Numerical Analysis

Kexing Ying

May 15, 2020

Contents

1 Introduction

This course is an introduction to numerical analysis and is built on top of last term's linear algebra on which many concepts will reappear. This time however, we will mostly work in specific spaces rather than arbitrary inner product spaces. We will also consider issues of implementing algorithms and their ability to scale to large problems. This will be achieved by examining their efficiency, accuracy and stability. We will also consider typical numerical concepts such as iterations, conditioning, error analysis and operations count.

As a outline, we will first delve into numerical linear algebra, in which we will study orthogonalisation, least-squares problems, linear equations and factorisations. We will the move on to gradients and hessians, interpolations with orthogonal and non-orthogonal polynomials, Fourier series and lastly, numerical integration.

We will in this course mostly deal with the vector space $\mathbb{R}^n$ while unless mentioned otherwise, algorithms can easily extended to $\mathbb{C}^n$. We will use inner product and dot product interchangeably and define the outer product (tensor product) between two vectors $\mathbf{a}, \mathbf{b}$ to be the matrix $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = A \in M_n(\mathbb{R})$ where $A_{ij} = a_i b_j$. We also note that the dot product on the reals form a bilinear form, and so all associated properties apply.

Lastly, we define the i-th standard basis $e_i \in \mathbb{R}^n$ as the vector with the j-th entry equal to the ij-Kronecker delta. That is, $[e_i]_j = 1$ is $i = j$ and 0 otherwise. This is a nice set of basis as it is orthonormal and given some vector $\mathbf{v}$, the dot product $\langle e_i, \mathbf{v} \rangle$ is the i-th entry of $\mathbf{v}$.

2 Numerical Linear Algebra

2.1 Gram-Schmidt and QR-Decomposition I

We would like to find an orthogonal basis from a set of linearly independent vectors. As, simply by normalising the resulting vectors, we also obtain an orthonormal basis. From first year, we know that this can be achieved through the Gram-Schmidt process while we will also take a look at another method which utilises the Householder transformation. In both cases, the methods are related to the QR-decomposition of a square matrix.

Suppose we have a set of n linearly independent vectors $\{a_k\}_{k=1}^n$ in the m-dimensional space $\mathbb{R}^m$ where $n \leq m$. It is often advantageous to convert this set of vectors into an orthonormal basis such that the span of this basis is the same as the span of $\{a_k\}_{k=1}^n$.

To achieve this, we propose the first procedural method – the *classical Gram-Schmidt* (cGS) procedure.

Algorithm 1 (Classical Gram-Schmidt Procedure)**.** Given a set of n linearly independent vectors $\{\mathbf{a}_k\}_{k=1}^n$ in the m-dimensional space $\mathbb{R}^m$, we obtain an orthonormal basis of $\text{sp}\{\mathbf{a}_k\}_{k=1}^n$.

1. Let $\mathbf{v}_1 := \mathbf{a}_1; \mathbf{q}_1 := \mathbf{v}_1/\|\mathbf{v}_1\|$. We call $\mathbf{v}_1$ the preliminary vector.
2. For $k = 2, \cdots, n$, let $\mathbf{v}_k := \mathbf{a}_k - \sum_{l=1}^{k-1} \langle \mathbf{a}_k, \mathbf{q}_l \rangle \mathbf{q}_l$; $\mathbf{q}_k := \mathbf{v}_k/\|\mathbf{v}_k\|$, that is, we define $\mathbf{v}_k$ such that it is orthogonal to all previous $\mathbf{q}_l$ while we normalise $\mathbf{v}_k$ resulting in $\mathbf{q}_k$.

Then, by the above procedure $\{\mathbf{q}_k\}_{k=1}^n$ is the required set of vectors.

Let us recall the proof for the correctness of the classical Gram-Schmidt procedure.

Proof. Normality and span (as $\mathbf{q}_k$ is a linear combination of $\mathbf{a}_i$ where $i \leq k$) is trivial so we shall show orthogonality.

We induction on n. For $n = 1$, orthogonality is trivial so let us assume the inductive hypothesis and suppose $n = i+1$. By the inductive hypothesis, the Gram-Schmidt procedure will result us with $\{\mathbf{q}_k\}_{k=1}^i$ – an orthonormal basis of $\text{sp}\{\mathbf{a}_k\}_{k=1}^i$, and so, it suffices to show,

$$\langle \mathbf{q}_j, \mathbf{v}_{i+1} \rangle = \left\langle \mathbf{q}_j, \mathbf{a}_{i+1} - \sum_{l=1}^{i} \langle \mathbf{a}_{i+1}, \mathbf{q}_l \rangle \mathbf{q}_l \right\rangle = 0,$$

for all $j = 1, \cdots, i$. But, this is true as,

$$\left\langle \mathbf{q}_j, \mathbf{a}_{i+1} - \sum_{l=1}^{i} \langle \mathbf{a}_{i+1}, \mathbf{q}_l \rangle \mathbf{q}_l \right\rangle = \langle \mathbf{q}_j, \mathbf{a}_{i+1} \rangle - \sum_{l=1}^{i} \langle \mathbf{a}_{i+1}, \mathbf{q}_l \rangle \langle \mathbf{q}_j, \mathbf{q}_l \rangle = \langle \mathbf{q}_j, \mathbf{a}_{i+1} \rangle - \langle \mathbf{q}_j, \mathbf{a}_{i+1} \rangle = 0,$$

where the second equality holds since by the inductive hypothesis $\langle q_j, q_l \rangle = \delta_{jl}$. □

While we have proved the correctness of the algorithm, the question remains on whether or not we can always perform such an procedure. We see that, by induction, to see that the algorithm can always be performed, it suffices to show that there does not exists a case where $\mathbf{v}_2 \neq 0$ (we can then apply induction).

Suppose $\mathbf{v}_2 = 0$, then $\mathbf{a}_2 - \langle \mathbf{a}_2, \mathbf{q}_1 \rangle \mathbf{q_1} = 0$. But this would mean $\mathbf{a}_2$ is a multiple of $\mathbf{q}_1$ which is in turn a multiple of $\mathbf{a}_1$, contradicting the linearly independent assumption. So, this cannot occur and so cGS can always be performed.

While the cGS is mathematically correct, when implementing the algorithm on computers, it is possible to find special cases where the cGS suffers from accuracy and stability. As we shall see on in an exercise, it is possible to construct a slightly different version of the Gram-Schmidt procedure – *modified Gram-Schmdit* (mGS) such that this is no longer a problem.

The Gram-Schmidt procedure can be use to decompose a matrix into a product of an orthogonal matrix and a upper-triangular matrix. This process is refereed to as QR-decomposition. The QR-decomposition is a very important decomposition in numerical analysis and we shall, in fact, look at two methods of achieving the QR-decomposition, and hence the name of this section. The QR-decomposition decomposes a matrix into the product of two matrices Q and R where Q is orthogonal and R is upper triangular.

Suppose we have the linearly independent sequence of vectors $\{\mathbf{a}_k\}_{k=1}^n$ and the orthonormal basis of this resulted from Gram-Schmidt $\{\mathbf{q}_k\}_{k=1}^n$ in $\mathbb{R}^m$. Then by defining

$$A := (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n) \in \mathbb{R}^{m\times n},$$

and similarly,

$$Q := (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_n) \in \mathbb{R}^{m\times n},$$

we seek to establish the relation $R \in \mathbb{R}^{n\times n}$ such that $A = QR$. Indeed, by considering the classical Gram-Schmidt procedure, where

$$\mathbf{v}_k := \mathbf{a}_k - \sum_{l=1}^{k-1} \langle \mathbf{a}_k, \mathbf{q}_l\rangle \mathbf{q}_l,$$

we see that $\mathbf{a}_j$ is a linear combination of $\mathbf{q}_i$ where $j \leq i$, and so it follows that R is upper triangular.

Suppose we denote the ij-th entry of R as r_{ij}, then $\mathbf{a}_k = \sum_{l=1}^k r_{lk}\mathbf{q}_l$, by the definition of matrix multiplication. Since $\mathbf{q}_1 = \mathbf{a}_1/\|\mathbf{a}_1\|$, we have $\mathbf{a}_1 = \|\mathbf{a}_1\|\mathbf{q}_1$, and so, $r_{11} = \|\mathbf{a}_1\|$. Similarly, for $\mathbf{a}_k$, we have

$$\mathbf{a}_k := \mathbf{v}_k + \sum_{l=1}^{k-1} \langle \mathbf{a}_k, \mathbf{q}_l\rangle \mathbf{q}_l,$$

where $\mathbf{v}_k = \|\mathbf{v}_k\|\mathbf{q}_k$, so,

$$\mathbf{a}_k := \|\mathbf{v}_k\|\mathbf{q}_k + \sum_{l=1}^{k-1} \langle \mathbf{a}_k, \mathbf{q}_l\rangle \mathbf{q}_l.$$

Thus, by comparing coefficients, we find $\|\mathbf{v}_k\| = r_{kk}$ and $\langle \mathbf{a}_k, \mathbf{q}_l\rangle = r_{lk}$ for $1 \leq l < k$. With this, using Gram-Schmdit, we have found a method to decompose a matrix as a product of an orthogonal and an upper triangular matrix.

However, the question of why this decomposition is importan remains. Suppose we would like to solve the linear system $A\mathbf{x} = \mathbf{b}$ (where A has full rank). If we have a QR-decomposition on A, say $A = QR$, then the problem becomes $QR\mathbf{x} = \mathbf{b}$. As Q is orthogonal, $Q^TQ = I$ and so,

$$\mathbf{d} := Q^T\mathbf{b} = Q^TQR\mathbf{x} = R\mathbf{x}.$$

Now, as R is upper triangular, the linear system $R\mathbf{x} = \mathbf{d}$ becomes easy to solve by **backwards substitution**; that is, since R is upper triangular, we have

$$x_n = d_n / r_{nn},$$

and

$$x_k = \frac{1}{r_{kk}} \left(d_k - \sum_{i=k+1}^{n} r_{ki} x_i \right).$$

We note that we claimed Q is orthogonal throughout the argument. This is true as the column vectors are orthonormal, and hence, by the definition of matrix multiplication, we have

$$[Q^T Q]_{ij} = \langle \mathbf{q}_i, \mathbf{q}_j \rangle = \delta_{ij},$$

and so $Q^T Q = I$.

Indeed, the orthogonal matrices are a nice set of matrices and and the dot product and hence the norm is invariant under transformations by orthogonal matrices. Indeed,

$$\langle Q\mathbf{x}, Q\mathbf{y} \rangle = \mathbf{x}^T Q^T Q \mathbf{y} = \mathbf{x}^T \mathbf{y} = \langle \mathbf{x}, \mathbf{y} \rangle.$$

By thinking in Euclidean spaces, we see that these types of transformations are rotations and so, the transformations induced by an orthogonal matrix is often refereed as a rotation.

2.2 Projectors and QR-decomposition II

Having established one algorithm for computing the QR-decomposition, we will now design a second and complementary method to accomplish the same task. The second method will depend on *projectors* and *reflectors* so we will discuss them first.

Definition 2.1 (Projector)**.** A matrix $P \in M_n(\mathbb{R})$ is a projector if

$$P^2 = P.$$

A projector matrix is refereed to as being idempotent.

Given a projector P, the matrix $I - P$ is also a projector. Indeed,

$$(I - P)^2 = I^2 - 2P + P^2 = 1 - P.$$

The matrix $I - P$ is refereed to as the *complementary projector* to P.

Straight away, we see that if a projector has full rank, than $P^2 = P \implies P^{-1}P^2 = P^{-1}P \implies P = I$, resulting in the trivial projector with the complementary projector the zero map $\mathbf{0}$. So, assuming a projector is non-trivial, then as P maps the vector space onto $\mathrm{Im}P$, we may interpret geometrically that P *projects* vectors onto its image.

Definition 2.2 (Orthogonal Projector)**.** A projector P is called an orthogonal projector if $P^T = P$.

As the name might suggest, if P is an orthogonal projector, then P projects vectors onto its image orthogonally. Indeed, if $\langle \cdot, \cdot \rangle$ is the dot product, then

$$\langle P\mathbf{v}, P\mathbf{v} - \mathbf{v} \rangle = \langle P\mathbf{v}, P\mathbf{v} \rangle - \langle P\mathbf{v}, \mathbf{v} \rangle = \mathbf{v}^T P^T P\mathbf{v} - \mathbf{v}^T P^T \mathbf{v} = \mathbf{v}^T P^2 \mathbf{v} - \mathbf{v}^T P\mathbf{v} = 0.$$

Furthermore, for an orthogonal projector P, its image and the image of its complementary projector are perpendicular. Say, if $\mathbf{x}, \mathbf{y}$ are vectors, then

$$\langle P\mathbf{x}, (I - P)\mathbf{y} \rangle = \mathbf{x}^T P^T (I - P)\mathbf{y} = \mathbf{x}^T (P - P^2)\mathbf{y} = 0.$$

There are many methods to construct orthogonal projectors. One simple method is to realise that, for all normalised vector $\mathbf{q}$, the outer product of $\mathbf{q}$ with itself is in fact an orthogonal projector. To see this, we have, for all $\mathbf{v}$, if $P = \mathbf{q} \otimes \mathbf{q}$,

$$P^2 = \mathbf{q}\mathbf{q}^T \mathbf{q}\mathbf{q}^T = \mathbf{q}\langle \mathbf{q}, \mathbf{q} \rangle \mathbf{q}^T = \mathbf{q}\mathbf{q}^T = P,$$

and

$$P\mathbf{v} = \mathbf{q}\mathbf{q}^T \mathbf{v} = \langle \mathbf{v}, \mathbf{q} \rangle \mathbf{q}.$$

Using this concept of projectors, we can find an different method for computing the QR-decomposition of an arbitrary matrix. Specifically, the method uses the *Householder reflectors*. As a general outline, this second method achieves the QR-decomposition by sequentially applying orthogonal matrices onto A, eventually resulting in a upper triangular matrix.

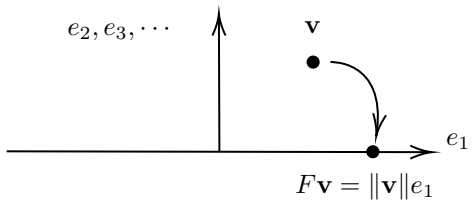
$$A = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} \xrightarrow{Q_1} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{pmatrix} \xrightarrow{Q_2} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{pmatrix} \xrightarrow{Q_3} \begin{pmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{pmatrix} = R$$

With this in mind, we see that the orthogonal matrices Q_k does not change the first $k-1$ rows, and so $Q_k = I_{k-1} \oplus F_{n-k+1}$ where F_{n-k+1} introduces zeros in the lower $n-k$ rows and are refereed to as the *Householder reflectors*.

Since the Householder reflectors are constrained by the fact that they are orthogonal, they must preserve the lengths of the vectors it acts on. So, as we require the Householder reflectors to introduce zeros, the vectors it acted on is therefore in the form

$$F\mathbf{x} = F(\times, \times, \times, \cdots)^T = (\|\mathbf{x}\|, 0, 0, \cdots)^T.$$

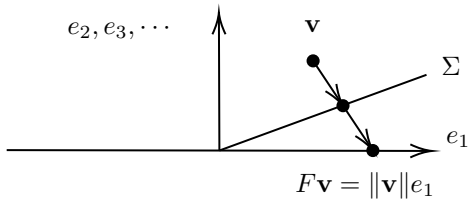
Geometrically consider the following diagram where we have labelled the first basis on one axis and the remaining bases on the other. We would like to find a mapping F such that $\mathbf{v}$ is mapped onto the horizontal axis. Since F needs to be orthogonal, the natural candidate for F is a rotation. To achieve this, we may construct a hyperplane Σ, between $\mathbf{v}$ and $F\mathbf{v}$, and by reflecting $\mathbf{v}$ by Σ, the transformation is equivalent to a single rotation.



Suppose we define $\mathbf{w} = \|\mathbf{v}\|e_1 - \mathbf{v}$, then, we have $\mathbf{w}$ is orthogonal to the hyperplane Σ. With that in mind, we see that the image of the projector defined by

$$P' = \frac{\mathbf{w}\mathbf{w}^T}{\mathbf{w}^T\mathbf{w}}$$

is the plane orthogonal to Σ. Hence, the projector to Σ is simply the complementary projector to P', $P = I - P'$.

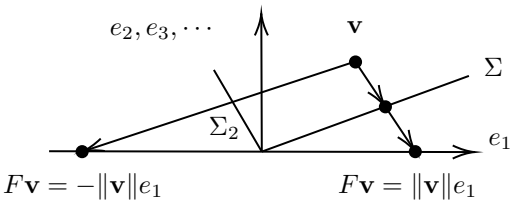


However, we do not wish to project into Σ but to reflect around it. Thus, to reflect around Σ we projecting twice as the distance in the same direction. This results in the desired Householder reflector,

$$F = I - 2\frac{\mathbf{w}\mathbf{w}^T}{\mathbf{w}^T\mathbf{w}},$$

where $\mathbf{w} = \|\mathbf{v}\|e_1 - \mathbf{v}$.

We note that this is not the only possible reflection as we may reflect around the hyperplane Σ_2 as demonstrated in the figure below.



Mathematically, the choice does not matter as both transformations results in the vector with only components in e_1, however, as often times in numerical analysis, we need to consider the computers in computing these values. Indeed, by considering that if we reflect by a plane which is almost parallel to the e_1-axis, the computer is required to work with large numbers and so, could possible encounter cancellation errors. Thus, it is usually better to choose the shallow plane to reflect around.

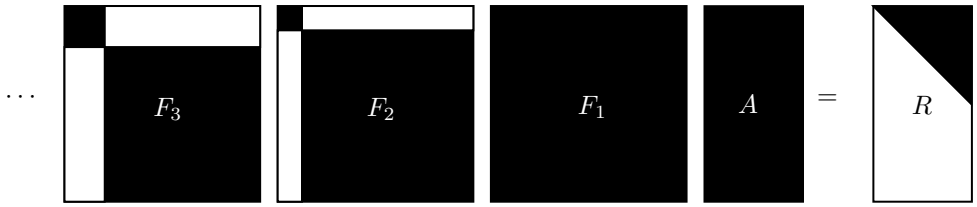
With that, we arrive at the Householder-based QR-decomposition.

Algorithm 2 (Householder-based QR-decomposition)**.** If $A \in M_{m\times n}(\mathbb{R})$ be the matrix we are decomposing, then, for $k = 1, \cdots n$, we define

- $\mathbf{x} := A_{k:m,k}$ that is, the k to m-th entry of the k-th row of A;
- $\mathbf{v}_k := \text{sign}(x_1)\|\mathbf{x}\|e_1 + \mathbf{x}$;
- $\mathbf{v}_k := \mathbf{v}_k/\|\mathbf{v}_k\|$;

- $A_{k:m,k:n} := A_{k:m,k:n} - 2\mathbf{v}_k(\mathbf{v}_k^T A_{k:m,k:n})$.

Thus, by successively applying the Householder reflection, we achieve the QR-decomposition.



By recalling the Gram-Schmdit QR-decomposition where we can consider the algorithm as sequentially applying upper triangular matrices to A until it becomes orthogonal, so,

$$AR_1R_2\cdots R_n = Q,$$

and thus, $\prod R_i = R^{-1}$. With that, we can describe the Gram-Schmidt process as a *triangular orthogonalisation* while, on the other hand, the Householder algorithm is a *orthogonal triangularisation*. Indeed, the two process are not equivalent and consequentially, the resulting QR-decompositions are different. We notice that the Gram-Schmidt procedure results in R being square while Q rectangular, that is, it yields the *reduced form* of the QR-decomposition. The Householder decomposition, on the other hand, results in Q being square while R being rectangular. This is refereed as the *full form* of the QR-decomposition and by noticing that the additional parts of Q are multiplied by 0, we have that the full form provides us with information about the null-space of A.

2.3 Least Square Problems

Least square problems are often found in data fitting, statistics, computational modelling and many other fields, e.g. minimising the mean square error for an estimator. Least square problems are also significant historically as they constitute one of the earliest problems formulated in matrix form by Gauss.

A least square problem aims at the solution of an overdetermined system of equations, that is, we have the linear system of the form

$$A\mathbf{x} = \mathbf{b},$$

where $A \in \mathbb{R}^{m\times n}$ for some $m > n$.

In general, we do not have a solution to this problem, however, we may minimise the 2-norm of the residual given by

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}.$$

Straight away, we see that, since we have more equations than unknowns, there exists a solution to our linear system of equations if and only if $\mathbf{b} \in \mathrm{Im}A$ and in this case, the residual $\mathbf{r} = 0$. On the other hand, if $\mathbf{b} \notin \mathrm{Im}A$, then we are tasked with finding $\mathbf{x} \in \mathbb{R}^n$ such that

$$\mathbf{x} = \mathrm{argmin}_{\mathbf{x}}\|\mathbf{b} - A\mathbf{x}\|_2,$$

where we uses the 2-norm as its derivative yields a linear system, and so, can be solved using standard linear algebra techniques.

Geometrically, we see that the residual vector $\mathbf{r}$ is orthogonal to the image of A whenever it is minimized, so we have the restriction that, for all $\mathbf{v} \in \mathbb{R}^n$,

$$0 = \langle A\mathbf{v}, \mathbf{r} \rangle = \mathbf{v}^T A^T \mathbf{r},$$

and so, $A^T\mathbf{r} = 0$. Indeed, by considering that, for all $\mathbf{x} \in \mathbb{R}^n$, $A\mathbf{x}$ is the orthogonal projection P of $\mathbf{b}$ onto the image of A, we have $A\mathbf{x} = P\mathbf{b}$. So, the least square problem is rephrased such that we need to find such a projection P.

We shall look at two methods for finding such a projection the first of which follows straight away when we impose the orthogonality condition. As we require the residual to be orthogonal to the image, we have

$$A^T(\mathbf{b} - A\mathbf{x}) = 0$$

and so,

$$A^T A\mathbf{x} = A^T\mathbf{b}.$$

This equation is known as the normal equation, and with this, we see that, since $A^T A$ is nonsingular if and only if A has full rank, the least square problem has a unique solution $\mathbf{x}$ if and only if A has full rank. Thus, if A has full rank, we have

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b},$$

and in some sense, the triple product $(A^T A)^{-1} A^T$ is a kind of (pseudo)-inverse.

Definition 2.3 (Pseudo-inverse)**.** Given a matrix $A \in \mathbb{R}^{n \times m}$, if A has full rank then, the pseudo-inverse of A is

$$A^+ = (A^T A)^{-1} A^T.$$

This naive approach to the least square problem can produce ill-conditioned matrices and is expensive since it required the computation of a inverse and therefore is rarely used to solve least square problems. A better approach is to use the QR-decomposition. Rather than orthogonalising the residual onto the image of A, we instead form the projector directly.

Suppose we have decomposed the matrix A in to $A = QR$, and let us define $P = QQ^T$. Then, by considering $P\mathbf{b} = QQ^T\mathbf{b}$, we have

$$QR\mathbf{x} = QQ^T\mathbf{b},$$

and so, by multiplying both side by Q^T, by the orthogonality of Q, we have $R\mathbf{x} = Q^T\mathbf{b}$. Thus, by backwards substitution, we may find $\mathbf{x}$ given R is full rank (which is true if and only if A is full rank). Furthermore, by comparing the two methods, since the solution is unique (provided A if full rank), we find

$$A^+ = R^{-1}Q^T,$$

and $\mathbf{x} = A^+\mathbf{b}$.

We note that in the second method above, we have used the reduced form of the QR-decomposition. Using the full form instead, we may obtain more information about the

residual of the least square problem. Suppose we decompose $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ and $R \in \mathbb{R}^{m\times n}$. As before, we project $\mathbf{b}$ onto the image of A by Q^T. However, since the full form also represents the kernel of A, we can decompose $Q^T\mathbf{b}$,

$$Q^T\mathbf{b} = \alpha + \beta = \begin{pmatrix} (Q^T\mathbf{b})_1 \\ \vdots \\ (Q^T\mathbf{b})_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ (Q^T\mathbf{b})_{n+1} \\ \vdots \\ (Q^T\mathbf{b})_m \end{pmatrix},$$

where $Q^T\beta \in \ker A$ and $Q^T\alpha \in \text{Im } A$. Consider the case when $\beta = 0$. Indeed, if $\beta = 0$ then we have,

$$R\mathbf{x} = Q^TQR\mathbf{x} = Q^TA\mathbf{x} = Q^T\mathbf{b} = \alpha.$$

So, $0 = R\mathbf{x} - Q^T\mathbf{b}$ and hence,

$$0 = QR\mathbf{x} - QQ^T\mathbf{b} = A\mathbf{x} - \mathbf{b}.$$

We note that in the full QR-decomposition, Q is a square orthogonal matrix and so $QQ^T = I$. Now, if $\beta \neq 0$, we have $R\mathbf{x} = Q^T\mathbf{b}$ is a inconsistent system. Since $Q^T\alpha \in \text{Im } A$, we have $\alpha \in \text{Im } R$ and so, there exists some $\mathbf{x}^* \in \mathbb{R}^n$ such that $R\mathbf{x}^* = \alpha$. This is in fact the solution to the least square problem since

$$\begin{aligned} \|A\mathbf{x} - b\|^2 &= \|Q^T(A\mathbf{x} - b)\|^2 \\ &= \|R\mathbf{x} - (\alpha + \beta)\|^2 \\ &= \langle (R\mathbf{x} - \alpha) - \beta, (R\mathbf{x} - \alpha) - \beta\rangle \\ &= \|R\mathbf{x} - \alpha\|^2 - 2\langle R\mathbf{x} - \alpha, \beta\rangle + \|\beta\|^2. \end{aligned}$$

Thus, by considering $\langle R\mathbf{x}-\alpha, \beta\rangle = \langle R(\mathbf{x}-\mathbf{x}^*), \beta\rangle = 0$ we have $\|A\mathbf{x}-b\|^2 = \|R\mathbf{x}-\alpha\|^2+\|\beta\|^2 = \|\beta\|^2$ whenever $x = x^*$. So, x^* is the solution and $\|\beta\|$ is the norm of the residual.

2.4 LU-Decomposition

As we have seen, the QR-decomposition provides us with a method for solving a system of equation with more equations than unknowns and this decomposition can also be used when ever we would like to solve a system with equal number of equations as unknowns. However, this problem can be solved more efficiently through the LU-decomposition.

Similar to the QR-decomposition, the LU-decompostion shall manipulate the matrix column by column in order to reduce the matrix to upper-triangular form. In contrast to the Householder QR-decomposition however, we shall achieve this through a sequence of lower-triangular matrices, i.e.

$$L_{m-1}\cdots L_2L_1A = U,$$

where $A \in \mathbb{R}^{m\times n}$, U upper-triangular and $(\prod L_i)^{-1}$ lower-triangular.

This decomposition is helpful in solving the linear equation $A\mathbf{x} = \mathbf{b}$ since if $A = LU$, we have $\mathbf{b} = L(U\mathbf{x}) = L\mathbf{y}$. Thus, by using forward substitution, we can solve $\mathbf{y}$ resulting in the system $U\mathbf{x} = \mathbf{y}$ which can be solved using backwards substitution.

Consider for each step, we require the operator L_k to turn a general vector into a vector with 0 below the k-th entries, i.e.

$$\begin{pmatrix} x_{1,k} \\ \vdots \\ x_{k,k} \\ x_{k+1,k} \\ \vdots \\ x_{m,k} \end{pmatrix} \xrightarrow{L_k} \begin{pmatrix} x_{1,k} \\ \vdots \\ x_{k,k} \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

and so, we see that

$$L_k = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{m,k} & & & 1 \end{pmatrix},$$

where $l_{j,k} = x_{j,k}/x_{k,k}$ for $k < j \leq m$. However, this algorithm as it stands is numerically unstable. Indeed, if there exists a 0 on the diagonal then the algorithm will need to divide by zero! To combat this, we will introduce an additional step called *partial pivoting*.

Consider instead, when acting on the k-th column of A, we denote i for the entry of $A_{k:m,k}$ with maximum absolute value. Then, by multiplying by a permutation matrix P, we may swap the i-th entry with the k-th entry of $A_{k:m,k}$ and we may proceed with the standard LU-step. Thus, the full decomposition becomes

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1A = U.$$

Now, (as seen on the problem sheet) we have,

$$L_{m-1}P_{m-1}\cdots L_2P_2L_1P_1 = L'_{m-1}\cdots L'_2L'_1P_{m-1}\cdots P_2P_1,$$

where L'_i are also lower-triangular matrices. Thus, this modified procedure results in the linear system

$$PA = LU,$$

where $P = \prod P_i$ is a permutation matrix. In other words, to achieve the LU-decomposition with partial pivoting all we have to do is to permute the rows of A before hand such that for each k, the maximum entry of $A_{k:m,k}$ is swapped with $A_{k,k}$ and then proceeding as usual.

2.5 Cholesky Decomposition

The Cholesky decomposition is a decomposition that applies to a narrow but important type of matrices – symmetric, positive definite matrices.

Definition 2.4 (Symmetric, Positive Definite Matrix)**.** A matrix A is symmetric and positive definite if $A^T = A$ and for all $\mathbf{x}$, $\mathbf{x}^T A\mathbf{x} > 0$.

Equivalently, we see that symmetric positive definite matrices are matrices whose induced bilinear form $\langle \cdot, \cdot \rangle_A : (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{x}^T A \mathbf{y}$ is symmetric and positive definite.

Symmetric positive definite matrices arise in many applications such as covariance matrices in statistics, as kernels in machine learning and as diffusion tensors in medical imaging and weighted norms, etc.

The Cholesky decomposition can be though of as a symmetric LU-decomposition, applying a LU-step from the left and right while maintaining the symmetry of the resulting matrix. Starting with a symmetric, positive definite matrix A and we write

$$A = \begin{pmatrix} a_{11} & \mathbf{w}^T \\ \mathbf{w} & K \end{pmatrix}$$

assuming $a_{11} > 0$. Then, by defining

$$R_1^T := \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \mathbf{w}/\sqrt{a_{11}} & I \end{pmatrix}, \quad A^{(1)} := \begin{pmatrix} 1 & 0 \\ 0 & K - \mathbf{w}\mathbf{w}^T/a_{11} \end{pmatrix}$$

by multiplying $A^{(1)}$ with R_1^T and R_1 on each side, we obtain

$$R_1^T A^{(1)} R_1 = \begin{pmatrix} \sqrt{a_{11}} & 0 \\ \mathbf{w}/\sqrt{a_{11}} & I \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & K - \mathbf{w}\mathbf{w}^T/a_{11} \end{pmatrix} \begin{pmatrix} \sqrt{a_{11}} & \mathbf{w}^T/\sqrt{a_{11}} \\ 0 & I \end{pmatrix} = \begin{pmatrix} a_{11} & \mathbf{w}^T \\ \mathbf{w} & K \end{pmatrix} = A.$$

Now, by induction, we find that $K - \mathbf{w}\mathbf{w}^T/a_{11}$ is also a symmetric, positive definite matrix, and so we can find R_2' and $A^{(2)*}$, such that $R_2'^T A^{(2)*} R_2' = K - \mathbf{w}\mathbf{w}^T/a_{11}$ and so, by letting $R_2 = 1 \oplus R_2'$ and $A^{(2)} = 1 \oplus A^{(2)*}$, we have $R_2^T A^{(2)} R^2 = A^{(1)}$. Hence, by repeatedly applying the process, we have

$$A = R_1^T R_2^T \cdots R_m^T I R_m \cdots R_2 R_1.$$

So, by letting $R := R_m \cdots R_2 R_1$, we have $A = R^T R$ where R is upper triangular.

Note that we assumed the upper-left element of each submatrix is positive. This must be true for a positive definite matrix since if $[A]_{11} = a_{11} < 0$, then $e_1^T A e_1 = a_{11} < 0$; contradicting positive definiteness.

Similar to the LU-decomposition, the Cholesky decomposition can be used to solve linear systems $A\mathbf{x} = \mathbf{b}$ where A is symmetric, positive definite. Indeed, if $A = R^T R$, then $R^T R\mathbf{x} = \mathbf{b}$ and so, we may solve for $R\mathbf{x}$ using a forward substitution, and then we may solve for $\mathbf{x}$ with a backwards substitution.

3 Polynomial Interpolations

Interpolation is a numerical technique to formulate a continuous representation based on discrete data points. This is applied in many fields especially data science. Interpolation requires a basis of functions that are tailored to the data points and we will concentrate on the set of polynomials. This is because polynomials are particularly practical as they can be easily differentiated and integrated, and hence provide us with the foundations for numerically approximating differential and integral expressions.

Definition 3.1 (Polynomial Interpolation)**.** Given the data points $(z_j, f_j)_{j=0}^n$ where $z_i \neq z_j$ for all $i \neq j$, the polynomial $p_n \in \mathbb{P}^n$ such that for all $j = 0, \cdots, n$, $p_n(z_j) = f_j$ is called the interpolating polynomial (or interpolant) of the data set.

3.1 Lagrange Interpolation

It is not always clear that we can always find such an interpolating polynomial for any given data sets so we shall present a proof the the existence by constructing the interpolant.

Proposition 1. Given the data points $(z_j, f_j)_{j=0}^n$ where $z_i \neq z_j$ for all $i \neq j$, there always exists a polynomial $p_n \in \mathbb{P}^n$ such that for all $j = 0, \cdots, n$, $p_n(z_j) = f_j$.

Proof. Define $l_j(z) = \prod_{k=0;k\neq j}^n \frac{z-z_k}{z_j-z_k}$ for all $j = 0, \cdots, n$. Then, we see $l_j \in \mathbb{P}^n$ and $l_j(z_r) = \delta_{jr}$ for all $j, r = 0, \cdots, n$. So, by defining

$$p_n(z) = \sum_{j=0}^{n} f_j l_j(z),$$

we see $p_n(z_k) = \sum_{j=0}^n f_j l_j(z_k) = \sum_{j=0}^n f_j \delta_{jk} = f_k$ and so p_n is a interpolant of our data set and we call the set of polynomials

$$L := \{l_j \mid j = 0, \cdots, n\},$$

the Lagrange basis functions. □

Furthermore, it turns out that our construction is unique.

Proposition 2. Given the data points $(z_j, f_j)_{j=0}^n$ where $z_i \neq z_j$ for all $i \neq j$, there always exists a **unique** polynomial $p_n \in \mathbb{P}^n$ such that for all $j = 0, \cdots, n$, $p_n(z_j) = f_j$.

Proof. As we have already proven existence, it suffices to show uniqueness. Suppose $p, q \in \mathbb{P}^n$ such that $p(z_j) = q(z_j) = f_j$ for all $j = 0, \cdots, n$. Then, $(p-q)(z_j) = p(z_j) - q(z_j) = 0$ for all j, that is z_j are the roots of $p - q$. However, as p, q have degree n, $p - q$ can at most have degree n, so if $p - q \neq 0$, then $p - q$ has at most n roots. This contradicts as we found $n + 1$ roots, and thus $p - q = 0$. □

However, while the construction of the interpolant through the Lagrange basis is correct, it is not necessarily nice to compute since we are required to multiple and add many terms.

Instead, we ask if it is possible to solve the system directly for the coefficients, i.e. can we find a_k directly such that

$$p_n(z_j) = \sum_{k=0}^{n} a_k z_j^k = f_j$$

for all $j = 0, \cdots, n$.

By writing the question as a linear system of $n+1$ equations, we see that the answer becomes obvious –

$$\begin{pmatrix} 1 & z_0 & z_0^2 & \cdots & z_0^n \\ 1 & z_1 & z_1^2 & \cdots & z_1^n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & z_n & z_n^2 & \cdots & z_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix}.$$

By observing that the first matrix is simply the Vandemonde matrix V, and by assumption, we have $z_i \neq z_j$ for all $i \neq j$, V is invertible and hence,

$$\begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = V^{-1} \begin{pmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \end{pmatrix},$$

solving our interpolation. Indeed, this also provide us with uniqueness as the inverse is unique and all interpolants satisfies this equation.

While this is mathematically correct, inverting the Vandemonde matrix becomes difficult as n increases and so, this method is ill-conditioned.

3.2 Divided Differences

Alternative to the Lagrange interpolation, the divided differences method provide us with a method to find the interpolant of a data set recursively.

Suppose we have found the interpolant $p_{n-1} \in \mathbb{P}^{n-1}$ of the data set $(z_j, f_k)_{j=0}^{n-1}$ and we are asked to find the interpolant of $(z_j, f_k)_{j=0}^{n}$, given the new data point (z_n, f_n). If we are using the Lagrange interpolation method, we are required to compute the entire Lagrange basis again, and so, we look for an alternative construction that will make use of p_{n-1}.

Suppose there exists some $q_n \in \mathbb{P}^n$, $q_n(z_j) = 0$ for all $j = 0, \cdots, n-1$ such that $p_n = p_{n-1} + q_n$. Since, be construction q_n has roots z_j for all $j = 0, \cdots, n-1$,

$$q_n(z) = C \prod_{k=0}^{n-1} (z - z_k)$$

for some C. So,

$$p_n = p_{n-1} + C \prod_{k=0}^{n-1} (z - z_k).$$

Evaluating both sides at z_n, we have

$$C = \frac{f_n - p_{n-1}(z_n)}{\prod_{k=0}^{n}(z_n - z_k)},$$

and so,

$$p_n(z) = p_{n-1}(z) + (f_n - p_{n-1}(z_n)) \prod_{k=0}^{n} \frac{z - z_k}{z_n - z_k}.$$

Classically, the following notation for C is used

$$f[z_0, z_1, \cdots, z_n] := C$$

and is called the *divided difference* of order n.

We note that $f[z_j] = f_j$ and $f[z_{\pi_0}, z_{\pi_1}, \cdots, z_{\pi_n}] = f[z_0, z_1, \cdots, z_n]$, and for any data set, we have

$$f[z_0, \cdots, z_n] = \sum_{j=0}^{n} \frac{f_j}{\prod_{k=0;k\neq j}^{n}(z_k - z_k)}$$

The divided differences method produce a hierarchy of interpolating polynomials, and with recursion, this method is easy to implement and results in the interpolant

$$p_n(z) = \sum_{j=0}^{j} f[z_0, \cdots, z_j] \prod_{k=0}^{j-1}(z - z_k)$$

and is refereed to as the *Newton form* of the interpolating polynomial.

However, while this form is rather nice, it is not necessarily obvious the speed at which we can generate the divided differences. It turns out that for distinct $z_0, \cdots, z_{n+1}$, we have

$$f[z_0, \cdots, z_{n+1}] = \frac{f[z_0, \cdots, z_n] - f[z_1, \cdots, z_{n+1}]}{z_0 - z_{n+1}}$$

(for proof see exercise), and so, we can use this recurrence relation to generate a divided difference tableau (see slides).