims to:

- Find a step length that gives a sufficient reduction in f
- Avoid the high cost of an exact line search

21.4.1 Armijo Condition

Choose $\alpha > 0$ such that,

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + \alpha\beta \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle \quad \text{for some } \beta \in (0, 1)$$

Or in other words:

- Choose a step size α such that moving from $\boldsymbol{x}_k$ in the descent direction by a step of size α takes you to a point that lies below the line through $f(\boldsymbol{x}_k)$ whose slope is a fraction β of the slope of the tangent line at $\boldsymbol{x}_k$ (see Figure 2).

Notes:

- Typically we set $\beta = 10^{-4}$
- Armijo condition alone still allows tiny steps. To avoid this we often look for the largest α that still satisfies the condition (with some caveats).
- You can use "backtracking" techniques that start with large values of α and reduce geometrically until the condition is satisfied

There is a lemma that tells us that (as long as $\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle < 0$) there will always be a step size that satisfies the Armijo condition.

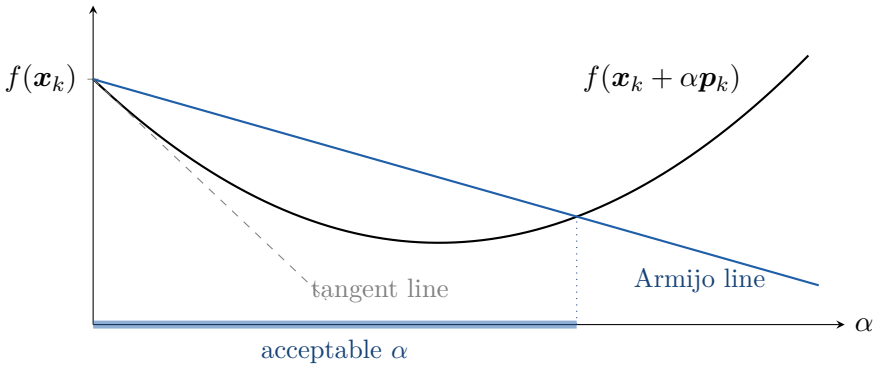


Figure 2: The Armijo condition. The tangent line is $f(\boldsymbol{x}_k) + \alpha \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle$ and the Armijo line is $f(\boldsymbol{x}_k) + \alpha\beta \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle$. Step sizes are acceptable where $f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)$ lies below the Armijo line. (Here β is much larger than the usual 10^{-4}, so that the two lines can be told apart.)

21.4.2 Wolfe Conditions

Find an α such that,

1. $f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k) \leq f(\boldsymbol{x}_k) + \alpha\beta_1 \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle$ (Armijo Condition)
2. $\langle \nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k), \boldsymbol{p}_k \rangle \geq \beta_2 \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle$ (Curvature Condition)

where $0 < \beta_1 < \beta_2 < 1$.

Essentially, on top of the Armijo condition, this is saying that the directional derivative should be less negative (i.e. greater) at $\boldsymbol{x}_k + \alpha \boldsymbol{p}_k$ than at $\boldsymbol{x}_k$. In other words we enforce that the gradient needs to flatten. This avoids choosing very small step sizes.

There is also the strong Wolfe condition, which takes the magnitude of both sides of the curvature condition: $|\langle \nabla f(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k), \boldsymbol{p}_k \rangle| \leq \beta_2 |\langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p}_k \rangle|$. As with Armijo, there is also an existence lemma for step sizes satisfying the Wolfe conditions.

22 Gradient Descent

In a general descent method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

In Gradient Descent we take,

$$\boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$$

This leads to the following algorithm,

Algorithm 2 Gradient descent

Input: initial guess $\boldsymbol{x}_0$, tolerance $\varepsilon > 0$

for $k = 0, 1, 2, \ldots$ **do**
  if $\|\nabla f(\boldsymbol{x}_k)\| \leq \varepsilon$ **then**
    return $\boldsymbol{x}_k$
  end if
  $\boldsymbol{p}_k \leftarrow -\nabla f(\boldsymbol{x}_k)$
  Choose a step size $\alpha_k > 0$ (e.g. by a line search, Section 21)
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
end for

The rest of this lecture goes into rates of convergence and other details, which aren't included here.

23 Newton's Method

In a general descent method,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$$

We then have the scaled gradient descent methods where we take,

$$\boldsymbol{p}_k = -\boldsymbol{S}(\boldsymbol{x}_k)\nabla f(\boldsymbol{x}_k), \quad \boldsymbol{S} \succ \boldsymbol{0}$$

Newton's method corresponds to taking,

$$\boldsymbol{S}(\boldsymbol{x}_k) = \left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1}$$

I.e. the method corresponds to taking,

$$\boldsymbol{p}_k = -\left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k)$$

For strongly convex problems, Newton's direction is always a descent direction since $\nabla^2 f(\boldsymbol{x}_k) \succ \boldsymbol{0}$ for all $\boldsymbol{x}_k \in \mathbb{R}^d$ which implies that $[\nabla^2 f(\boldsymbol{x}_k)]^{-1} \succ \boldsymbol{0}$.

The algorithm is as follows,

Algorithm 3 Newton's method

Input: initial guess $\boldsymbol{x}_0$, tolerance $\varepsilon > 0$

for $k = 0, 1, 2, \ldots$ **do**
  if $\|\nabla f(\boldsymbol{x}_k)\| \leq \varepsilon$ **then**
    return $\boldsymbol{x}_k$
  end if
  Solve $\nabla^2 f(\boldsymbol{x}_k)\, \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k)$ for $\boldsymbol{p}_k$
  Choose a step size $\alpha_k > 0$ ($\alpha_k = 1$ gives the pure Newton step)
  $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k$
end for

23.1 Another Interpretation

Newton's descent direction can be interpreted as minimising the second-order Taylor approximation of f around $\boldsymbol{x}_k$.

$$f(\boldsymbol{x}_k + \boldsymbol{p}) \approx f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p} \rangle + \frac{1}{2}\left\langle \boldsymbol{p}, \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p} \right\rangle$$

So we take $\boldsymbol{p}_k$ such that,

$$\begin{aligned} \boldsymbol{p}_k &= \operatorname*{arg\,min}_{\boldsymbol{p} \in \mathbb{R}^d} \left[f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{p} \rangle + \frac{1}{2}\left\langle \boldsymbol{p}, \nabla^2 f(\boldsymbol{x}_k)\boldsymbol{p} \right\rangle \right] \\ &= -\left[\nabla^2 f(\boldsymbol{x}_k)\right]^{-1} \nabla f(\boldsymbol{x}_k) \end{aligned}$$

(The rest of the lecture note, on convergence rates and so on, isn't covered here.)

24 Quasi-Newton Methods

24.1 The Idea

Gradient descent was good but it had several disadvantages:

- zig-zag behaviour
- sensitivity to change in coordinates
- slow convergence

On the other hand Newton's method was much better but it required computing the Hessian which can be expensive (both to compute and to store). This made applying Newton's method to large problems infeasible.

This motivated algorithms that attempt to "approximate" the curvature (i.e. the Hessian). These algorithms:

- retain much of Newton's fast convergence
- are computationally much cheaper

24.2 Secant Based Approximations

In the univariate case, Newton's update for an optimisation problem is,

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

If we don't have f'' (the second derivative) we can approximate it using the secant method where,

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

(assuming that x_k and x_{k-1} are sufficiently close this will be a good approximation).

This gives us the following update rule,

$$x_{k+1} = x_k - \left(\frac{x_k - x_{k-1}}{f'(x_k) - f'(x_{k-1})}\right) f'(x_k)$$

In Quasi-Newton Methods (QNMs) we just extend this to the multivariate case. Notice that the secant approximation is saying that $f''(x_k)$ should approximately satisfy,

$$f''(x_k)(x_k - x_{k-1}) \approx f'(x_k) - f'(x_{k-1})$$

The analogous multivariate case is,

$$\nabla^2 f(\boldsymbol{x}_k)(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \approx \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})$$

And so we can approximate $\nabla^2 f(\boldsymbol{x}_k)$ with some matrix $\boldsymbol{B}_k$ such that,

$$\underbrace{\boldsymbol{B}_k(\boldsymbol{x}_k - \boldsymbol{x}_{k-1}) \approx \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})}_{\text{the secant condition}}$$

Looking at this condition:

- involves d equations
- but we have d^2 unknowns (i.e. many more degrees of freedom than restrictions)

We can impose some restrictions on the solution $\boldsymbol{B}_k$ such as,

- symmetry (since the Hessian is symmetric for smooth functions)
- positive definiteness (since this is required for the resulting direction to be a descent direction)

But these two restrictions are still not enough to mandate a unique solution.