APPENDIX A

Background Mathematics

A.1 Linear Algebra

A.1.1 Vector algebra

Let $\mathbf{x}$ denote the n-dimensional column vector with components

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

The vector with all elements equal to 1 is written $\mathbf{1}$, and similarly the vector with all zero values is written $\mathbf{0}$.

Definition A.1 (scalar product)**.** The *scalar product* $\mathbf{w} \cdot \mathbf{x}$ is defined as:

$$\mathbf{w} \cdot \mathbf{x} = \sum_{i=1}^{n} w_i x_i = \mathbf{w}^\mathsf{T}\mathbf{x} \tag{A.1.1}$$

The length of a vector is denoted $|\mathbf{x}|$, the squared length is given by

$$|\mathbf{x}|^2 = \mathbf{x}^\mathsf{T}\mathbf{x} = \mathbf{x}^2 = x_1^2 + x_2^2 + \cdots + x_n^2 \tag{A.1.2}$$

A *unit vector* $\mathbf{x}$ has $|\mathbf{x}| = 1$. The scalar product has a natural geometric interpretation as:

$$\mathbf{w} \cdot \mathbf{x} = |\mathbf{w}|\,|\mathbf{x}| \cos(\theta) \tag{A.1.3}$$

where θ is the angle between the two vectors. Thus if the lengths of two vectors are fixed their inner product is largest when $\theta = 0$, whereupon one vector is a constant multiple of the other. If the scalar product $\mathbf{x}^\mathsf{T}\mathbf{y} = 0$, then $\mathbf{x}$ and $\mathbf{y}$ are *orthogonal* (they are at right angles to each other). A set of vectors *orthonormal* if they are mutually orthogonal and have unit length.

Definition A.2 (Linear dependence)**.** A set of vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n$ is linearly dependent if there exists a vector $\mathbf{x}^j$ that can be expressed as a linear combination of the other vectors. If the only solution to

$$\sum_{i=1}^{n} \alpha_i \mathbf{x}^i = \mathbf{0} \tag{A.1.4}$$

is for all $\alpha_i = 0, i = 1, \ldots, n$, the vectors $\mathbf{x}^1, \ldots, \mathbf{x}^n$ are linearly independent.

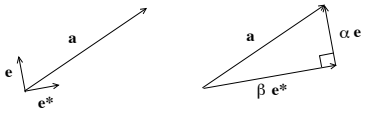

Figure A.1: Resolving a vector $\mathbf{a}$ into components along the orthogonal directions $\mathbf{e}$ and $\mathbf{e}^*$. The projection of $\mathbf{a}$ onto these two directions are lengths α and β along the directions $\mathbf{e}$ and $\mathbf{e}^*$.

A.1.2 The scalar product as a projection

Suppose that we wish to resolve the vector $\mathbf{a}$ into its components along the orthogonal directions specified by the unit vectors $\mathbf{e}$ and $\mathbf{e}^*$, see fig(A.1). That is $|\boldsymbol{e}| = |\boldsymbol{e}|^* = 1$ and $\mathbf{e} \cdot \mathbf{e^*} = 0$. We are required to find the scalar values α and β such that

$$\mathbf{a} = \alpha\mathbf{e} + \beta\mathbf{e}^* \tag{A.1.5}$$

From this we obtain

$$\mathbf{a} \cdot \mathbf{e} = \alpha\mathbf{e} \cdot \mathbf{e} + \beta\mathbf{e^*} \cdot \mathbf{e}, \qquad \mathbf{a} \cdot \mathbf{e^*} = \alpha\mathbf{e} \cdot \mathbf{e^*} + \beta\mathbf{e^*} \cdot \mathbf{e^*} \tag{A.1.6}$$

From the orthogonality and unit lengths of the vectors $\mathbf{e}$ and $\mathbf{e}^*$, this becomes simply

$$\mathbf{a} \cdot \mathbf{e} = \alpha, \qquad \mathbf{a} \cdot \mathbf{e^*} = \beta \tag{A.1.7}$$

This means that we can write the vector $\mathbf{a}$ in terms of the orthonormal components $\mathbf{e}$ and $\mathbf{e}^*$ as

$$\mathbf{a} = (\mathbf{a} \cdot \mathbf{e})\,\mathbf{e} + (\mathbf{a} \cdot \mathbf{e}^*)\,\mathbf{e}^* \tag{A.1.8}$$

The scalar product between $\mathbf{a}$ and $\mathbf{e}$ projects the vector $\mathbf{a}$ onto the (unit) direction $\mathbf{e}$. The projection of a vector $\mathbf{a}$ onto a direction specified by general $\mathbf{f}$ is $\frac{\mathbf{a}\cdot\mathbf{f}}{|\boldsymbol{f}|^2}\mathbf{f}$.

A.1.3 Lines in space

A line in 2 (or more) dimensions can be specified as follows. The vector of any point along the line is given, for some s, by the equation

$$\mathbf{p} = \mathbf{a} + s\mathbf{u}, \qquad s \in \mathcal{R}. \tag{A.1.9}$$

where $\mathbf{u}$ is parallel to the line, and the line passes through the point $\mathbf{a}$, see fig(A.2). An alternative specification can be given by realising that all vectors along the line are orthogonal to the normal of the line, $\mathbf{n}$ ($\mathbf{u}$ and $\mathbf{n}$ are orthonormal). That is

$$(\mathbf{p} - \mathbf{a}) \cdot \mathbf{n} = 0 \Leftrightarrow \mathbf{p} \cdot \mathbf{n} = \mathbf{a} \cdot \mathbf{n} \tag{A.1.10}$$

If the vector $\mathbf{n}$ is of unit length, the right hand side of the above represents the shortest distance from the origin to the line, drawn by the dashed line in fig(A.2) (since this is the projection of $\mathbf{a}$ onto the normal direction).

A.1.4 Planes and hyperplanes

To define a two-dimensional plane (in arbitrary dimensional space) one may specify two vectors $\mathbf{u}$ and $\mathbf{v}$ that lie in the plane (they need not be mutually orthogonal), and a position vector $\mathbf{a}$ in the plane, see fig(A.3). Any vector $\mathbf{p}$ in the plane can then be written as

$$\mathbf{p} = \mathbf{a} + s\mathbf{u} + t\mathbf{v}, \qquad (s, t) \in \mathcal{R}. \tag{A.1.11}$$

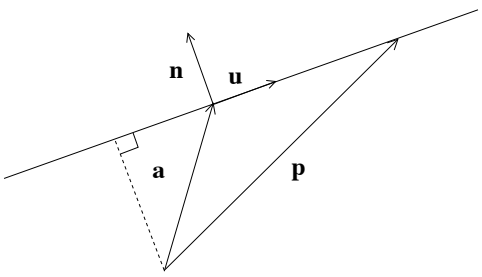

Figure A.2: A line can be specified by some position vector on the line, $\mathbf{a}$, and a unit vector along the direction of the line, $\mathbf{u}$. In 2 dimensions, there is a unique direction, $\mathbf{n}$, perpendicular to the line. In three dimensions, the vectors perpendicular to the direction of the line lie in a plane, whose normal vector is in the direction of the line, $\mathbf{u}$.

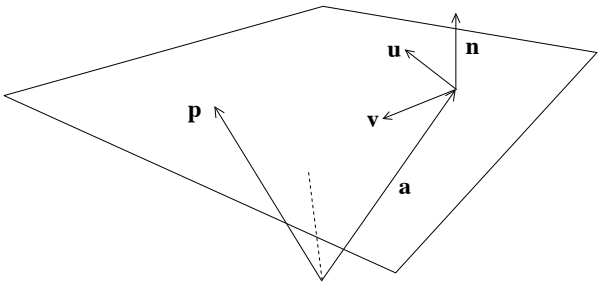

Figure A.3: A plane can be specified by a point in the plane, **a** and two, non-parallel directions in the plane, **u** and **v**. The normal to the plane is unique, and in the same direction as the directed line from the origin to the nearest point on the plane.

An alternative definition is given by considering that any vector within the plane must be orthogonal to the normal of the plane **n**.

$$(\mathbf{p}-\mathbf{a})\cdot\mathbf{n}=0\Leftrightarrow\mathbf{p}\cdot\mathbf{n}=\mathbf{a}\cdot\mathbf{n} \tag{A.1.12}$$

The right hand side of the above represents the shortest distance from the origin to the plane, drawn by the dashed line in fig(A.3). The advantage of this representation is that it has the same form as a line. Indeed, this representation of (hyper)planes is independent of the dimension of the space. In addition, only two quantities need to be defined – the normal to the plane and the distance from the origin to the plane.

A.1.5 Matrices

An $m\times n$ matrix $\mathbf{A}$ is a collection of scalar values arranged in a rectangle of m rows and n columns. A vector can be considered as an $n\times 1$ matrix. The i,j element of matrix $\mathbf{A}$ can be written A_{ij} or more conventionally a_{ij}. Where more clarity is required, one may write $[\mathbf{A}]_{ij}$.

Definition A.3 (Matrix addition)**.** For two matrices $\mathbf{A}$ and $\mathbf{B}$ of the same size,

$$[\mathbf{A}+\mathbf{B}]_{ij}=[\mathbf{A}]_{ij}+[\mathbf{B}]_{ij} \tag{A.1.13}$$

Definition A.4 (Matrix multiplication)**.** For an l by n matrix $\mathbf{A}$ and an n by m matrix B, the product $\mathbf{AB}$ is the l by m matrix with elements

$$[\mathbf{AB}]_{ik}=\sum_{j=1}^{n}[\mathbf{A}]_{ij}[\mathbf{B}]_{jk}\;;\qquad i=1,\ldots,l\quad k=1,\ldots,m\,. \tag{A.1.14}$$

Note that in general $\mathbf{BA}\neq\mathbf{AB}$. When $\mathbf{BA}=\mathbf{AB}$ we say that they $\mathbf{A}$ and $\mathbf{B}$ *commute*. The matrix $\mathbf{I}$ is the *identity matrix*, necessarily square, with 1's on the diagonal and 0's everywhere else. For clarity we may also write $\mathbf{I}_m$ for a square $m\times m$ identity matrix. Then for an $m\times n$ matrix $\mathbf{A}$, $\mathbf{I}_m\mathbf{A}=\mathbf{A}\mathbf{I}_n=\mathbf{A}$. The identity matrix has elements $[\mathbf{I}]_{ij}=\delta_{ij}$ given by the *Kronecker delta*:

$$\delta_{ij}\equiv\begin{cases}1 & i=j\\ 0 & i\neq j\end{cases} \tag{A.1.15}$$

Definition A.5 (Transpose)**.** The transpose $\mathbf{B}^\mathsf{T}$ of the n by m matrix $\mathbf{B}$ is the m by n matrix with components

$$\left[\mathbf{B}^\mathsf{T}\right]_{kj}=\mathbf{B}_{jk}\,;\qquad k=1,\ldots,m\quad j=1,\ldots,n\,. \tag{A.1.16}$$

$\left(\mathbf{B}^\mathsf{T}\right)^\mathsf{T}=\mathbf{B}$ and $(\mathbf{AB})^\mathsf{T}=\mathbf{B}^\mathsf{T}\mathbf{A}^\mathsf{T}$. If the shapes of the matrices $\mathbf{A}$,$\mathbf{B}$ and $\mathbf{C}$ are such that it makes sense to calculate the product $\mathbf{ABC}$, then

$$(\mathbf{ABC})^\mathsf{T}=\mathbf{C}^\mathsf{T}\mathbf{B}^\mathsf{T}\mathbf{A}^\mathsf{T} \tag{A.1.17}$$

A square matrix $\mathbf{A}$ is symmetric if $\mathbf{A}^\mathsf{T} = \mathbf{A}$. A square matrix is called *Hermitian* if $\mathbf{A} = \mathbf{A}^{\mathsf{T}*}$ where * denotes the complex conjugate operator. For Hermitian matrices, the eigenvectors form an orthogonal set with real eigenvalues.

Definition A.6 (Trace)**.**

$$\text{trace}\,(\mathbf{A}) = \sum_i A_{ii} = \sum_i \lambda_i \tag{A.1.18}$$

where λ_i are the eigenvalues of $\mathbf{A}$.

A.1.6 Linear transformations

If we define $\mathbf{u}_i$ to be the vector with zeros everywhere expect for the i^{th} entry, then a vector can be expressed as $\mathbf{x} = \sum_i x_i \mathbf{u}_i$. Then a linear transformation of $\mathbf{x}$ is given by

$$\mathbf{A}\mathbf{x} = \sum_i x_i \mathbf{A}\mathbf{u}_i = \sum_i x_i \mathbf{a}_i \tag{A.1.19}$$

where $\mathbf{a}_i$ is the i^{th} column of $\mathbf{A}$.

Rotations

The unit vectors $(1,0)^\mathsf{T}$ and $(0,1)^\mathsf{T}$ under rotation by θ radians transform to the vectors

$$\begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, \qquad \begin{pmatrix} -\sin\theta \\ \cos\theta \end{pmatrix} \tag{A.1.20}$$

which thus form the columns of the rotation matrix:

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{A.1.21}$$

Multiplying a vector by $\mathbf{R}$, $\mathbf{R}\mathbf{x}$, rotates the vector through θ radians.

A.1.7 Determinants

Definition A.7 (Determinant)**.** For a square matrix $\mathbf{A}$, the determinant is the volume of the transformation of the matrix $\mathbf{A}$ (up to a sign change). That is, we take a hypercube of unit volume and map each vertex under the transformation. The volume of the resulting object is defined as the determinant. Writing $[\mathbf{A}]_{ij} = a_{ij}$,

$$\det\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}a_{22} - a_{21}a_{12} \tag{A.1.22}$$

$$\det\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = a_{11}\left(a_{22}a_{33} - a_{23}a_{32}\right) - a_{12}\left(a_{21}a_{33} - a_{31}a_{23}\right) + a_{13}\left(a_{21}a_{32} - a_{31}a_{22}\right) \tag{A.1.23}$$

The determinant in the (3×3) case has the form

$$a_{11}\det\begin{pmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{pmatrix} - a_{12}\det\begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{pmatrix} + a_{13}\det\begin{pmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \tag{A.1.24}$$

The determinant of the (3×3) matrix $\mathbf{A}$ is given by the sum of terms $(-1)^{i+1}a_{1i}\det(\mathbf{A}_i)$ where $\mathbf{A}_i$ is the (2×2) matrix formed from $\mathbf{A}$ by removing the i^{th} row and column. This form of the determinant

generalises to any dimension. That is, we can define the determinant recursively as an expansion along the top row of determinants of reduced matrices. The absolute value of the determinant is the volume of the transformation.

$$\det\left(\mathbf{A}^\mathsf{T}\right) = \det\left(\mathbf{A}\right) \tag{A.1.25}$$

For square matrices $\mathbf{A}$ and $\mathbf{B}$ of equal dimensions,

$$\det\left(\mathbf{AB}\right) = \det\left(\mathbf{A}\right)\det\left(\mathbf{B}\right), \qquad \det\left(\mathbf{I}\right) = 1 \Rightarrow \det\left(\mathbf{A}^{-1}\right) = 1/\det\left(\mathbf{A}\right) \tag{A.1.26}$$

Definition A.8 (Orthogonal matrix)**.** A square matrix $\mathbf{A}$ is orthogonal if $\mathbf{A}\mathbf{A}^\mathsf{T} = \mathbf{I} = \mathbf{A}^\mathsf{T}\mathbf{A}$. From the properties of the determinant, we see therefore that an orthogonal matrix has determinant ± 1 and hence corresponds to a volume preserving transformation.

Definition A.9 (Matrix rank)**.** For an $m \times n$ matrix $\mathbf{X}$ the rank of $\mathbf{X}$ is the maximum number of linearly independent columns (or equivalently rows). The matrix is full rank if it has rank equal to $\min(m, n)$ – otherwise the matrix is rank deficient. A square matrix that is rank deficient is singular.

A.1.8 Matrix inversion

Definition A.10 (Matrix inversion)**.** For a square matrix $\mathbf{A}$, its inverse satisfies

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{I} = \mathbf{A}\mathbf{A}^{-1} \tag{A.1.27}$$

It is not always possible to find a matrix $\mathbf{A}^{-1}$ such that $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$, in which case $\mathbf{A}$ is *singular*. Geometrically, singular matrices correspond to projections: if we transform each of the vertices $\mathbf{v}$ of a binary hypercube using $\mathbf{Av}$, the volume of the transformed hypercube is zero. Hence if $\det\left(\mathbf{A}\right) = 0$, the matrix $\mathbf{A}$ is a form of projection or 'collapse' which means $\mathbf{A}$ is singular. Given a vector $\mathbf{y}$ and a singular transformation, $\mathbf{A}$, one cannot uniquely identify a vector $\mathbf{x}$ for which $\mathbf{y} = \mathbf{Ax}$. Provided the inverses exist

$$\left(\mathbf{AB}\right)^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \tag{A.1.28}$$

For a non-square matrix $\mathbf{A}$ such that $\mathbf{A}\mathbf{A}^\mathsf{T}$ is invertible, then the right pseudo inverse, defined as

$$\mathbf{A}^\dagger = \mathbf{A}^\mathsf{T}\left(\mathbf{A}\mathbf{A}^\mathsf{T}\right)^{-1} \tag{A.1.29}$$

satisfies $\mathbf{A}\mathbf{A}^\dagger = \mathbf{I}$. The left pseudo inverse is given by

$$\mathbf{A}^\dagger = \left(\mathbf{A}^\mathsf{T}\mathbf{A}\right)^{-1}\mathbf{A}^\mathsf{T} \tag{A.1.30}$$

and satisfies $\mathbf{A}^\dagger\mathbf{A} = \mathbf{I}$.

Definition A.11 (Matrix inversion lemma (Woodbury formula))**.** Provided the appropriate inverses exist:

$$\left(\mathbf{A} + \mathbf{U}\mathbf{V}^\mathsf{T}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\left(\mathbf{I} + \mathbf{V}^\mathsf{T}\mathbf{A}^{-1}\mathbf{U}\right)^{-1}\mathbf{V}^\mathsf{T}\mathbf{A}^{-1} \tag{A.1.31}$$

$$\det\left(\mathbf{A} + \mathbf{U}\mathbf{V}^\mathsf{T}\right) = \det\left(\mathbf{A}\right)\det\left(\mathbf{I} + \mathbf{V}^\mathsf{T}\mathbf{A}^{-1}\mathbf{U}\right) \tag{A.1.32}$$

Definition A.12 (Block matrix inversion)**.** For matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$, provided the appropriate inverses exist:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} & -\left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}\left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1} & \mathbf{D}^{-1}\mathbf{C}\left(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}\right)^{-1}\mathbf{B}\mathbf{D}^{-1} \end{bmatrix} \tag{A.1.33}$$

A.1.9 Computing the matrix inverse

For a 2×2 matrix, $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, the inverse matrix has elements

$$\frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \mathbf{A}^{-1} \tag{A.1.34}$$

The quantity $ad - bc$ is the determinant of $\mathbf{A}$. There are many ways to compute the inverse of a general matrix, and we refer the reader to more specialised texts, such as [279, 129].

If one wants to solve only a linear system, $\mathbf{A}\mathbf{x} = \mathbf{b}$, algebraically, the solution is given by $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. This would suggest that one needs to compute the $n \times n$ matrix $\mathbf{A}^{-1}$. However, in practice, $\mathbf{A}^{-1}$ is not explicitly required – only $\mathbf{x}$ is needed. This can be obtained more rapidly and with greater numerical precision using Gaussian elimination[279, 129].

A.1.10 Eigenvalues and eigenvectors

The eigenvectors of a matrix correspond to a natural coordinate system in which the geometric transformation represented by $\mathbf{A}$ can be most easily understood.

Definition A.13 (Eigenvalues and Eigenvectors)**.** For an $n \times n$ square matrix $\mathbf{A}$, $\mathbf{e}$ is an eigenvector of $\mathbf{A}$ with eigenvalue λ if

$$\mathbf{A}\mathbf{e} = \lambda\mathbf{e} \tag{A.1.35}$$

Geometrically, the eigenvectors are special directions such that the effect of the transformation $\mathbf{A}$ along a direction $\mathbf{e}$ is simply to scale the vector $\mathbf{e}$. For a rotation matrix $\mathbf{R}$ in general there will be no direction preserved under the rotation so that the eigenvalues and eigenvectors are complex valued (which is why the Fourier representation, which corresponds to representation in a rotated basis, is necessarily complex).

For an $(n \times n)$ dimensional matrix, there are (including repetitions) n eigenvalues, each with a corresponding eigenvector. We can reform equation (A.1.35) as

$$\left(\mathbf{A} - \lambda\mathbf{I}\right)\mathbf{e} = \mathbf{0} \tag{A.1.36}$$

We can write equation (A.1.36) as $\mathbf{B}\mathbf{e} = \mathbf{0}$, where $\mathbf{B} \equiv \mathbf{A} - \lambda\mathbf{I}$. If $\mathbf{B}$ has an inverse, then a solution is $\mathbf{e} = \mathbf{B}^{-1}\mathbf{0} = \mathbf{0}$, which trivially satisfies the eigen-equation. For any non-trivial solution to the problem $\mathbf{B}\mathbf{e} = \mathbf{0}$, we therefore need $\mathbf{B}$ to be non-invertible. This is equivalent to the condition that $\mathbf{B}$ has zero determinant. Hence λ is an eigenvalue of $\mathbf{A}$ if

$$\det\left(\mathbf{A} - \lambda\mathbf{I}\right) = 0 \tag{A.1.37}$$

This is known as the characteristic equation. This determinant equation will be a polynomial in λ of degree n and the resulting equation is known as the characteristic polynomial. Once we have found an eigenvalue, the corresponding eigenvector can be found by substituting this value for λ in equation (A.1.35) and solving the linear equations for $\mathbf{e}$. It may be that for an eigenvalue λ the eigenvector is not unique and there is a space of corresponding vectors. An important relation between the determinant and the eigenvalues of a matrix is

$$\det\left(\mathbf{A}\right) = \prod_{i=1}^{n} \lambda_i \tag{A.1.38}$$

Hence a matrix is singular if it has a zero eigenvalue. The trace of a matrix can be expressed as

$$\text{trace}\,(\mathbf{A}) = \sum_i \lambda_i \tag{A.1.39}$$

For a real symmetric matrix $\mathbf{A} = \mathbf{A}^\mathsf{T}$, and eigenvectors $\mathbf{e}^i$, $\mathbf{e}^j$, then $(\mathbf{e}^i)^\mathsf{T}\mathbf{e}^j = 0$ if the eigenvalues λ_i and λ_j are different. This can be shown by considering:

$$\mathbf{A}\mathbf{e}^i = \lambda_i \mathbf{e}^i \Rightarrow (\mathbf{e}^j)^\mathsf{T}\mathbf{A}\mathbf{e}^i = \lambda_i (\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i \tag{A.1.40}$$

Since $\mathbf{A}$ is symmetric, then

$$((\mathbf{e}^j)^\mathsf{T}\mathbf{A})\mathbf{e}^i = (\mathbf{A}\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i = \lambda_j (\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i \Rightarrow \lambda_i (\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i = \lambda_j (\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i \tag{A.1.41}$$

If $\lambda_i \neq \lambda_j$, this condition can be satisfied only if $(\mathbf{e}^j)^\mathsf{T}\mathbf{e}^i = 0$, namely that the eigenvectors are orthogonal.

Definition A.14 (Trace-Log formula)**.** For a positive definite matrix $\mathbf{A}$,

$$\text{trace}\,(\log \mathbf{A}) \equiv \log\det(\mathbf{A}) \tag{A.1.42}$$

Note that the above logarithm of a matrix is not the element-wise logarithm. In MATLAB the required function is `logm`. In general for an analytic function $f(x)$, $f(\mathbf{M})$ is defined via the power-series expansion of the function. On the right, since $\det(\mathbf{A})$ is a scalar, the logarithm is the standard logarithm of a scalar.

A.1.11 Matrix decompositions

Definition A.15 (Spectral decomposition)**.** A real $n \times n$ symmetric matrix $\mathbf{A}$ has an eigen-decomposition

$$\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{e}_i \mathbf{e}_i^\mathsf{T} \tag{A.1.43}$$

where λ_i is the eigenvalue of eigenvector $\mathbf{e}_i$ and the eigenvectors form an orthogonal set,

$$\left(\mathbf{e}^i\right)^\mathsf{T}\mathbf{e}^j = \delta_{ij}\left(\mathbf{e}^i\right)^\mathsf{T}\mathbf{e}^i \tag{A.1.44}$$

In matrix notation

$$\mathbf{A} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^\mathsf{T} \tag{A.1.45}$$

where $\mathbf{E} = \left[\mathbf{e}^1, \ldots, \mathbf{e}^n\right]$ is the matrix of eigenvectors and $\mathbf{\Lambda}$ the corresponding diagonal eigenvalue matrix. More generally, for a square non-symmetric diagonalisable $\mathbf{A}$ we can write

$$\mathbf{A} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^{-1} \tag{A.1.46}$$

Definition A.16 (Singular Value Decomposition)**.** The SVD decomposition of a $n \times p$ matrix $\mathbf{X}$ is

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T} \tag{A.1.47}$$

where $\dim \mathbf{U} = n \times n$ with $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}_n$. Also $\dim \mathbf{V} = p \times p$ with $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}_p$.

The matrix $\mathbf{S}$ has $\dim \mathbf{S} = n \times p$ with zeros everywhere except on the diagonal entries. The singular values are the diagonal entries $[\mathbf{S}]_{ii}$ and are non-negative. The singular values are ordered so that the upper The left diagonal element of $\mathbf{S}$ contains the largest singular value and $S_{ii} \geq S_{jj}$ for $i < j$. Assuming $n < p$ (otherwise transpose $\mathbf{X}$), the computational complexity of computing the SVD is $O\left(4n^2p + 8np^2 + 9n^3\right)$ [129]. For the 'thin' SVD with $n > p$ only the first p columns of U and $\mathbf{S}$ are computed giving a decomposition

$$\mathbf{X} = \mathbf{U}_p\mathbf{S}_p\mathbf{V}^\mathsf{T} \tag{A.1.48}$$

where $\mathbf{U}_p$ has dimension $n \times p$, and $\mathbf{S}_p$ is the $p \times p$ diagonal matrix of singular values.

Definition A.17 (Quadratic form)**.**

$$\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} + \mathbf{x}^\mathsf{T}\mathbf{b} \tag{A.1.49}$$

Definition A.18 (Positive definite matrix)**.** A symmetric matrix $\mathbf{A}$ with the property that $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} \geq 0$ for any vector $\mathbf{x}$ is called positive semidefinite. A symmetric matrix $\mathbf{A}$, with the property that $\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} > 0$ for any vector $\mathbf{x} \neq \mathbf{0}$ is called positive definite. A positive definite matrix has full rank and is thus invertible. Using the eigen-decomposition of $\mathbf{A}$,

$$\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} = \sum_i \lambda_i \mathbf{x}^\mathsf{T}\mathbf{e}^i(\mathbf{e}^i)^\mathsf{T}\mathbf{x} = \sum_i \lambda_i \left(\mathbf{x}^\mathsf{T}\mathbf{e}^i\right)^2 \tag{A.1.50}$$

which is greater than zero if and only if all the eigenvalues are positive. Hence $\mathbf{A}$ is positive definite if and only if all its eigenvalues are positive.

Eigenfunctions

$$\int_x K(x', x)\phi_a(x) = \lambda_a \phi_a(x') \tag{A.1.51}$$

The eigenfunctions of a real symmetric kernel, $K(x', x) = K(x, x')$ are orthogonal:

$$\int_x \phi_a(x)\phi_b^*(x) = \delta_{ab} \tag{A.1.52}$$

where $\phi^*(x)$ is the complex conjugate of $\phi(x)$. A kernel has a decomposition (provided the eigenvalues are countable)

$$K(x^i, x^j) = \sum_\mu \lambda_\mu \phi_\mu(x^i)\phi_\mu^*(x^j) \tag{A.1.53}$$

Then

$$\sum_{i,j} y_i K(x^i, x^j) y_j = \sum_{i,j,\mu} \lambda_\mu y_i \phi_\mu(x^i)\phi_\mu^*(x^j) y_j = \sum_\mu \lambda_\mu \underbrace{\left(\sum_i y_i \phi_\mu(x^i)\right)}_{z_i} \underbrace{\left(\sum_i y_i \phi_\mu^*(x^i)\right)}_{z_i^*} \tag{A.1.54}$$

which is greater than zero if the eigenvalues are all positive (since for complex z, $zz^* \geq 0$). If the eigenvalues are uncountable the appropriate decomposition is

$$K(x^i, x^j) = \int \lambda(s)\phi(x^i, s)\phi^*(x^j, s)ds \tag{A.1.55}$$

A.2 Multivariate Calculus

Definition A.19 (Partial derivative)**.** Consider a function of n variables, $f(x_1, x_2, \ldots, x_n) \equiv f(\mathbf{x})$. The partial derivative of f *w.r.t.* x_i is defined as the following limit (when it exists)

$$\frac{\partial f}{\partial x_i} = \lim_{h \to 0} \frac{f(x_1, \ldots, x_{i-1}, x_i + h, x_{i+1}, \ldots, x_n) - f(\mathbf{x})}{h} \tag{A.2.1}$$

The *gradient vector* of f is denoted ∇f or $\mathbf{g}$:

$$\nabla f(\mathbf{x}) \equiv \mathbf{g}(\mathbf{x}) \equiv \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix} \tag{A.2.2}$$

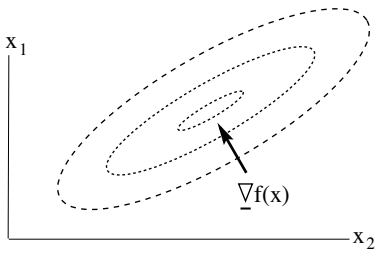

Figure A.4: Interpreting the gradient. The ellipses are contours of constant function value, $f = \mathsf{const}$. At any point $\mathbf{x}$, the gradient vector $\nabla f(\mathbf{x})$ points along the direction of maximal increase of the function.

A.2.1 Interpreting the gradient vector

Consider a function $f(\mathbf{x})$ that depends on a vector $\mathbf{x}$. We are interested in how the function changes when the vector $\mathbf{x}$ changes by a small amount : $\mathbf{x} \to \mathbf{x} + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a vector whose length is very small. According to a Taylor expansion, the function f will change to

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + \sum_i \delta_i \frac{\partial f}{\partial x_i} + O\left(\boldsymbol{\delta}^2\right) \tag{A.2.3}$$

We can interpret the summation above as the scalar product between the vector ∇f with components $[\nabla f]_i = \frac{\partial f}{\partial x_i}$ and $\boldsymbol{\delta}$.

$$f(\mathbf{x} + \boldsymbol{\delta}) = f(\mathbf{x}) + (\nabla f) \cdot \boldsymbol{\delta} + O\left(\boldsymbol{\delta}^2\right) \tag{A.2.4}$$

The gradient points along the direction in which the function increases most rapidly. To see this, consider a direction $\hat{\mathbf{p}}$ (a unit length vector). Then a displacement, δ units along this direction changes the function value to

$$f(\mathbf{x} + \delta\hat{\mathbf{p}}) \approx f(\mathbf{x}) + \delta \nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}} \tag{A.2.5}$$

The direction $\hat{\mathbf{p}}$ for which the function has the largest change is that which maximises the overlap

$$\nabla f(\mathbf{x}) \cdot \hat{\mathbf{p}} = |\nabla f(\mathbf{x})||\hat{\mathbf{p}}| \cos\theta = |\nabla f(\mathbf{x})| \cos\theta \tag{A.2.6}$$

where θ is the angle between $\hat{\mathbf{p}}$ and $\nabla f(\mathbf{x})$. The overlap is maximised when $\theta = 0$, giving $\hat{\mathbf{p}} = \nabla f(\mathbf{x})/|\nabla f(\mathbf{x})|$. Hence, the direction along which the function changes the most rapidly is along $\nabla f(\mathbf{x})$.

A.2.2 Higher derivatives

The second derivative of an n-variable function is defined by

$$\frac{\partial}{\partial x_i}\left(\frac{\partial f}{\partial x_j}\right) \qquad i = 1, \ldots, n; \;\; j = 1, \ldots, n \tag{A.2.7}$$

which is usually written

$$\frac{\partial^2 f}{\partial x_i \partial x_j}, \;\; i \neq j \qquad \frac{\partial^2 f}{\partial {x_i}^2}, \;\; i = j \tag{A.2.8}$$

If the partial derivatives $\partial^2 f/\partial x_i \partial x_j$ and $\partial^2 f/\partial x_j \partial x_i$ exist then

$$\partial^2 f/\partial x_i \partial x_j = \partial^2 f/\partial x_j \partial x_i \,. \tag{A.2.9}$$

This is also denoted by $\nabla\nabla f$. These n^2 second partial derivatives are represented by a square, symmetric matrix called the *Hessian* matrix of $f(\mathbf{x})$.

$$\mathbf{H}_f(\mathbf{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial {x_n}^2} \end{pmatrix} \tag{A.2.10}$$

Definition A.20 (Chain rule)**.** Let each x_j be parameterized by $u_1, \ldots, u_m$, *i.e.* $x_j = x_j(u_1, \ldots, u_m)$.

$$\frac{\partial f}{\partial u_\alpha} = \sum_{j=1}^{n} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial u_\alpha} \tag{A.2.11}$$

or in vector notation

$$\frac{\partial}{\partial u_\alpha} f(\mathbf{x}(\mathbf{u})) = \nabla f^{\mathsf{T}}(\mathbf{x}(\mathbf{u})) \frac{\partial \mathbf{x}(\mathbf{u})}{\partial u_\alpha} \tag{A.2.12}$$

Definition A.21 (Directional derivative)**.** Assume f is differentiable. We define the scalar directional derivative $(D_{\mathbf{v}} f)(\mathbf{x}^*)$ of f in a direction $\mathbf{v}$ at a point $\mathbf{x}^*$. Let $\mathbf{x} = \mathbf{x}^* + h\mathbf{v}$, Then

$$(D_{\mathbf{v}} f)(\mathbf{x}^*) = \left. \frac{d}{dh} f(\mathbf{x}^* + h\mathbf{v}) \right|_{h=0} = \left. \sum_j v_j \frac{\partial f}{\partial x_j} \right|_{\mathbf{x}=\mathbf{x}^*} = \nabla f^{\mathsf{T}} \mathbf{v} \tag{A.2.13}$$

A.2.3 Matrix calculus

Definition A.22 (Derivative of a matrix trace)**.** For matrices $\mathbf{A}$ and $\mathbf{B}$

$$\frac{\partial}{\partial \mathbf{A}} \text{trace}\,(\mathbf{AB}) \equiv \mathbf{B}^{\mathsf{T}} \tag{A.2.14}$$

Definition A.23 (Derivative of $\log\det(\mathbf{A})$)**.**

$$\partial \log\det(\mathbf{A}) = \partial \text{trace}\,(\log \mathbf{A}) = \text{trace}\left(\mathbf{A}^{-1} \partial \mathbf{A}\right) \tag{A.2.15}$$

So that

$$\frac{\partial}{\partial \mathbf{A}} \log\det(\mathbf{A}) = \mathbf{A}^{-\mathsf{T}} \tag{A.2.16}$$

Definition A.24 (Derivative of a matrix inverse)**.** For an invertible matrix $\mathbf{A}$,

$$\partial \mathbf{A}^{-1} \equiv -\mathbf{A}^{-\mathsf{T}} \partial \mathbf{A} \mathbf{A}^{-1} \tag{A.2.17}$$

A.3 Inequalities

A.3.1 Convexity

Definition A.25 (Convex function)**.** A function $f(x)$ is defined as convex if for any x, y and $0 \leq \lambda \leq 1$

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y) \tag{A.3.1}$$

If $-f(x)$ is convex, $f(x)$ is called concave.

An intuitive picture is given by considering the quantity $\lambda x + (1-\lambda)y$. As we vary λ from 0 to 1, this traces points between x ($\lambda = 0$) and y ($\lambda = 1$). Hence for $\lambda = 0$ we start at the point $x, f(x)$ and as λ increase trace a straight line towards the point $y, f(y)$ at $\lambda = 1$. Convexity states that the function f always lies

below this straight line. Geometrically this means that the function $f(x)$ is always always non-decreasing. Hence if $d^2f(x)/dx^2 \geq 0$ the function is convex. As an example, the function $\log x$ is concave since its second derivative is negative:

$$\frac{d}{dx}\log x = \frac{1}{x}, \qquad \frac{d^2}{dx^2}\log x = -\frac{1}{x^2} \tag{A.3.2}$$

A.3.2 Jensen's inequality

For a convex function, $f(x)$, it follows directly from the definition of convexity that

$$f(\langle x \rangle_{p(x)}) \leq \langle f(x) \rangle_{p(x)} \tag{A.3.3}$$

for any distribution $p(x)$.

A.4 Optimisation

Definition A.26 (Critical point)**.** When all first-order partial derivatives at a point are zero (*i.e.* $\nabla f = \mathbf{0}$) then the point is said to be a stationary or critical point. A critical point can correspond to a minimum, maximum or saddle point of the function.

There is a minimum of f at $\mathbf{x}^*$ if $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for all $\mathbf{x}$ sufficiently close to $\mathbf{x}^*$. This requires $\mathbf{x}^*$ to be a stationary point, $\nabla f(\mathbf{x}^*) = \mathbf{0}$. The Taylor expansion of f at the optimum is given by

@@ $$f(\mathbf{x}^* + h\mathbf{v}) = f(\mathbf{x}^*) + \frac{1}{2}h^2\mathbf{v}^\mathsf{T}\mathbf{H}_f\mathbf{v} + O(h^3) \tag{A.4.1}$$

Thus the minimum condition requires that $\mathbf{v}^\mathsf{T}\mathbf{H}_f\mathbf{v} \geq 0$, *i.e.* the Hessian is non-negative definite.

Definition A.27 (Conditions for a minimum)**.** Sufficient conditions for a minimum at $\mathbf{x}^*$ are (i) $\nabla f(\mathbf{x}^*) = 0$ and (ii) $\mathbf{H}_f(\mathbf{x}^*)$ is positive definite.

For a quadratic function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - \mathbf{b}^\mathsf{T}\mathbf{x} + c$, with symmetric $\mathbf{A}$ the condition $\nabla f(\mathbf{x}^*) = \mathbf{0}$ reads:

$$\mathbf{A}\mathbf{x}^* - \mathbf{b} = 0 \tag{A.4.2}$$

If $\mathbf{A}$ is invertible this equation has the unique solution $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$. If $\mathbf{A}$ is positive definite then $\mathbf{x}^*$ corresponds to the minimum point.

A.5 Multivariate Optimisation

In most cases the optima of functions cannot be found by algebraic means alone and numerical techniques are required. The search techniques that we consider here are iterative, *i.e.* we proceed towards the minimum $\mathbf{x}^*$ by a sequence of steps. Perhaps the simplest approach to minimising a multivariate function $f(\mathbf{x})$ is to break the problem into a sequence of one-dimensional problems. By initialising the elements of $\mathbf{x}$ at random, one then selects a component of $\mathbf{x}$ to update, keeping all others fixed. This coordinatewise optimisation decreases the function via a sequence of one-dimensional minimisations. Whilst such a procedure is simple, it can be inefficient in high-dimensions, particularly when there are strong dependencies between the components in $\mathbf{x}$. For this reason, it is useful to use gradient-based procedures that take the local geometry of the objective into account. We will consider a general class of iterative gradient-based method for which, on the k^{th} step, we take a step of length α_k in the direction $\mathbf{p}_k$,

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k \tag{A.5.1}$$

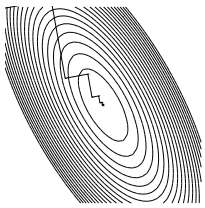

Figure A.5: Optimisation using line search along steepest descent directions. Following the steepest way downhill from a point (and continuing for a finite time in that direction) doesn't always result in the fastest way to get to the bottom.

A.5.1 Gradient descent with fixed stepsize

Locally, if we are at point $\mathbf{x}_k$, we can decrease $f(\mathbf{x})$ by taking a step in the direction $-\mathbf{g}(\mathbf{x})$. To see why gradient descent works, consider the general update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla_{\mathbf{x}} f \tag{A.5.2}$$

For small α we can expand f around $\mathbf{x}_k$ using Taylor's theorem:

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) \approx f(\mathbf{x}_k) - \alpha ||\nabla_{\mathbf{x}} f||| \tag{A.5.3}$$

so that the change in f is $\Delta f = -\alpha ||\nabla_{\mathbf{x}} f||^2$. If α is non-infinitesimal, it is always possible that we will step over the true minimum. Making η very small guards against this, but means that the optimization process will take a very long time to reach a minimum. A simple idea that can improve convergence of gradient descent is to include at each iteration a proportion of the change from the previous iteration, $\mathbf{p}_k = -\mathbf{g}_k - \beta \mathbf{g}_{k-1}$, where β is the *momentum coefficient*.

An unfortunate aspect of gradient descent is that the change in the function value depends on the coordinate system. Consider a new coordinate system $\mathbf{x} = \mathbf{M}\mathbf{y}$ for an invertible square matrix $\mathbf{M}$. Define $\hat{f}(\mathbf{y}) \equiv f(\mathbf{x})$. @@ Then the change in the function $\hat{f}$ under a gradient update is

$$\Delta \hat{f} \equiv \hat{f}(\mathbf{y} - \alpha \nabla_{\mathbf{y}} \hat{f}) - \hat{f}(\mathbf{y}) \approx -\alpha ||\nabla_{\mathbf{y}} \hat{f}||^2 \tag{A.5.4}$$

In the original coordinate system, the change in $\mathbf{x}$ is

$$\Delta \mathbf{x} = -\alpha \nabla_{\mathbf{x}} f$$

Since $\nabla_{\mathbf{y}} \hat{f} = \mathbf{M}^{\mathsf{T}} \nabla_{\mathbf{x}} f(\mathbf{x})$, transforming the change in $\mathbf{y}$ to a change in $\mathbf{x}$ we have

$$\mathbf{M}\Delta \mathbf{y} = -\alpha \mathbf{M}\mathbf{M}^{\mathsf{T}} \nabla_{\mathbf{x}} f(\mathbf{x})$$

which, unless $\mathbf{M}\mathbf{M}^{\mathsf{T}} = \mathbf{I}$, is not equal to $\Delta \mathbf{x}$. Similarly the change in the function value is

$$\Delta \hat{f} = -\alpha \nabla_{\mathbf{x}} f(\mathbf{x})^{\mathsf{T}} \mathbf{M}\mathbf{M}^{\mathsf{T}} \nabla_{\mathbf{x}} f(\mathbf{x}) \tag{A.5.5}$$

which, except for an orthogonal $\mathbf{M}$, is not equal to $\Delta f = -\alpha ||\nabla_{\mathbf{x}} f||^2$.

A.5.2 Gradient descent with line searches

An extension to the idea of gradient descent is to choose the direction of steepest descent, as indicated by the gradient $\mathbf{g}$, but to calculate the value of the step to take which most reduces the value of f when moving in that direction. This involves solving the one-dimensional problem of minimizing $f(\mathbf{x}_k + \alpha_k \mathbf{g}_k)$ with respect to α_k, and is known as a line search. To find the optimal step size at the k-th step, we choose α_k to minimize $f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$. So setting $F(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{p}_k)$, at this step we solve the one-dimensional minimization problem for $F(\lambda)$. Thus our choice of $\alpha_k = \lambda^*$ will satisfy $F'(\alpha_k) = 0$. Now

$$\begin{aligned} F'(\alpha_k) &= \frac{d}{dh} F(\alpha_k + h)_{|h=0} = \frac{d}{dh} f(\mathbf{x}_k + \alpha_k \mathbf{p}_k + h\mathbf{p}_k)_{|h=0} \\ &= \frac{d}{dh} f(\mathbf{x}_{k+1} + h\mathbf{p}_k)_{|h=0} = (D_{\mathbf{p}_k} f)(\mathbf{x}_{k+1}) = \nabla f^{\mathsf{T}}(\mathbf{x}_{k+1}) \mathbf{p}_k \end{aligned} \tag{A.5.6}$$

So $F'(\alpha_k) = 0$ means the directional derivative in the search direction must vanish at the new point and this gives condition $0 = \mathbf{g}_{k+1}^{\mathsf{T}} \mathbf{p}_k$. If the step size is chosen to reduce f as much as it can in that direction, then no further decrease in E can be made by moving in that direction for the moment. Thus the next step will have no component in that direction and will be at right angles to the previous just taken. This can lead to zig-zag type behaviour in the optimisation.

A.5.3 Minimising quadratic functions using line search

Consider minimising the quadratic function

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - \mathbf{b}^\mathsf{T}\mathbf{x} + c \tag{A.5.7}$$

where $\mathbf{A}$ is positive definite and symmetric. Although we know where the minimum of this simple function, based on using linear algebra, we wish to use this function as a toy model for more complex functions. As we 'zoom in' to the minimum of a smooth function, it will increasingly appear quadratic. Hence at such small scales, methods that work for the quadratic case should work in general smooth functions. If the general function looks roughly quadratic on the larger scale, then these methods will also work well in that case. One approach is to search along a particular direction $\mathbf{p}$, and find a minimum along this direction. We can then search for a deeper minimum by looking in different directions. That is, we can search firstly along a line $\mathbf{x} + \lambda\mathbf{p}$ such that the function attains a minimum. This has solution,

$$\lambda = \frac{(\mathbf{b} - \mathbf{A}\mathbf{x}) \cdot \mathbf{p}}{\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{p}} = \frac{-\nabla f(\mathbf{x}) \cdot \mathbf{p}}{\mathbf{p}^\mathsf{T}\mathbf{A}\mathbf{p}} \tag{A.5.8}$$

How should we now choose the next line search direction $\mathbf{p}^{new}$? It would seem sensible to choose successive line search directions $\mathbf{p}$ according to $\mathbf{p}^{new} = -\nabla f(\mathbf{x})$, so that each time we minimise the function along the line of steepest descent. However, this is generally not the optimal choice, see fig(A.5). If the matrix $\mathbf{A}$ were diagonal, then the minimisation is straightforward and can be carried out independently for each dimension. If we could therefore find an invertible matrix $\mathbf{P}$ with the property that $\mathbf{P}^\mathsf{T}\mathbf{A}\mathbf{P}$ is diagonal then the solution is easy since for

$$\hat{f}(\hat{\mathbf{x}}) = \frac{1}{2}\hat{\mathbf{x}}^\mathsf{T}\mathbf{P}^\mathsf{T}\mathbf{A}\mathbf{P}\hat{\mathbf{x}} - \mathbf{b}^\mathsf{T}\mathbf{P}\hat{\mathbf{x}} + c \tag{A.5.9}$$

with $\mathbf{x} = \mathbf{P}\hat{\mathbf{x}}$, we can compute the minimum for each dimension of $\hat{\mathbf{x}}$ separately and then retransform to find $\mathbf{x}^* = \mathbf{P}\hat{\mathbf{x}}^*$. The columns of such a matrix $\mathbf{P}$ are called conjugate vectors.

Definition A.28 (Conjugate vectors)**.** The vectors $\mathbf{p}_i$, $i = 1, \ldots, k$ are called conjugate to the matrix $\mathbf{A}$, if and only if for $i, j = 1, \ldots, k$ and $i \neq j$:

$$\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_j = 0 \quad \text{and} \quad \mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_i > 0\,. \tag{A.5.10}$$

The two conditions guarantee that conjugate vectors are linearly independent: Assume that

$$\mathbf{0} = \sum_{j=1}^{k} \alpha_j \mathbf{p}_j = \sum_{j=1}^{i-1} \alpha_j \mathbf{p}_j + \alpha_i \mathbf{p}_i + \sum_{j=i+1}^{k} \alpha_j \mathbf{p}_j \tag{A.5.11}$$

Now multiplying from the left with $\mathbf{p}_i^\mathsf{T}\mathbf{A}$ yields $0 = \alpha_i \mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_i$. So α_i is zero since we know that $\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_i > 0$. As we can make this argument for any $i = 1, \ldots, k$, all of the α_i must be zero.

A.5.4 Gram-Schmidt construction of conjugate vectors

Assume we already have k conjugate vectors $\mathbf{p}_1, \ldots, \mathbf{p}_k$ and let $\mathbf{v}$ be a vector which is linearly independent of $\mathbf{p}_1, \ldots, \mathbf{p}_k$. We then use a Gram-Schmidt procedure:

$$\mathbf{p}_{k+1} = \mathbf{v} - \sum_{j=1}^{k} \frac{\mathbf{p}_j^\mathsf{T}\mathbf{A}\mathbf{v}}{\mathbf{p}_j^\mathsf{T}\mathbf{A}\mathbf{p}_j}\mathbf{p}_j \tag{A.5.12}$$

for which it is clear that the vectors $\mathbf{p}_1, \ldots, \mathbf{p}_{k+1}$ are conjugate if $\mathbf{A}$ is positive definite. We can construct n conjugate vectors for a positive definite matrix in the following way: We start with n linearly independent vectors $\mathbf{u}_1, \ldots, \mathbf{u}_n$. We then set $\mathbf{p}_1 = \mathbf{u}_1$ and use (A.5.12) to compute $\mathbf{p}_2$ from $\mathbf{p}_1$ and $\mathbf{v} = \mathbf{u}_2$. Next we set $\mathbf{v} = \mathbf{u}_3$ and compute $\mathbf{p}_3$ from $\mathbf{p}_1, \mathbf{p}_2$ and $\mathbf{v}$. Continuing in this manner we obtain n conjugate vectors. Note that at each stage of the procedure the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ span the same subspace as the vectors $\mathbf{p}_1, \ldots, \mathbf{p}_k$.

A.5.5 The conjugate vectors algorithm

Let us assume that when minimising $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x} - \mathbf{b}^\mathsf{T}\mathbf{x} + c$ we first construct n vectors $\mathbf{p}_1, \ldots, \mathbf{p}_n$ conjugate to $\mathbf{A}$ which we use as our search directions. Using this our iterative solution takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \,. \tag{A.5.13}$$

where at each step we choose α_k by an exact line search with

$$\alpha_k = -\frac{\mathbf{p}_k^\mathsf{T}\mathbf{g}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k} \,. \tag{A.5.14}$$

This conjugate vectors algorithm has the geometrical interpretation that not only is the directional derivative zero at the new point along the direction $\mathbf{p}_k$, it is zero along all the previous search directions $\mathbf{p}_1, \ldots, \mathbf{p}_k$, known as the Luenberger expanding subspace theorem. In particular $\nabla f^\mathsf{T}(\mathbf{x}_{n+1})\mathbf{p}_i = 0$, for $i = 1, \ldots, n$; that is

$$\nabla f^\mathsf{T}(\mathbf{x}_{n+1})(\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n) = \mathbf{0} \,. \tag{A.5.15}$$

The square matrix $\mathbf{P} = (\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_n)$ is invertible since the $\mathbf{p}_i$ are conjugate, so $\nabla f(\mathbf{x}_{n+1}) = \mathbf{0}$ and the point $\mathbf{x}_{n+1}$ is therefore the minimum $\mathbf{x}^*$ of the quadratic function f. So in contrast to gradient descent, for a quadratic function the conjugate vectors algorithm converges in a finite number of steps.

A.5.6 The conjugate gradients algorithm

The conjugate gradients algorithm is a special case of the conjugate vectors algorithm in which we construct the conjugate vectors on-the-fly. After k-steps of the conjugate vectors algorithm we need to construct a vector $\mathbf{p}_{k+1}$ which is conjugate to $\mathbf{p}_1, \ldots, \mathbf{p}_k$. In the conjugate gradients algorithm one makes the special choice $\mathbf{v} = -\nabla f(\mathbf{x}_{k+1})$. The gradient at the new point $\mathbf{x}_{k+1}$ is orthogonal to $\mathbf{p}_i$, $i = 1, \ldots, k$, so $\nabla f(\mathbf{x}_{k+1})$ is linearly independent of $\mathbf{p}_1, \ldots, \mathbf{p}_k$ and a valid choice for $\mathbf{v}$, unless $\nabla f(\mathbf{x}_{k+1}) = \mathbf{0}$. In the latter case $\mathbf{x}_{k+1}$ is our minimum and the algorithm terminates. Using the notation $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$, the equation for the new search direction given by the Gram-Schmidt procedure equation (A.5.12) is:

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \sum_{i=1}^{k} \frac{\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{g}_{k+1}}{\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_i}\mathbf{p}_i \,. \tag{A.5.16}$$

Since $\mathbf{g}_{k+1}$ is orthogonal to $\mathbf{p}_i$, $i = 1, \ldots, k$, we have $\mathbf{p}_{k+1}^\mathsf{T}\mathbf{g}_{k+1} = -\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}$. So α_{k+1} can be written as

$$\alpha_{k+1} = \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{p}_{k+1}^\mathsf{T}\mathbf{A}\mathbf{p}_{k+1}} \,, \tag{A.5.17}$$

and in particular $\alpha_{k+1} \neq 0$. We now want to show that because we have been using the conjugate gradients algorithm at the previous steps as well, in equation (A.5.16) all terms but the last in the sum over i vanish. We shall assume that $k > 0$ since in the first step ($k = 0$) we just set $\mathbf{p}_1 = -\mathbf{g}_1$. First note that

$$\mathbf{g}_{i+1} - \mathbf{g}_i = \mathbf{A}\mathbf{x}_{i+1} - \mathbf{b} - (\mathbf{A}\mathbf{x}_i - \mathbf{b}) = \mathbf{A}(\mathbf{x}_{i+1} - \mathbf{x}_i) = \alpha_i \mathbf{A}\mathbf{p}_i \tag{A.5.18}$$

and since $\alpha_i \neq 0$, then $\mathbf{A}\mathbf{p}_i = (\mathbf{g}_{i+1} - \mathbf{g}_i)/\alpha_i$. So in equation (A.5.16):

$$\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{g}_{k+1} = \mathbf{g}_{k+1}^\mathsf{T}\mathbf{A}\mathbf{p}_i = \mathbf{g}_{k+1}^\mathsf{T}(\mathbf{g}_{i+1} - \mathbf{g}_i)/\alpha_i = (\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{i+1} - \mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_i)/\alpha_i \tag{A.5.19}$$

Since the $\mathbf{p}_i$ were obtained by applying the Gram-Schmidt procedure to the gradients $\mathbf{g}_i$, we have $\mathbf{g}_{k+1}^\mathsf{T}\mathbf{p}_i = 0$ and $\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_i = 0$ for $i = 1, \ldots, k$. This shows that

$$\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{g}_{k+1} = (\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{i+1} - \mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_i)/\alpha_i = \begin{cases} 0 & \text{if } 1 \leq i < k \\ \mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}/\alpha_k & \text{if } i = k \end{cases} \tag{A.5.20}$$

Hence equation (A.5.16) simplifies to

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}/\alpha_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}\mathbf{p}_k \,. \tag{A.5.21}$$

Algorithm A.1 Conjugate Gradients for minimising a function $f(\mathbf{x})$

1: $k = 1$
2: Choose $\mathbf{x}_1$.
3: $\mathbf{p}_1 = -\mathbf{g}_1$
4: **while** $\mathbf{g}_k \neq \mathbf{0}$ **do**
5: $\quad \alpha_k = \underset{\alpha_k}{\operatorname{argmin}}\ f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ ▷ Line Search
6: $\quad \mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$
7: $\quad \beta_k := \mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}/(\mathbf{g}_k^\mathsf{T}\mathbf{g}_k)$
8: $\quad \mathbf{p}_{k+1} := -\mathbf{g}_{k+1} + \beta_k \mathbf{p}_k$
9: $\quad k = k + 1$
10: **end while**

This can be brought into an even simpler form by applying equation (A.5.17) to α_k:

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}\frac{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}{\mathbf{g}_k^\mathsf{T}\mathbf{g}_k}\mathbf{p}_k = -\mathbf{g}_{k+1} + \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{g}_k^\mathsf{T}\mathbf{g}_k}\mathbf{p}_k \tag{A.5.22}$$

We shall write this in the form

$$\mathbf{p}_{k+1} = -\mathbf{g}_{k+1} + \beta_k \mathbf{p}_k \quad \text{where } \beta_k = \frac{\mathbf{g}_{k+1}^\mathsf{T}\mathbf{g}_{k+1}}{\mathbf{g}_k^\mathsf{T}\mathbf{g}_k}\,. \tag{A.5.23}$$

The formula (A.5.23) for β_k is due to Fletcher and Reeves[244]. Since the gradients are orthogonal, β_k can also be written as

$$\beta_k = \frac{\mathbf{g}_{k+1}^\mathsf{T}(\mathbf{g}_{k+1} - \mathbf{g}_k)}{\mathbf{g}_k^\mathsf{T}\mathbf{g}_k}\,, \tag{A.5.24}$$

this is the Polak-Ribière formula. The choice between the two expressions for β_k can be of some importance if f is not quadratic with the Polak-Ribière formula the most commonly used. The important point here is that we derived the above algorithm for the particular case of minimising a quadratic function f. By writing the algorithm however only in terms of f and its gradients, we make no explicit reference to the fact that f is quadratic. This means that we can therefore apply this algorithm to non-quadratic functions as well. The more similar the function f is to a quadratic function, the more confident we can be that the algorithm will find the minimum.

A.5.7 Newton's method

Consider a function $f(\mathbf{x})$ that we wish to find the minimum of. A Taylor expansion up to second order gives

$$f(\mathbf{x} + \boldsymbol{\Delta}) = f(\mathbf{x}) + \boldsymbol{\Delta}^\mathsf{T}\nabla f + \frac{1}{2}\boldsymbol{\Delta}^\mathsf{T}\mathbf{H}_f\boldsymbol{\Delta} + O(|\boldsymbol{\Delta}|^3) \tag{A.5.25}$$

The matrix $\mathbf{H}_f$ is the Hessian. Differentiating the right hand side with respect to $\boldsymbol{\Delta}$ (or, equivalently, completing the square), we find that the right hand side (ignoring the $O(|\boldsymbol{\Delta}|^3)$ term) has its lowest value when

$$\nabla f = -\mathbf{H}_f\boldsymbol{\Delta} \Rightarrow \boldsymbol{\Delta} = -\mathbf{H}_f^{-1}\nabla f \tag{A.5.26}$$

@@ Hence, an optimisation routine to minimise f is given by the Newton update

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \epsilon\mathbf{H}_f^{-1}\nabla f \tag{A.5.27}$$

where the scalar $0 < \epsilon < 1$ is often used in practice to improve the convergence. A benefit of Newton method over gradient descent is that the decrease in the objective function is invariant under a linear change of co-
@@ ordinates, $\mathbf{x} = \mathbf{M}\mathbf{y}$. Defining $\hat{f}(\mathbf{y}) \equiv f(\mathbf{x})$, since $\mathbf{H}_{\hat{f}} = \mathbf{M}^\mathsf{T}\mathbf{H}_f\mathbf{M}$ and $\nabla_\mathbf{y}\hat{f} = \mathbf{M}^\mathsf{T}\nabla_\mathbf{x}f$ the change in the original $\mathbf{x}$ system is given by

$$-\epsilon\mathbf{M}\Delta\mathbf{y} = -\epsilon\mathbf{M}\left(\mathbf{M}^\mathsf{T}\mathbf{H}_f\mathbf{M}\right)^{-1}\mathbf{M}^\mathsf{T}\nabla_\mathbf{x}f = -\epsilon\mathbf{H}_f^{-1}\nabla_\mathbf{x}f$$

which means that the change in the variable (and hence function value) is independent of the coordinate system.

Algorithm A.2 Quasi-Newton for minimising a function $f(\mathbf{x})$

1: $k = 1$
2: Choose $\mathbf{x}_1$
3: $\tilde{\mathbf{H}}_1 = \mathbf{I}$
4: **while** $\mathbf{g}_k \neq \mathbf{0}$ **do**
5: $\quad \mathbf{p}_k = -\tilde{\mathbf{H}}_k \mathbf{g}_k$
6: $\quad \alpha_k = \underset{\alpha_k}{\operatorname{argmin}}\ f(\mathbf{x}_k + \alpha_k \mathbf{p}_k)$ ▷ Line Search
7: $\quad \mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{p}_k$
8: $\quad \mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$, and update $\tilde{\mathbf{H}}_{k+1}$
9: $\quad k = k + 1$
10: **end while**

Quasi-Newton methods

For large-scale problems both storing the Hessian and solving the resulting linear system is computationally demanding, especially if the matrix is close to singular. An alternative is to set up the iteration

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{S}_k \mathbf{g}_k. \tag{A.5.28}$$

For $\mathbf{S}_k = \mathbf{A}^{-1}$ we have Newton's method, while if $\mathbf{S}_k = \mathbf{I}$ we have steepest descent. In general it would seem to be a good idea to choose $\mathbf{S}_k$ to be an approximation to the inverse Hessian. Also note that it is important that $\mathbf{S}_k$ be positive definite so that for small α_k we obtain a descent method. The idea behind most quasi-Newton methods is to try to construct an approximate inverse Hessian $\tilde{\mathbf{H}}_k$ using information gathered as the descent progresses, and to set $\mathbf{S}_k = \tilde{\mathbf{H}}_k$. As we have seen, for a quadratic optimization problem we have the relationship

$$\mathbf{g}_{k+1} - \mathbf{g}_k = \mathbf{A}(\mathbf{x}_{k+1} - \mathbf{x}_k) \tag{A.5.29}$$

Defining

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k \qquad \text{and} \qquad \mathbf{y}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \tag{A.5.30}$$

we see that equation A.5.29 becomes

$$\mathbf{y}_k = \mathbf{A}\mathbf{s}_k \tag{A.5.31}$$

It is therefore reasonable to demand that

$$\tilde{\mathbf{H}}_{k+1}\mathbf{y}_i = \mathbf{s}_i \qquad 1 \leq i \leq k \tag{A.5.32}$$

After n linearly independent steps we would then have $\tilde{\mathbf{H}}_{n+1} = \mathbf{A}^{-1}$. For $k < n$ there are an infinity of solutions for $\tilde{\mathbf{H}}_{k+1}$ satisfying equation A.5.32. A popular choice is the Broyden-Fletcher-Goldfarb-Shanno (or BFGS) update, given by

$$\tilde{\mathbf{H}}_{k+1} = \tilde{\mathbf{H}}_k + \left(1 + \frac{\mathbf{y}_k^\mathsf{T}\tilde{\mathbf{H}}_k\mathbf{y}_k}{\mathbf{y}_k^\mathsf{T}\mathbf{s}_k}\right)\frac{\mathbf{s}_k\mathbf{s}_k^\mathsf{T}}{\mathbf{s}_k^\mathsf{T}\mathbf{y}_k} - \frac{\mathbf{s}_k\mathbf{y}_k^\mathsf{T}\tilde{\mathbf{H}}_k + \tilde{\mathbf{H}}_k\mathbf{y}_k\mathbf{s}_k^\mathsf{T}}{\mathbf{s}_k^\mathsf{T}\mathbf{y}_k} \tag{A.5.33}$$

This is a rank-2 correction to $\tilde{\mathbf{H}}_k$ constructed from the vectors $\mathbf{s}_k$ and $\tilde{\mathbf{H}}_k\mathbf{y}_k$. The direction vectors, $\mathbf{p}_k = -\tilde{\mathbf{H}}_k\mathbf{g}_k$, produced by the algorithm obey

$$\mathbf{p}_i^\mathsf{T}\mathbf{A}\mathbf{p}_j = 0 \qquad 1 \leq i < j \leq k, \qquad \tilde{\mathbf{H}}_{k+1}\mathbf{A}\mathbf{p}_i = \mathbf{p}_i \qquad 1 \leq i \leq k \tag{A.5.34}$$

The storage requirements for Quasi Newton methods scale quadratically with the number of variables, and hence these methods tend to be used only for smaller problems. Limited memory BFGS reduces the storage by only using the l latest updates in computing the approximate Hessian inverse, equation (A.5.33). In contrast, the memory requirements for pure Conjugate Gradient methods scale only linearly with the dimension of $\mathbf{x}$. As before, although the algorithm was derived with a quadratic function in mind, the final form of the algorithm depends only on f and its gradients and may therefore be applied to non-quadratic functions f.

A.6 Constrained Optimisation using Lagrange Multipliers

Consider first the problem of minimising $f(\mathbf{x})$ subject to a single constraint $c(\mathbf{x}) = 0$. A formal treatment of this problem is beyond the scope of these notes and requires understanding the conditions under which the optimum can be found[48]. As an informal argument, however, imagine that we have already identified an $\mathbf{x}$ that satisfies the constraint, that is $c(\mathbf{x}) = 0$. How can we tell if this $\mathbf{x}$ minimises the function f? We are only allowed to search for lower function values around this $\mathbf{x}$ in directions which are consistent with the constraint. For a small change $\boldsymbol{\delta}$, the change in the constraint is,

$$c(\mathbf{x} + \boldsymbol{\delta}) \approx c(\mathbf{x}) + \boldsymbol{\delta} \cdot \nabla c(\mathbf{x}) \tag{A.6.1}$$

Let us also explore the change in f along a direction $\boldsymbol{\delta}$ where $\boldsymbol{\delta} \cdot \nabla c(\mathbf{x}) = 0$,

$$f(\mathbf{x} + \boldsymbol{\delta}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \cdot \boldsymbol{\delta}. \tag{A.6.2}$$

We are looking for a point $\mathbf{x}$ and direction $\boldsymbol{\delta}$ such that the change in f and c is minimal. This dual requirement can be represented as to find $\mathbf{x}$ and $\boldsymbol{\delta}$ that minimise

$$|\boldsymbol{\delta} \cdot \nabla f(\mathbf{x})|^2 + \gamma \, |\boldsymbol{\delta} \cdot \nabla c(\mathbf{x})|^2 \tag{A.6.3}$$

where $\gamma > 0$. This is a simple quadratic form and optimising for $\boldsymbol{\delta}$ gives that

$$\nabla f(\mathbf{x}) = -\gamma \frac{\boldsymbol{\delta} \cdot \nabla c(\mathbf{x})}{\boldsymbol{\delta} \cdot \nabla f(\mathbf{x})} \nabla c(\mathbf{x}) \tag{A.6.4}$$

Hence at the constrained optimum

$$\nabla f(\mathbf{x}) = \lambda \nabla c(\mathbf{x}) \tag{A.6.5}$$

for some scalar λ. We can formulate this requirement as to look for $\mathbf{x}$ and λ that constitute a stationary point of the *Lagrangian*

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda c(\mathbf{x}) \tag{A.6.6}$$

Differentiating with respect to $\mathbf{x}$, we obtain the requirement $\nabla f(\mathbf{x}) = \lambda \nabla c(\mathbf{x})$, and differentiating with respect to λ, we get that $c(\mathbf{x}) = 0$.

In the multiple constraint case $\{c_i(\mathbf{x}) = 0\}$ we find the stationary point of the Lagrangian

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x}). \tag{A.6.7}$$

A.6.1 Lagrange Dual

Consider the 'primal' problem

$$\text{minimise } f(\mathbf{x}) \;\; \text{subject to } c(\mathbf{x}) = 0 \tag{A.6.8}$$

The Lagrange dual is defined as

$$\mathcal{L}(\lambda) = \min_{\mathbf{x}} \left[f(\mathbf{x}) + \lambda c(\mathbf{x}) \right] \tag{A.6.9}$$

By construction, for any $\mathbf{x}$,

$$\mathcal{L}(\lambda) \leq f(\mathbf{x}) + \lambda c(\mathbf{x}) \tag{A.6.10}$$

Now consider the optimal $\mathbf{x}^*$ that solves the primal problem equation (A.6.8). Then

$$\mathcal{L}(\lambda) \leq f(\mathbf{x}^*) + \lambda c(\mathbf{x}^*) = f(\mathbf{x}^*) \tag{A.6.11}$$

where the last step follows since $\mathbf{x}^*$ solves the primal problem, meaning that $c(\mathbf{x}^*) = 0$. Since $\mathcal{L}(\lambda) \leq f(\mathbf{x}^*)$, the optimal λ is given by solving the unconstrained 'dual' problem

$$\max_{\lambda} \mathcal{L}(\lambda) \tag{A.6.12}$$

In addition to providing a bound that enables one to bracket the optimal primal value,

$$\mathcal{L}(\lambda) \leq f(\mathbf{x}^*) \leq f(\mathbf{x}), \qquad \text{for } c(\mathbf{x}) = 0 \tag{A.6.13}$$

the dual possesses the interesting property that it is concave, irrespective of whether f is convex. This follows since by construction the function $f(\mathbf{x}) + \lambda c(\mathbf{x})$ is concave in λ for any point $\mathbf{x}$. More explicitly, consider

$$\mathcal{L}(\lambda + \delta) = \min_{\mathbf{x}} [f(\mathbf{x}) + (\lambda + \delta)c(\mathbf{x})] \leq f(\mathbf{x}) + \lambda c(\mathbf{x}) + \delta c(\mathbf{x}) \tag{A.6.14}$$

Similarly,

$$\mathcal{L}(\lambda - \delta) \leq f(\mathbf{x}) + \lambda c(\mathbf{x}) - \delta c(\mathbf{x}) \tag{A.6.15}$$

Averaging equation (A.6.14) and equation (A.6.15), and taking the minimum over $\mathbf{x}$ of both sides, we have

$$\frac{1}{2}\left(\mathcal{L}(\lambda - \delta) + \mathcal{L}(\lambda + \delta)\right) \leq \mathcal{L}(\lambda) \tag{A.6.16}$$

which shows that $\mathcal{L}$ is concave.