Model Order Reduction : Theory Guide
Isaac Newton Institute tutorial, Cambridge University

Karthik Duraisamy
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI 48109

January 15, 2023

Contents

Chapter 1

Compression, Sensing and Reconstruction

In working with large sets of data, the natural question to ask is whether we can design techniques to efficiently capture the information in the data. For instance, instead of collecting large amounts of data and compressing it, we might want to just collect a few measurements at some sensor locations, but be able to reconstruct the entire data as needed.

1.1 Sampling and Reconstruction

Let's take a signal $\mathbf{x} \in \mathbb{R}^m$ which may be represented in terms of some basis $\boldsymbol{\Psi} \in \mathbb{R}^{m \times n}$

$$\mathbf{x} = \Psi \mathbf{a},$$

where $\mathbf{a} \in \mathbb{R}^n$ are the basis coefficients.

Let's say we can sub-sample the signal (i.e. we pick elements of $\mathbf{x}$) by multiplying it by a sensor selection matrix $\mathbf{P} \in \mathbb{R}^{p \times m}$. For instance, if the signal $\mathbf{x} \in \mathbb{R}^5$ and we want to just pick the fourth and second measurements (i.e. $p = 2$), then

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{1.1}$$

Then we have a few measurements $\mathbf{y} = \mathbf{P}\mathbf{x} \in \mathbb{R}^p$, for which

$$\mathbf{P}\mathbf{x} = \mathbf{y} = \mathbf{P}\Psi\mathbf{a}.$$

Now, based on the few samples, we can estimate $\hat{\mathbf{a}}$ using

$$\hat{\mathbf{a}} = [\mathbf{P}\Psi]^{+}\mathbf{y}.$$

and we can reconstruct $\mathbf{x}$ as

$$\hat{\mathbf{x}} = \Psi[\mathbf{P}\Psi]^{+}\mathbf{y}.$$

Note that if the original signal had precisely p non-zero coefficients, then we can exactly reconstruct the signal from the p measurements. This is the key idea of sparse sampling leading to reconstruction. There are, however, a number of things we have to take care of, such as:

- How to choose optimal sensor/sampling locations ?
- How to ensure sparsity in $\hat{\mathbf{a}}$?
- How to assure efficiency and robustness? etc.

1.2 Compressed sensing

Compressed sensing strategy is ideal for the recovery of a high-dimensional signal of unknown content using random measurements in a universal basis (rather than a basis that is tailored to the data as in the next section). Again, it makes sense to explicitly find the sparsest possible $\mathbf{a}$, so the problem can be posed as [1]

$$\hat{\mathbf{a}} = min_{\mathbf{a}} \ \ \|\mathbf{a}\|_0 \ \ such \ \ that \ \ \mathbf{y} = \mathbf{P}\Psi\mathbf{a}.$$

However, this turns out to be intractable for high-dimensional $\mathbf{a}$ because of the L^0 semi-norm.

A major innovation, due to Candes et al., proved that for many problems it is probable that the L^1 norm is equivalent to the L^0 semi-norm. This technique, called compressed sensing in the signal processing community relaxes the above problem and instead uses

$$\hat{\mathbf{a}} = min_{\mathbf{a}} \ \ \|\mathbf{a}\|_1 \ \ such \ \ that \ \ \mathbf{y} = \mathbf{P}\Psi\mathbf{a}.$$

This is a convex optimization problem.

For the L^1 norm to mimic the L^0 semi-norm with a high probability, $\mathbf{P}$ has to satisfy the restricted isometry property:

$$(1-\delta_k)\|\mathbf{a}\|_2^2 \leq \|\mathbf{P}\Psi\mathbf{a}\|_2^2 \leq (1+\delta_k)\|\mathbf{a}\|_2^2,$$

where $\delta_k > 0$ is a small constant. This is typically satisfied if $\mathbf{P}$ is incoherent to Ψ (i.e. rows of $\mathbf{P}$ are uncorrelated with columns of Ψ) and $p \sim O(Klog(m/K))$, where K is the sparsity of $\mathbf{a}$.

In compressed sensing, the measurement locations are typically randomized and the strategy relies on inherent sparsity and a sufficient number of measurements. Given the sparsity, compressed sensing can beat the Shannon-Nyquist theorem (which assumes densely populated basis and constant sampling rates).

In the presence of noise, i.e. $\mathbf{y} = \mathbf{P}\Psi\mathbf{a} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$,

$$\hat{\mathbf{a}} = min_{\mathbf{a}} \ \ \|\mathbf{a}\|_1 \ \ such \ \ that \ \ \|\mathbf{y} - \mathbf{P}\Psi\mathbf{a}\|_2 \leq \sigma.$$

[1] Typically compressed sensing is posed in a non-tailored basis such as Fourier basis. However, we live in a free world and so we can apply it to tailored basis such as POD.

1.3 Sensing: Empirical interpolation

Manohar et al. note that "If information is available about the type of signal (e.g., the signal is a turbulent velocity field or an image of a human face), it is possible to design optimized sensors that are tailored for the particular signals of interest. Dominant features are extracted from a training dataset consisting of representative examples, for example using the proper orthogonal decomposition (POD). These low-rank features, mined from patterns in the data, facilitate the design of specialized sensors that are tailored to a specific problem."

Given a basis Ψ, the idea then is to find a $\mathbf{P}^*$ that minimizes the difference between the real and reconstructed signal. In other words,

$$\mathbf{P}^* = min_{\mathbf{P}} \|\mathbf{x} - \Psi[\mathbf{P}\Psi]^{+}\mathbf{y}\|_2.$$

In popular approaches, $p = n$ is used and $\mathbf{P}^*$ is chosen as

$$\mathbf{P}^* = min_{\mathbf{P}} \|\mathbf{x} - \Psi[\mathbf{P}\Psi]^{-1}\mathbf{y}\|_2.$$

Sorensen et al. have shown that, for orthonormal Ψ,

$$\|\mathbf{x} - \Psi[\mathbf{P}\Psi]^{-1}\mathbf{P}\mathbf{x}\|_2 \leq \|[\mathbf{P}\Psi]^{-1}\|_2\|[\mathbf{I} - \Psi\Psi^T]\mathbf{x}\|_2$$

The second term in the RHS is basically projection error (a compression problem), and the first term is related to sampling error (a sensing problem). Thus Discrete Empirical Interpolation Method (DEIM)-based approaches simplify the sensor selection problem to:

$$\mathbf{P}^* = min_{\mathbf{P}} \|[\mathbf{P}\Psi]^{-1}\|_2.$$

Perhaps the most elegant way of approaching this problem is the QDEIM method which relies on the pivoted QR decomposition.

Pivoted QR decomposition
Given a matrix $\mathbf{W} \in \mathbb{R}^{n \times m}$, if we use a column pivoted QR decomposition

$$\mathbf{W}\Phi = \mathbf{Q}\mathbf{R},$$

where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{n \times m}$, then we can separate this into

$$\mathbf{W}\Phi = \mathbf{Q}[\mathbf{R}_1 \;\; \mathbf{R}_2], \tag{1.2}$$
$$\tag{1.3}$$

where $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$ is an upper triangular matrix. Thus, the columns of the matrix $\mathbf{W}$ are effectively permuted (using the matrix $\mathbf{\Phi}$ such that the diagonal elements of $\mathbf{R}_1$ are non-increasing. Manohar et al. further note that:

$$\sigma_i^2 = |r_{ii}|^2 \;\; ; \;\; 1 \leq i \leq n. \tag{1.4}$$

Now for our problem, set $\mathbf{W} = \Psi^T$. What we want is $\mathbf{P}\Psi = [\mathbf{QR}_1]^T = \mathbf{R}_1^T\mathbf{Q}^T$. Thus

$$\|\mathbf{P}\Psi\|_2 = \|\mathbf{R}_1^T\mathbf{Q}^T\|_2 = \sigma_{max}(\mathbf{R}_1).$$

and

$$\|[\mathbf{P}\Psi]^{-1}\|_2 = \frac{1}{\sigma_{min}(\mathbf{R}_1)}.$$

It is thus in our best interests to keep $\sigma_{min}(\mathbf{R}_1)$ as large as possible. The column pivoted QR decomposition accomplishes this courtesy of eq. 1.4 and the columns of $\boldsymbol{\Psi}$ give us sensor locations of decreasing importance.

So the QDEIM algorithm computes the column-pivoted $\mathbf{QR}$ decomposition

$$\boldsymbol{\Psi^T\Phi = QR_1}$$

and uses $\mathbf{P}^* = \boldsymbol{\Phi}^T$.

Just.so.elegant.

1.4 Compression: POD

There are many ways to find a basis, but perhaps the most pertinent and popular is the Proper Orthogonal Decomposition. Assume a set of samples of a state variable $\mathbf{x}_i \in \mathbb{R}^m$, where $1 \leq i \leq t$. The objective of POD analysis is to find the optimal basis vectors (and the projection) that can best represent the data $\mathbf{X} = [\mathbf{x}_1 \;\; \mathbf{x}_2 \;\; ....\mathbf{x}_t]$. While several interpretations and formulations of this problem are possible, a couple are given below:

$$\min_{\Psi \in \mathbb{R}^{m\times n}} \|\mathbf{X} - \Psi\Psi^T\mathbf{X}\|_F, \;\; subject \;\; to \;\; \Psi^T\Psi = \mathbf{I}_n.$$

In other words, we want to find an optimal projection (in a Frobenius norm sense) of rank $n \leq t$.

Note the above is the same as

$$\min_{\Psi \in \mathbb{R}^{m\times n}} \sum_{i=1}^{t} \|[\mathbf{I} - \Psi\Psi^T]\mathbf{x}_i\|_2, \;\; subject \;\; to \;\; \Psi^T\Psi = \mathbf{I}_n,$$

so this is the equivalent of reducing the projection error over every snapshot $\mathbf{x}$.

According to the Schmidt-Mirsky-Eckard-Young theorem, the solution Ψ is given by the first n left singular vectors of $\mathbf{X}$.

$$\mathbf{X} = \mathbf{U\Sigma V}^*.$$

$\mathbf{U}$ is an orthonormal matrix, $\mathbf{V}$ is a unitary matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix with entries $\{\sigma_1, \sigma_2, ..., \sigma_t\}$ (the singular values of $\mathbf{X}$, typically arranged in descending order.)

Therefore, $\Psi = \mathbf{U}(:, 1 : n)$ (that is, the first n columns of $\mathbf{U}$) would be a sensible option for compression.

1.5 References

This chapter heavily draws from

1 S. S. Chaturantabut and D. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing, 2012.

2. Z. Drmac and S. Gugercin, A new selection operator for the DEIM – improved a priori error bound and extensions, SIAM Journal on Scientific Computing, 2016.

3. Manohar, K., Brunton, B.W., Kutz, J.N. and Brunton, S.L., Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns. IEEE Control Systems, 2018.

Chapter 2

Linear Systems

We are interested in linear time-invariant (LTI) systems here, that arise from semidiscretization of partial differential equations (PDEs). This LTI system therefore, represents a large-scale system in many cases. The purpose of this chapter is to lay out the theories that will be used for reducing the order of such systems via projection-based and data-driven approaches.

2.1 External Description

External description of a system is a mapping from the inputs $\mathbf{u}$ to the outputs $\mathbf{y}$, that for a discrete-time system takes the following form,

$$\mathbf{y} = \mathcal{S}(\mathbf{u}) = \mathbf{h} * \mathbf{u}, \tag{2.1}$$

where $\mathcal{S}$ is a linear operator and $\mathbf{h}$ is a weighting pattern, such that,

$$\mathbf{y}(i) = \sum_{j \in \mathbb{Z}} \mathbf{h}(i,j)\mathbf{u}(j), \qquad i \in \mathbb{Z}. \tag{2.2}$$

The system is time-invariant if,

$$\mathbf{h}(i,j) = \mathbf{h}_{i-j} \quad \in \mathbb{R}^{p \times m}, \tag{2.3}$$

where p is the number of outputs and m is the number of inputs. For a LTI system, the sequence (2.2) represents the impulse response of the system. For a single-input single-output system (i.e., $m = p = 1$), the impulse response is the output of the system when the system is excited by unit impulse,

$$\mathbf{u}(t) = \delta(t) = \begin{cases} 1, & t = 0, \\ 0, & t \neq 0. \end{cases} \tag{2.4}$$

The first term of the impulse response sequence $\mathbf{y}(0)$ is the instantaneous action of the system and the proceeding terms are the delayed action of the system.

For a continuous-time system, the impulse response sequence reads,

$$\mathbf{y}(t) = \mathcal{S}(\mathbf{u}), \tag{2.5}$$

where $\mathcal{S}$ is a convolution operator,

$$\mathcal{S} = \int_{-\infty}^{\infty} \mathbf{h}(t,\tau)\mathbf{u}(\tau)d\tau, \qquad t \in \mathbb{R}, \tag{2.6}$$

and the system is time-variant if,

$$\mathbf{h}(t,\tau) = \mathbf{h}(t-\tau). \tag{2.7}$$

Therefore, the system response can be separated into the instantaneous and dynamic responses as,

$$\mathbf{y}(t) = \mathbf{h}_0\mathbf{u}(t) + \int_{-\infty}^{t} \mathbf{h}_d(t-\tau)\mathbf{u}(\tau)d\tau \tag{2.8}$$

where,

$$\mathbf{h}(t) = \mathbf{h}_0\delta(t) + \mathbf{h}_d(t), \qquad t \geq 0, \tag{2.9}$$

and δ is the delta-distribution. Therefore, $\mathbf{h}$ is the impulse response of the system.

The continuous-time system is causal if,

$$\mathbf{h}(t,\tau) = 0, \qquad t \leq \tau, \tag{2.10}$$

which is equivalent to,

$$\mathbf{h}(i,j) = \mathbf{0}, \qquad i \leq j, \tag{2.11}$$

in the discrete-time system.

For a time-invariant, causal, and smooth continuous-time system, and a time-invariant, causal, discrete-time system, the sequence of $p \times m$ matrices $\mathbf{h}_i$,

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{h}_0 & \mathbf{h}_1 & \dots & \mathbf{h}_i & \dots \end{bmatrix} \qquad \mathbf{h}_i \in \mathbb{R}^{p\times m}, \tag{2.12}$$

is called the sequence of Markov parameters.

Taking the Laplace transform of the external description (2.1), we have,

$$\mathbf{Y}(s) = \mathbf{H}(s)\mathbf{U}(s), \tag{2.13}$$

which gives the transfer function of the system,

$$\mathbf{G}(s) = \frac{\mathbf{Y}(s)}{\mathbf{U}(s)}, \tag{2.14}$$

that identifies the input-output behavior of the system.

2.2 Internal Description

Another way of representing a dynamical system is through its internal description, which employs input $\mathbf{u}$, output $\mathbf{y}$ and state $\mathbf{x}$. For a continuous-time system, the following first-order differential equation is called the state-space representation of the system,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \qquad t \in \mathbb{R}, \tag{2.15}$$

where in a LTI system, $\mathbf{A} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{B} : \mathbb{R}^m \to \mathbb{R}^n$ are constant linear maps. For a discrete-time system, the state-space representation takes the form,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{2.16}$$

where k is the time index.

The output equation is an algebraic equation for both continuous-time and discrete-time systems,

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{2.17}$$

where, $\mathbf{C} : \mathbb{R}^n \to \mathbb{R}^p$ is the output map, and $\mathbf{D}\mathbf{u}$ is called the feedthrough term.

A continuous-time LTI system is stable if all of the eigenvalues of $\mathbf{A}$ have negative real components. Similarly, a discrete-time LTI system is stable if all of the eigenvalues of $\mathbf{A}$ are located inside the unit circle.

The differential equation (2.15) has an analytical solution,

$$\mathbf{x}(t) = e^{\mathbf{A}(t-t_0)}\mathbf{x}_0 + \int_{t_0}^{t} e^{\mathbf{A}(\mathbf{t}-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau, \qquad t \geq t_0. \tag{2.18}$$

Therefore, with $t_0 = -\infty$ and $\mathbf{x}_0 = 0$, the impulse response of the continuous-time system is,

$$\mathbf{y}(t) = \begin{cases} \mathbf{C}e^{\mathbf{A}t}\mathbf{B} + \delta(t)\mathbf{D}, & t \geq 0, \\ \mathbf{0}, & t < 0. \end{cases} \tag{2.19}$$

The solution of the discrete-time system (2.16) takes the form,

$$\mathbf{x}_k = \mathbf{A}^{k-k_0}\mathbf{x}_{k_0} + \sum_{j=k_0}^{k-1} \mathbf{A}^{k-1-j}\mathbf{B}\mathbf{u}(j), \qquad k \geq k_0, \tag{2.20}$$

which yields the following expression for the impulse response,

$$\mathbf{y}_k = \begin{cases} \mathbf{C}\mathbf{A}^{k-1}\mathbf{B}, & k > 0, \\ \mathbf{D}, & k = 0, \\ \mathbf{0}, & k < 0, \end{cases} \tag{2.21}$$

and the sequence of Markov parameters is,

$$\boldsymbol{M} = \begin{bmatrix} \mathbf{D} & \mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{k-1}\mathbf{B} & \dots \end{bmatrix} \tag{2.22}$$

Note that substituting equation (2.16) into (2.17) and taking a Laplace transform, gives the transfer function of the LTI system,

$$\mathbf{G}(s) = \mathbf{D} + \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \tag{2.23}$$

where, $\mathbf{I}$ is the identity matrix.

Proposition

The system transfer function and therefore, the sequence of Markov parameters are invariant under coordinate transformation.

(This can be easily proved by substituting the state transformation $\tilde{\mathbf{x}} = \mathbf{Tx}$ in the state-space and output representations and deriving the transfer function of the transformed system.)

2.3 Reachability

Reachability of a state is a binary definition that identifies whether or not a state $\mathbf{x}$ can be steered via an input $\mathbf{u}$. A system is reachable if all of its states can be excited by the control action.

Consider the solution to the discrete-time system (2.20), a state $\hat{\mathbf{x}}$ of the system (2.16) is reachable from the zero state if there exists a control input $\hat{\mathbf{u}}$ such that,

$$\hat{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{A}^k\mathbf{x}_0 = \begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{k-1}\mathbf{B}\end{bmatrix}\begin{bmatrix}\mathbf{u}_{k-1}\\ \mathbf{u}_{k-2}\\ \mathbf{u}_{k-3}\\ \vdots\\ \mathbf{u}_0\end{bmatrix}, \tag{2.24}$$

where, $\mathscr{P} = \begin{bmatrix}\mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} & \dots & \mathbf{A}^{k-1}\mathbf{B}\end{bmatrix}$ is called the reachability matrix. The rank of this matrix is identified by its first n terms, where n is the order of the system.

Theorem An LTI system (continuous-time or discrete-time) is reachable, if and only if $\mathcal{R}(\mathscr{P}) = n$, where $\mathcal{R}$ denotes matrix rank.

According to the above theorem, in a LTI system, the originally analytical definition of reachability reduces to an algebraic definition that depends only on the system matrices rather than time and input function.

Reachability is basis independent, that is, it does not change under coordinate transformation. For a nonsingular transformation matrix $\mathbf{T}$ (i.e., $det(\mathbf{T}) \neq 0$), we have,

$$\mathcal{R}(\mathbf{T}\mathscr{P}\mathbf{T}^*) = \mathbf{T}\mathcal{R}(\mathscr{P}).$$

Reachability and controllability are equivalent in continuous-time systems. Controllability is identified by driving the system from a non-zero state to the zero state. In discrete-time systems, $\mathbb{X}_{reach} \subset \mathbb{X}_{contr}$, where $\mathbb{X}_{reach}$ is the reachable subspace and $\mathbb{X}_{contr}$ is the controllable subspace. Therefore, controllability is a weaker concept than reachability in discrete-time systems, while for continuous systems $\mathbb{X}_{reach} = \mathbb{X}_{contr}$.

2.4 Observability

The concept of observability determines whether we are able to identify the state from the output of the system. Therefore, a state $\hat{\mathbf{x}}$ is unobservable if $\mathbf{y}(t) = 0$ for all $t \geq 0$. The observability matrix is defined as below,

$$\mathscr{O} = \begin{bmatrix} \mathbf{C}^* & \mathbf{A}^*\mathbf{C}^* & (\mathbf{A}^*)^2\mathbf{C}^* & \dots \end{bmatrix}^*. \tag{2.25}$$

Similar to the reachability matrix, the rank of the observability matrix is identified by its first n terms. Therefore, it is sufficient to compute only the first n terms of this matrix, that is,

$$\mathscr{O} = \begin{bmatrix} \mathbf{C}^* & \mathbf{A}^*\mathbf{C}^* & (\mathbf{A}^*)^2\mathbf{C}^* & \dots & (\mathbf{A}^*)^{n-1}\mathbf{C}^* \end{bmatrix}^*. \tag{2.26}$$

Observability of both continuous-time and discrete-time systems requires that $\mathcal{R}(\mathscr{O}) = n$. The observability matrix is invariant under coordinate transformation.

Theorem Observability and reachability are dual definitions, that is, a system is reachable if and only if its dual (adjoint) system is observable.

2.5 System Realization

Having the internal description of the system (i.e., the inputs, outputs, and the state), it is easy to obtain the external description that relies only on the input-output mapping. On the other hand, given the external description (i.e., the transfer function or the impulse response) of the system, it is not trivial to obtain the internal description (i.e., the triplet $(\mathbf{A}, \mathbf{B}, \mathbf{C})$). This is called the realization problem.

For any system, there are infinitely many realizations that will generate an identical output for a particular input. The realization with the smallest state-space dimension among all realizations is called the *minimum realization*.

One of the basic components of system realization is the generalized Hankel matrix that is built by the sequence of Markov parameters,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \dots & \mathbf{h}_{m_p} \\ \mathbf{h}_2 & \mathbf{h}_3 & \dots & \mathbf{h}_{m_p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{m_o} & \mathbf{h}_{m_o+1} & \dots & \mathbf{h}_{m_o+m_p+1}. \end{bmatrix} \tag{2.27}$$

It is easy to observe that $\mathbf{H} = \mathscr{O}\mathscr{P}$. Therefore, if the system is reachable and observable, $m_p \geq n$, and $m_o \geq n$, the Hankel matrix is of rank n, where n is the order of the system.

2.6 System Gramians

Reachability and observability are binary definitions that identify whether or not a system is reachable and observable. The degree of reachability and observability however, are quantities that can be measured by the system Gramians.

For a continuous-time system, the finite reachability Gramian is defined as,

$$\mathscr{W}_p(t) = \int_0^t e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*\tau} d\tau, \qquad t \in \mathbb{R}_+. \tag{2.28}$$

The reachability Gramian for a discrete-time system is,

$$\mathscr{W}_p(t) = \mathscr{C}\mathscr{C}^* = \sum_{k=0}^{t-1} \mathbf{A}^k \mathbf{B}\mathbf{B}^*(\mathbf{A}^*)^k, \qquad t \in \mathbb{Z}_+. \tag{2.29}$$

A LTI system is reachable if and only if $\mathscr{W}_p$ is positive-definite for some $t > 0$.

Proposition

If a system is reachable, the minimum energy required to reach a state $\hat{\mathbf{x}}$ at time $\hat{T}$ is obtained by $\hat{\mathbf{x}}^*\mathscr{W}_p(\hat{T})^{-1}\hat{\mathbf{x}}$. Therefore, the reachability Gramian provides a measure of the degree of the reachability of the system in a certain direction.

Similarly, the finite observability Gramian is defined as,

$$\mathscr{W}_o(t) = \int_0^t e^{\mathbf{A}^*\tau}\mathbf{C}^*\mathbf{C} e^{\mathbf{A}\tau} d\tau, \qquad t \in \mathbb{R}_+, \tag{2.30}$$

for a continuous-time system, and $\mathscr{W}_o(t) = \mathscr{O}^*\mathscr{O}$ for $t \in \mathbb{Z}_+$ in a discrete-time system. Thus, the output energy of the system at time T caused by the initial state $\mathbf{x}$ can be identified by $\mathbf{x}^*\mathscr{W}_o(T)\mathbf{x}$.

If the continuous-time system is stable (i.e., its eigenvalues are located in the left half of the complex plane), we can define the infinite reachability Gramian,

$$\mathscr{W}_p = \int_0^\infty e^{\mathbf{A}\tau}\mathbf{B}\mathbf{B}^* e^{\mathbf{A}^*\tau} d\tau. \tag{2.31}$$

This Gramian is the solution to the following Lyapunov equation,

$$\mathbf{A}\mathscr{W}_p + \mathscr{W}_p\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathbf{0}. \tag{2.32}$$

Similarly, the infinite observability Gramian for a continuous-time system is,

$$\mathscr{W}_o = \int_0^\infty e^{\mathbf{A}^*\tau}\mathbf{C}^*\mathbf{C} e^{\mathbf{A}\tau} d\tau, \tag{2.33}$$

which is the solution to another Lyapunov equation,

$$\mathbf{A}^*\mathscr{W}_o + \mathscr{W}_o\mathbf{A} + \mathbf{C}^*\mathbf{C} = \mathbf{0}. \tag{2.34}$$

The infinite Gramians can also be defined for a discrete-time system by computing the reachability and observability matrices for an infinite time horizon. The discrete-time infinite reachability Gramian is the solution of the discrete-time Lyapunov equation,

$$\mathbf{A}\mathscr{W}_p\mathbf{A}^* + \mathbf{B}\mathbf{B}^* = \mathscr{W}_p. \tag{2.35}$$

The discrete-time infinite observability Gramian is the solution to a different discrete-time Lyapunov equation,

$$\mathbf{A}^*\mathscr{W}_o\mathbf{A} + \mathbf{C}^*\mathbf{C} = \mathscr{W}_o. \tag{2.36}$$

Proposition

The eigenvalues of the product of the reachability and observability Gramians (i.e., $\mathscr{W}_p\mathscr{W}_o$) are input-output invariant. These eigenvalues are related to Hankel singular values by $\sigma_i(\mathbf{H}) = \sqrt{\lambda_i(\mathscr{W}_p\mathscr{W}_o)}$.

The product of the reachability and observability matrices forms a third Gramian called the cross Gramian. For a discrete-time system, the cross Gramian $\mathscr{W}_x$ is an $n \times n$ matrix given by $\mathscr{W}_x = \mathscr{P}\mathscr{O}$. The continuous-time cross Gramian can be obtained by,

$$\mathscr{W}_x = \int_0^\infty e^{\mathbf{A}t}\mathbf{B}\mathbf{C}e^{\mathbf{A}t}dt. \tag{2.37}$$

While, eigenvalues of the reachability and observability Gramians are not invariant, similar to the product $\mathscr{W}_p\mathscr{W}_o$, eigenvalues of the cross Gramian are also input-output invariant for both continuous- and discrete-time systems. In fact, eigenvalues of the cross Gramian are the same as eigenvalues of the Hankel operator.

2.7 References

1. Antoulas, A.C., 2005. Approximation of large-scale dynamical systems. Society for Industrial and Applied Mathematics.

2. Juang, J.N., 1994. Applied system identification. Prentice-Hall, Inc.

3. Benner, P., Ohlberger, M., Cohen, A. and Willcox, K. eds., 2017. Model reduction and approximation: theory and algorithms. Society for Industrial and Applied Mathematics.

Chapter 3

Reduced Order Models

In many problems in science and engineering, we may be able to generate numerical solutions using a known set of "high-fidelity" model simulations. However, we might be able to do this only for a handful of operating conditions (or input parameters) $\boldsymbol{\mu}$. Reduced order models (ROMs) seek to derive a set of equations of reduced complexity based on high-fidelity data and knowledge of the governing equations of the high-fidelity model (HFM).

To emphasize the basic idea, consider the solution $\mathbf{q} \in \mathbb{R}^n$ of a high-fidelity model. Let's say the high-fidelity model is given by

$$\frac{d\mathbf{q}(t)}{dt} = \mathbf{f}(\mathbf{q}(t)) \quad ; \quad \mathbf{q}(0) = \mathbf{q}_0. \tag{3.1}$$

We are looking for the solution of a reduced set of variables $\mathbf{q}_r \in \mathbb{R}^k$,

$$\frac{d\mathbf{q}_r(t)}{dt} = \mathbf{f}_r(\mathbf{q}_r(t)) \quad ; \quad \mathbf{q}_r(0) = \mathbf{q}_{r0}, \tag{3.2}$$

where $k << n$.

ROMs can be obtained in an intrusive fashion or a non-intrusive fashion.

3.1 Projections

In this chapter, we will specifically consider projection-based ROMs. Let us first define projections in a rigorous sense.

Consider a sub-space $\mathcal{S}_1 \subset \mathbb{R}^m$ spanned by the basis functions $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ...\mathbf{u}_n\}$, where $\mathbf{U} \in \mathbb{R}^{m \times n}$ and $\mathbf{u} \in \mathbb{R}^m$.

If $\mathbf{U}$ is an orthonormal matrix, then $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.

A projection matrix $\boldsymbol{\Pi} \in \mathbb{R}^{m \times m}$ satisfies the following property : $\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}$.

If $rank(\boldsymbol{\Pi}) = k$, then there exists a basis $\mathbf{X}$ such that

$$\boldsymbol{\Pi} = \mathbf{X} \begin{bmatrix} \mathbf{I}_k & \cdot \\ \cdot & \mathbf{0}_{m-k} \end{bmatrix} \mathbf{X}^{-1}.$$

We can separate the bases into two sets

$$\mathbf{X} = [\mathbf{X}_1 \quad ; \quad \mathbf{X}_2], \quad with \ \ \mathbf{X}_1 \in \mathbb{R}^{m \times k}, \quad \mathbf{X}_2 \in \mathbb{R}^{m \times (m-k)}.$$

For any vector $\mathbf{x} \in \mathbb{R}^m$, we have

- $\mathbf{\Pi x} \in \quad range(\mathbf{X}_1) = \quad range(\mathbf{\Pi}) = \mathcal{S}_1$
- $(\mathbf{I} - \mathbf{\Pi})\mathbf{x} \in \quad range(\mathbf{X}_2) = range(\mathbf{I} - \mathbf{\Pi}) = \quad kernel(\mathbf{\Pi}) = \mathcal{S}_2$.

So $\mathbf{\Pi}$ defines the projection onto $\mathcal{S}_1$ parallel to $\mathcal{S}_2$.

Also, $\mathcal{S}_1 \oplus \mathcal{S}_2 = \mathbb{R}^m$.

An **orthogonal projection** is defined by $\mathbf{\Pi} = \mathbf{U}\mathbf{U}^T$. In this case, $\mathcal{S}_2 = \mathcal{S}_1^{\perp}$.

For an **oblique projection**, let's consider $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_n\}$ as a basis for $\mathcal{S}_2^{\perp}$.

Since $\mathbf{\Pi x} \in \mathcal{S}_1$, we can write it as $\mathbf{\Pi x} = \mathbf{U y}$ for some $\mathbf{y} \in \mathbb{R}^n$. Then $(\mathbf{I} - \mathbf{\Pi})\mathbf{x} \in \mathcal{S}_2$, we have

$$\mathbf{W}^T(\mathbf{x} - \mathbf{U y}) = \mathbf{0} \rightarrow \mathbf{W}^T\mathbf{x} = \mathbf{W}^T\mathbf{U y} \rightarrow \mathbf{y} = [\mathbf{W}^T\mathbf{U}]^{-1}\mathbf{W}^T\mathbf{x}.$$

Therefore the oblique projection is

$$\mathbf{\Pi} = \mathbf{U}[\mathbf{W}^T\mathbf{U}]^{-1}\mathbf{W}^T.$$

This describes the projection onto $\mathcal{S}_1$ (spanned by $\mathbf{U}$) perpendicular to $\mathcal{S}_2^{\perp}$ (spanned by $\mathbf{W}$).

3.2 Projection-based ROMs

Projection-based ROMs derive the reduced set of equations using projection operators as discussed above. First, define a trial basis $\mathbf{V} \in \mathbb{R}^{n \times k}$ that spans a subspace $\mathcal{V} \subset \mathbb{R}^n$. Then we can write

$$\mathbf{q} = \mathbf{V}\mathbf{q}_r + \mathbf{V}^{\perp}\mathbf{q}_p.$$

The truncated basis $\mathbf{V}^{\perp}$ span a subspace $\mathcal{V}^{\perp}$, such that $\mathcal{V} \oplus \mathcal{V}^{\perp} = \mathbb{R}^n$.

In ROMs, we assume

$$\tilde{\mathbf{q}} = \mathbf{V}\mathbf{q}_r$$

is a reasonable approximation to $\mathbf{q}$.

and assume that the following equation holds [1]:

$$\frac{d\mathbf{V}\mathbf{q}_r(t)}{dt} = \mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t), \quad \mathbf{V}\mathbf{q}_r(0) = \mathbf{q}_0 \tag{3.3}$$

Now define a test basis $\mathbf{W} \in \mathbb{R}^{n \times k}$ that spans a subspace $\mathcal{W} \subset \mathbb{R}^n$, then

$$\mathbf{W}^T\mathbf{V}\frac{d\mathbf{q}_r(t)}{dt} = \mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t), \quad \mathbf{W}^T\mathbf{V}\mathbf{q}_r(0) = \mathbf{W}^T\mathbf{q}_0. \tag{3.4}$$

[1] This is not a precise statement because there is a truncation error, the effect of which results in the so-called closure problem. That is for later

Then, if $\mathbf{W}^T\mathbf{V}$ is non-singular, we have the ROM-equations

$$\frac{d\mathbf{q}_r(t)}{dt} = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t), \quad \mathbf{q}_r(0) = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{q}_0. \tag{3.5}$$

This is, of course a lower-dimensional system of equations to solve and $\mathbf{f}_r(\mathbf{q}_r(t)) = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t)$.

Special case: Galerkin projection
In Galerkin projection, the test basis is taken to be the same as the trial basis. Thus $\mathbf{W} = \mathbf{V}$ and $\mathbf{\Pi} = \mathbf{V}\mathbf{V}^T$. So the ROM is

$$\frac{d\mathbf{q}_r(t)}{dt} = \mathbf{V}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t), \quad \mathbf{q}_r(0) = \mathbf{V}^T\mathbf{q}_0. \tag{3.6}$$

Effective full-order approximation
Note that the approximate full order system can be recovered using $\tilde{\mathbf{q}} = \mathbf{V}\mathbf{q}_r$, so the equivalent full order system is

$$\frac{d\tilde{\mathbf{q}}(t)}{dt} = \mathbf{V}[\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{f}(\tilde{\mathbf{q}}(t), t), \quad \tilde{\mathbf{q}}(0) = \mathbf{V}[\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{q}_0. \tag{3.7}$$

Defining a projector $\mathbf{\Pi} = \mathbf{V}[\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T$ (as in the previous chapter), we have:

$$\frac{d\tilde{\mathbf{q}}(t)}{dt} = \mathbf{\Pi}\mathbf{f}(\tilde{\mathbf{q}}(t), t), \quad \tilde{\mathbf{q}}(0) = \mathbf{\Pi}\mathbf{q}_0. \tag{3.8}$$

This is the effective full-order system that is solved by the ROM.
NOTE: $\mathbf{\Pi}\mathbf{q} \in \quad range(\mathbf{V})$ and $(\mathbf{I} - \mathbf{\Pi})\mathbf{q} \quad range(\mathbf{V}_\perp)$.
This describes the projection onto $\mathcal{V}$ (spanned by $\mathbf{V}$) perpendicular to a subspace $\mathcal{W}$ spanned by $\mathbf{W}$.

3.3 Balanced Truncation

A popular class of projection-based model reduction methods is Balanced Truncation (BT). This method was developed by Moore in 1981. Measuring the importance of dynamical structures according to their energy in POD-based ROMs causes practical challenges in applications that involve low-energy structures that are critical in the dynamics of the system (e.g., acoustic waves). These low-energy structures are the states that are not easy to reach, but on the other hand, a small actuation energy is sufficient to excite them to the extent that their influence on the system output is highly observable. In other words, certain low-energy structures are hardly reachable but highly observable. These structures are typically lost through the energy-based modal truncation in POD.

BT addresses this issue by first transforming the high-dimensional system,

$$\begin{aligned} \frac{d\mathbf{x}(t)}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \end{aligned} \tag{3.9}$$

into coordinates in which the most reachable directions are aligned with the most observable directions. Therefore, the first step in BT is to find a transformation matrix $\mathbf{T}$ such that,

$$\mathbf{T}\mathscr{W}_p\mathbf{T}^* = \mathbf{T}^{-*}\mathscr{W}_o\mathbf{T}^{-1} = \mathbf{\Sigma}, \tag{3.10}$$

where, $\mathbf{\Sigma}$ is a diagonal matrix and its diagonal elements are the Hankel singular values σ_i, $i = 1, \ldots, n$. Therefore, in the new coordinate system the reachability and observability Gramians are equal and diagonal, and $\mathbf{T}$ is called a balancing transformation. Each state in the new coordinate system is as observable as it is reachable. Knowing this, we can truncate the states that are difficult to reach, as these states are also difficult to observe, that is, they have a small influence on the output of the system. ROMs constructed by balanced truncation are bounded above by twice the sum of the truncated Hankel singular values.

The original BT method first balances the system and then truncates the uncontrollable and unobservable states. Therefore, it involves inversion of high-dimensional matrices that is ill-conditioned in stiff systems. It is better advised to use a method called square root balancing. First Cholesky factorization of the reachability Gramian $\mathscr{W}_P = \mathbf{U}\mathbf{U}^*$ and the observability Gramian $\mathscr{W}_o = \mathbf{L}\mathbf{L}^*$ are computed. Then following singular value decomposition of the product $\mathbf{U}^*\mathbf{L} = \mathbf{W}\mathbf{\Sigma}\mathbf{V}^*$, singular vectors corresponding to smaller singular values (diagonal elements of $\mathbf{\Sigma}$) are truncated. Next, the balancing transformation,

$$\mathbf{T} = \mathbf{\Sigma}_r^{-1/2}\mathbf{V}_r^*\mathbf{L}^*, \tag{3.11}$$

and the inverse transformation,

$$\mathbf{T}^{-1} = \mathbf{U}\mathbf{W}_r\mathbf{\Sigma}_r^{-1/2}, \tag{3.12}$$

are computed, where the subscript r denotes matrices after truncation. The original high-dimensional system is then transformed to the balanced coordinates, which results in the following low-dimensional system or the balanced reduced-order model,

$$\begin{aligned} \frac{d\mathbf{x}_r(t)}{dt} &= \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C}_r\mathbf{x}_r + \mathbf{D}\mathbf{u}, \end{aligned} \tag{3.13}$$

where, $\mathbf{A}_r = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$, $\mathbf{B}_r = \mathbf{T}\mathbf{B}$, and $\mathbf{C}_r = \mathbf{C}\mathbf{T}^{-1}$. Therefore, BT can be viewed as a Petrov-Galerkin projection, where the test subspace is obtained by the observability Gramian. BT enables theoretical error bounds,

$$||\mathbf{G} - \mathbf{G}_r||_\infty \leq 2 \sum_{j=r+1}^{n} \sigma_j, \tag{3.14}$$

$$||\mathbf{G} - \mathbf{G}_r||_\infty > \sigma_{r+1}, \tag{3.15}$$

where, $\mathbf{G}$ and $\mathbf{G}_r$ are the transfer functions of the full- and reduced-order models, and σ_j is the jth Hankel singular value (i.e.,diagonal entries of the balanced Gramians). The lower-bound is satisfied by any ROM, but the upper-bound is a result of balancing transformation.

3.4 Approximate Balanced Truncation

There are multiple scenarios in which analytical BT cannot be implemented. For example when the FOM is unstable, Gramians cannot be computed analytically. On the other hand, even though the square root algorithm balances the system after truncation of the smaller Hankel singular values and therefore, improves the condition number of the transformation matrices, Cholesky factorization of the high-dimensional Gramians may introduce numerical instabilities in highly stiff systems. Approximate BT, also known as balanced POD (BPOD) is an empirical approach developed to address these issues. In this method, instead of solving the Lyapunov equations, Gramians are computed empirically using the impulse response of the system.

Consider the discrete-time system,

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \tag{3.16}$$

it is possible to compute the empirical reachability matrix using the impulse response of this system (the sequence of Markov parameters),

$$\mathscr{P} = \begin{bmatrix}\mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{m_p-1}\mathbf{B}\end{bmatrix}, \tag{3.17}$$

where, $m_p = n$. The empirical reachability Gramian is then computed as,

$$\mathscr{W}_p = \mathscr{P}\mathscr{P}^*. \tag{3.18}$$

To compute the empirical observability Gramian, the impulse response response of the adjoint system,

$$\mathbf{x}_{k+1} = \mathbf{A}^*\mathbf{x}_k + \mathbf{C}^*\mathbf{y}_k, \tag{3.19}$$

needs to be collected. Having the empirical observability matrix,

$$\mathscr{O} = \begin{bmatrix}\mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{m_o-1}\end{bmatrix}, \tag{3.20}$$

where $m_o = n$. The empirical observability Gramian is computed as,

$$\mathscr{W}_o = \mathscr{O}^*\mathscr{O}. \tag{3.21}$$

To obtain empirical reachability and observability matrices from the continuous-time system, it is possible to uniformly sample the output of the direct and adjoint systems.

Note that in the case of multi-input systems with m inputs, a separate direct impulse response should be collected for each of the m inputs. Similarly, for a multi-output system with p outputs, a separate adjoint impulse response should be collected for each of the p outputs to compute the empirical Gramians. Therefore, this approach can become expensive for systems with many inputs or outputs (e.g., full-state output scenarios).

There are two methods to reduce the order of the system using the direct and adjoint impulse response snapshots. One way is to directly use the Cholesky factorization of the empirical Gramians to obtain the balancing transformation. Another more efficient way is to use what is called the method of snapshots (Sirovich, 1987) to bypass computation of the empirical Gramians and instead, use the product $\mathscr{O}\mathscr{P}$ to assemble the generalized Hankel matrix,

$$\mathbf{H} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^{m_o-1} \end{bmatrix} \begin{bmatrix} \mathbf{B} & \mathbf{AB} & \dots & \mathbf{A}^{m_p-1}\mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{CB} & \mathbf{CAB} & \dots & \mathbf{CA}^{m_p-1}\mathbf{B} \\ \mathbf{CAB} & \mathbf{CA}^2\mathbf{B} & \dots & \mathbf{CA}^{m_p}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA}^{m_o-1}\mathbf{B} & \mathbf{CA}_{m_o}\mathbf{B} & \dots & \mathbf{CA}^{m_p+m_o-2}\mathbf{B} \end{bmatrix}. \tag{3.22}$$

Next, SVD of the Hankel matrix is computed,

$$\mathbf{H} = \mathbf{U\Sigma V}^* = \begin{bmatrix} \mathbf{U}_r & \mathbf{U}_t \end{bmatrix} \begin{bmatrix} \mathbf{\Sigma}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_t \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^* & \mathbf{V}_t^*. \end{bmatrix} \tag{3.23}$$

The first r singular values and singular vectors of the Hankel matrix are retained and the rest of them are truncated.

Using the impulse response snapshots and the Hankel singular values, the direct modes,

$$\mathbf{\Psi} = \mathscr{P}\mathbf{V}_r\mathbf{\Sigma}_r^{-1/2}, \tag{3.24}$$

and the adjoint modes,

$$\mathbf{\Phi} = \mathscr{O}^*\mathbf{U}_r\mathbf{\Sigma}_r^{-1/2}, \tag{3.25}$$

are computed. The direct and adjoint modes are biorthogonal (i.e., $\mathbf{\Phi}^*\mathbf{\Psi} = \mathbf{I}$). These modes are then used for approximate balancing of the LTI system, where the reduced-order system matrices are obtained by $\mathbf{A}_r = \mathbf{\Phi}^*\mathbf{A\Psi}$, $\mathbf{B}_r = \mathbf{\Phi}^*\mathbf{B}$, and $\mathbf{C}_r = \mathbf{C\Psi}$.

In practice, computing empirical Gramians may be subject to errors and therefore, the theoretical error bounds provided by the analytical BT may not be satisfied here.

BPOD is equivalent to POD, when the impulse response snapshots are used to train the POD-based ROM and the observability Gramian is used as the inner product to compute the POD modes and project the governing equations onto the low-dimensional space.

One way to avoid the prohibitive cost of adjoint simulations in systems with a large number of outputs (e.g., full-state) is to project the system output $\mathbf{y}$ onto the leading POD modes of the direct system impulse response snapshots and solve the output-projected adjoint equations,

$$\mathbf{x}_{k+1} = \mathbf{A}^*\mathbf{x}_k + \mathbf{C}^*\boldsymbol{\xi}_i\hat{\mathbf{y}}_k, \tag{3.26}$$

where, $\boldsymbol{\xi}_i$ is the matrix containing the first i POD modes of the direct system impulse response, and $\hat{\mathbf{y}}_k \in \mathbb{R}^i$. Therefore, instead of performing adjoint simulations p times, where p is the presumably large number of outputs, we run adjoint simulations only i times. However, this method is only efficient if the energy of the POD modes decays fast and i is small. In certain applications, a large number of POD modes is required to capture most of the energy in the impulse response and adjoint simulations with output projection may still be expensive. In addition, we do not have access to the adjoint system in experiments. Therefore, BPOD is not applicable to experiments. Both of these drawbacks are addressed by a non-intrusive balancing transformation method called the eigensystem realization algorithm (ERA).

3.5 Errors in ROMs

The error in the ROM is defined as the difference between the solution of the FOM $\mathbf{q}$ and the approximate solution $\tilde{\mathbf{q}} = \mathbf{V}\mathbf{q}_r$:

$$\begin{aligned}
\boldsymbol{\epsilon}(t) &= \mathbf{q}(t) - \tilde{\mathbf{q}}(t) && (3.27)\\
&= \mathbf{q}(t) - \mathbf{V}\mathbf{q}_r(t) && (3.28)\\
&= \mathbf{q}(t) - \boldsymbol{\Pi}\mathbf{q}(t) + \boldsymbol{\Pi}\mathbf{q}(t) - \mathbf{V}\mathbf{q}_r(t) && (3.29)\\
&= [(\mathbf{I} - \boldsymbol{\Pi})\mathbf{q}(t)] + [\boldsymbol{\Pi}\mathbf{q}(t) - \mathbf{V}\mathbf{q}_r(t)] && (3.30)\\
&= \boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t) + \boldsymbol{\epsilon}_{\parallel}(t). && (3.31)
\end{aligned}$$

We can interpret this as an ‘projection’ error and a parallel ‘ROM’ error.

$\|\boldsymbol{\epsilon}(t)\|_2^2 \leq \|\boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t)\|_2^2 + \|\boldsymbol{\epsilon}_{\parallel}(t)\|_2^2$.

$\boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t)$ is thus a lower bound of the error and can be computed a priori.

It is clear that we cannot do much to control the projection error. This is present even at the initial condition.

To better understand the evolution of the parallel error, let’s consider

$$\begin{aligned}
\frac{d\boldsymbol{\epsilon}_{\parallel}}{dt} &= \frac{d}{dt}[\boldsymbol{\Pi}\mathbf{q}(t)] - \frac{d}{dt}[\mathbf{V}\mathbf{q}_r(t))] && (3.32)\\
&= \frac{d}{dt}[\boldsymbol{\Pi}\mathbf{q}(t)] - \frac{d}{dt}\tilde{\mathbf{q}}(t) && (3.33)\\
&= \frac{d}{dt}[\boldsymbol{\Pi}\mathbf{q}(t)] - \boldsymbol{\Pi}\mathbf{f}(\tilde{\mathbf{q}}(t), t) && (3.34)
\end{aligned}$$

With an initial condition $\boldsymbol{\epsilon}_{\parallel} = \mathbf{0}$.

Thus we have the error transport equation

$$\frac{d\boldsymbol{\epsilon}_\parallel}{dt} = \frac{d}{dt}[\boldsymbol{\Pi}\mathbf{q}(t)] - \boldsymbol{\Pi}\mathbf{f}(\tilde{\mathbf{q}}(t), t) \quad ; \quad \boldsymbol{\epsilon}_\parallel(0) = \mathbf{0}. \tag{3.35}$$

If $\boldsymbol{\Pi}$ is constant in time, then

$$\frac{d\boldsymbol{\epsilon}_\parallel}{dt} = \boldsymbol{\Pi}\left[\mathbf{f}(\mathbf{q}(t), t) - \mathbf{f}(\tilde{\mathbf{q}}(t), t)\right] \tag{3.36}$$

The RHS in Eq. 3.36 is important, because if one considers the exact equation (and Π to be constant in time)

$$\begin{aligned} \frac{d\mathbf{q}(t)}{dt} &= \mathbf{f}(\mathbf{q}(t), t) &(3.37)\\ \boldsymbol{\Pi}\frac{d\mathbf{q}(t)}{dt} + (\mathbf{I}-\boldsymbol{\Pi})\frac{d\mathbf{q}(t)}{dt} &= \mathbf{f}(\boldsymbol{\Pi}\mathbf{q}(t), t) + [\mathbf{f}(\mathbf{q}(t), t) - \mathbf{f}(\boldsymbol{\Pi}\mathbf{q}(t), t)] &(3.38)\\ \frac{d\boldsymbol{\Pi}\mathbf{q}(t)}{dt} &= \boldsymbol{\Pi}\mathbf{f}(\boldsymbol{\Pi}\mathbf{q}(t), t) + \boldsymbol{\Pi}[\mathbf{f}(\mathbf{q}(t), t) - \mathbf{f}(\boldsymbol{\Pi}\mathbf{q}(t), t)] &(3.39) \end{aligned}$$

Thus if we have the exact $\tilde{\mathbf{q}}(t) = \boldsymbol{\Pi}\mathbf{q}(t)$,

$$\frac{d\tilde{\mathbf{q}}(t)}{dt} = \boldsymbol{\Pi}\mathbf{f}(\tilde{\mathbf{q}}(t), t) + \boldsymbol{\Pi}\left[\mathbf{f}(\mathbf{q}(t), t) - \mathbf{f}(\tilde{\mathbf{q}}(t), t)\right] \tag{3.40}$$

Thus, the term $\boldsymbol{\Pi}\left[\mathbf{f}(\mathbf{q}(t), t) - \mathbf{f}(\tilde{\mathbf{q}}(t), t)\right]$ represents the impact of the unresolved modes on the resolved modes, also known as the sub-scale terms. If not accounted for, this term contributes directly to the evolution of the parallel error.

3.5.1 Stability

Rewriting the RHS in Eq. 3.36 as

$$\frac{d\boldsymbol{\epsilon}_\parallel}{dt} = \boldsymbol{\Pi}\left[\mathbf{f}(\tilde{\mathbf{q}}(t) + \boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t) + \boldsymbol{\epsilon}_\parallel(t), t) - \mathbf{f}(\tilde{\mathbf{q}}(t))\right], \tag{3.41}$$

it is possible to get additional insight.

Linear systems

Let's consider an autonomous linear system $\mathbf{f}(\mathbf{q}(t), t) = \mathbf{A}\mathbf{q}(t)$, then

$$\frac{d\boldsymbol{\epsilon}_\parallel}{dt} = \boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_\parallel(t) + \boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t). \tag{3.42}$$

Thus, even in a linear problem, the in-plane error can be continually forced by the projection error.

If we examine stability

$$\begin{aligned}\boldsymbol{\epsilon}_{\parallel}^T \frac{d\boldsymbol{\epsilon}_{\parallel}}{dt} &= \boldsymbol{\epsilon}_{\parallel}^T \boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\parallel} + \boldsymbol{\epsilon}_{\parallel}^T \boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\boldsymbol{\Pi}} && (3.43)\\ \frac{1}{2}\frac{d\boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\epsilon}_{\parallel}}{dt} &= \frac{1}{2}\boldsymbol{\epsilon}_{\parallel}^T[\boldsymbol{\Pi}\mathbf{A} + [\boldsymbol{\Pi}\mathbf{A}]^T]\boldsymbol{\epsilon}_{\parallel} + \boldsymbol{\epsilon}_{\parallel}^T \boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\perp} && (3.44)\end{aligned}$$

we get the necessary condition [2], that $\boldsymbol{\Pi}\mathbf{A} + [\boldsymbol{\Pi}\mathbf{A}]^T$ should be negative definite. Additionally, the interaction between the parallel and orthogonal errors may also affect stability in a profound manner.

In Galerkin ROMs, we do not have a great degree of control over $\boldsymbol{\Pi}$. Petrov Galerkin methods give us additional control knobs to improve both accuracy and stability.

Non-linear systems

Let's linearize Eq. 3.41:

$$\begin{aligned}\frac{d\boldsymbol{\epsilon}_{\parallel}}{dt} &\approx \boldsymbol{\Pi}\left[\mathbf{f}(\tilde{\mathbf{q}}(t)) + \left[\frac{\partial \mathbf{f}(\tilde{\mathbf{q}})}{\partial \mathbf{q}}\right][\boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t) + \boldsymbol{\epsilon}_{\parallel}(t)] - \mathbf{f}(\tilde{\mathbf{q}}(t))\right] && (3.45)\\ &= \boldsymbol{\Pi}\left[\frac{\partial \mathbf{f}(\tilde{\mathbf{q}})}{\partial \tilde{\mathbf{q}}}\right][\boldsymbol{\epsilon}_{\boldsymbol{\Pi}}(t) + \boldsymbol{\epsilon}_{\parallel}(t)] && (3.46)\end{aligned}$$

Thus,

$$\frac{1}{2}\frac{d\boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\epsilon}_{\parallel}}{dt} \approx \boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\Pi}\left[\frac{\partial \mathbf{f}(\tilde{\mathbf{q}})}{\partial \mathbf{q}}\right]\boldsymbol{\epsilon}_{\parallel} + \boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\Pi}\left[\frac{\partial \mathbf{f}(\tilde{\mathbf{q}})}{\partial \mathbf{q}}\right]\boldsymbol{\epsilon}_{\boldsymbol{\Pi}} \qquad (3.47)$$

3.6 Off-line/On-line costs in ROMs

Even though the idea of projection-based ROMs is the same for both linear and non-linear problems, we will see that non-linear problems require special treatment if we are to reduce on-line costs.

3.6.1 Linear systems

Consider a linear system

$$\frac{d\mathbf{q}(t)}{dt} = \mathbf{A}\mathbf{q}(t) + \mathbf{B}\mathbf{u}(t) \;\; ; \;\; \mathbf{q}(0) = \mathbf{q}_0, \qquad (3.48)$$

where $\mathbf{A} \in \mathbb{R}^{n\times n}$ and $\mathbf{B} \in \mathbb{R}^{n\times p}$ and $\mathbf{u} \in \mathbb{R}^p$.

[2] $\boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\parallel} = \frac{1}{2}\boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\parallel} + \frac{1}{2}\left[[\boldsymbol{\Pi}\mathbf{A}]^T\boldsymbol{\epsilon}_{\parallel}\right]^T\boldsymbol{\epsilon}_{\parallel} = \frac{1}{2}\boldsymbol{\epsilon}_{\parallel}^T\boldsymbol{\Pi}\mathbf{A}\boldsymbol{\epsilon}_{\parallel} + \frac{1}{2}\boldsymbol{\epsilon}_{\parallel}^T[\boldsymbol{\Pi}\mathbf{A}]^T\boldsymbol{\epsilon}_{\parallel} = \frac{1}{2}\boldsymbol{\epsilon}_{\parallel}^T\left[\boldsymbol{\Pi}\mathbf{A} + [\boldsymbol{\Pi}\mathbf{A}]^T\right]\boldsymbol{\epsilon}_{\parallel}$

Then the ROM that solves of $\mathbf{q}_r \in \mathbb{R}^k$ is

$$\frac{d\mathbf{q}_r(t)}{dt} = \mathbf{A}_r\mathbf{q}_r(t) + \mathbf{B}_r\mathbf{u}(t) \;\; ; \;\; \mathbf{q}_r(0) = \mathbf{V}\mathbf{q}_0, \tag{3.49}$$

where

$$\mathbf{A}_r \;=\; [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{A}\mathbf{V} \in \mathbb{R}^{k\times k} \tag{3.50}$$

$$\mathbf{B}_r \;=\; [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{B} \in \mathbb{R}^{k\times p} \tag{3.51}$$

If the reduced basis representation is effective (i.e. $k << p$ provides good accuracy), it is easy to see that the ROM will be effective as the matrices $\mathbf{A}_r$ and $\mathbf{B}_r$ can be pre-computed (off-line).

3.6.2 Non-linear systems & sparse sampling

In a non-linear system, the ROM equations are

$$\frac{d\mathbf{q}_r(t)}{dt} = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t), \;\; \mathbf{q}_r(0) = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{q}_0. \tag{3.52}$$

so even though the term $[\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t) \in \mathbb{R}^k$, it involves the computation of $\mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t) \in \mathbb{R}^n$. This defeats the purpose of ROMs as the on-line cost scales as $O(n)$. So we have to come up with a way of reducing the complexity of the non-linear ROM. One way of doing this is to sample the residual in $O(k)$ locations and reconstruct the function elsewhere, as required. Fortunately, this sounds very much like the technique we learnt in the sensors chapter.

To do this, we can represent snapshots of the non-linear function $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^n$ in terms of some basis $\boldsymbol{\Psi} \in \mathbb{R}^{n\times s}$

$$\mathbf{f} = \Psi\mathbf{a},$$

where $\mathbf{a} \in \mathbb{R}^s$ are the basis coefficients. These basis functions, can, for instance be constructed using POD on a collection of snapshots

$$\mathbf{F} \in \mathbb{R}^{n\times m} = [\mathbf{f}(\mathbf{x}(t_1)) \;\; \mathbf{f}(\mathbf{x}(t_2)) \;\; \mathbf{f}(\mathbf{x}(t_3))...\mathbf{f}(\mathbf{x}(t_m)).$$

Then, we can sub-sample $\mathbf{f}$ by multiplying it by a sample selection matrix $\mathbf{P} \in \mathbb{R}^{s\times n}$.

Then we have a few measurements $\mathbf{P}\mathbf{f} = \mathbf{f}_s \in \mathbb{R}^s$, for which

$$\mathbf{P}\mathbf{f} = \mathbf{P}\Psi\mathbf{a}.$$

Now, based on the few samples, we can estimate $\hat{\mathbf{a}}$ using

$$\hat{\mathbf{a}} = [\mathbf{P}\Psi]^{+}\mathbf{f}_s.$$

and we can reconstruct the entire $\mathbf{f}$ as

$$\hat{\mathbf{f}} = \Psi[\mathbf{P}\Psi]^{+}\mathbf{f}_s.$$

One other way to look at this is to write it as an oblique projection

$$\hat{\mathbf{f}} = \mathbf{\Pi}_s \mathbf{f},$$

where [3] $\mathbf{\Pi}_s = \mathbf{V}_s[\mathbf{W}_s^T \mathbf{V}_s]^{-1}\mathbf{W}_s^T$, with $\mathbf{V}_s = \mathbf{\Psi}$ and $\mathbf{W}_s = \mathbf{P}^T$.

Of course, there are a number of techniques for sample selection that we have encountered, such as DEIM and compressed sensing.

Then, the required term in eqn. 3.52 is

$$\begin{aligned} \frac{d\mathbf{q}_r(t)}{dt} &= [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\Psi[\mathbf{P}\Psi]^{+}\mathbf{f}_s(t) && (3.53) \\ &= \mathbf{S}\mathbf{f}_s(t), && (3.54) \end{aligned}$$

where $\mathbf{S} \in \mathbb{R}^{k\times s} = [\mathbf{W}^T\mathbf{V}]^{-1}\mathbf{W}^T\Psi[\mathbf{P}\Psi]^{+}$ can be pre-computed.

Hence, even though the off-line costs are higher (as we have to compute a new basis $\mathbf{\Psi}$), the on-line cost scales as $O(k)$.

3.7 ROMs using Residual Minimization

One way of thinking about Petrov-Galerkin projections is to look at ROMs from a residual minimization standpoint. Let's start with fully discrete schemes. For instance, with the Euler explicit time-stepping scheme:

$$\frac{\mathbf{V}\mathbf{q}_r^n - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} = \mathbf{f}(\mathbf{V}\mathbf{q}_r^{n-1}).$$

Let's define a residual

$$\mathbf{r}(\mathbf{q}_r^n) = \frac{\mathbf{V}\mathbf{q}_r^n - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} - \mathbf{f}(\mathbf{V}\mathbf{q}_r^{n-1}).$$

In an Euler implicit scheme,

$$\mathbf{r}(\mathbf{q}_r^n) = \frac{\mathbf{V}\mathbf{q}_r^n - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} - \mathbf{f}(\mathbf{V}\mathbf{q}_r^{n}).$$

Then in a residual minimization approach, the update to the next time step is

$$\mathbf{q}_r^n = arg \min_{\hat{\mathbf{q}}_r \in range(\mathbf{V})} \|\mathbf{r}(\hat{\mathbf{q}}_r)\|_2^2. \qquad (3.55)$$

We can use the generalization of the Newton's method - the Gauss Newton method to solve this problem. Defining the function to be minimized as

$$m(\mathbf{q}_r) = [\mathbf{r}(\mathbf{q}_r^n)]^T[\mathbf{r}(\mathbf{q}_r^n)].$$

We are looking for $\nabla m(\mathbf{q}_r) = 0$, and hence

[3] $s = k$ is assumed

$$\left[\frac{\partial \mathbf{r}(\mathbf{q}_r^n)}{\partial \mathbf{q}_r^n}\right]^T [\mathbf{r}(\mathbf{q}_r^n)] = 0. \tag{3.56}$$

For the Euler explicit scheme,

$$\left[\frac{\partial \mathbf{r}(\mathbf{q}_r^n)}{\partial \mathbf{q}_r^n}\right] = \frac{\mathbf{V}}{\Delta t}$$

and so the discretized scheme looks like a simple Galerkin ROM (Euler explicit discretization of Eq. 3.6).

$$\mathbf{q}_r^n - \mathbf{q}_r^{n-1} = \Delta t \mathbf{V}^T \mathbf{f}(\mathbf{V}\mathbf{q}_r^{n-1}). \tag{3.57}$$

For the Euler implicit scheme,

$$\left[\frac{\partial \mathbf{r}(\mathbf{q}_r^n)}{\partial \mathbf{q}_r^n}\right] = \frac{\mathbf{V}}{\Delta t} - \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\mathbf{V},$$

and therefore we have

$$\left[\frac{\mathbf{V}^T}{\Delta t} - \mathbf{V}^T\mathbf{J}^T(\mathbf{q}_r^n)\right]\left[\frac{\mathbf{V}\mathbf{q}_r^n - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} - \mathbf{f}(\mathbf{V}\mathbf{q}_r^n)\right] \quad = \quad 0, \tag{3.58}$$

where $\mathbf{J}(\mathbf{q}_r^n) = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]_{\mathbf{q}=\mathbf{V}\mathbf{q}_r^n}$.

Therefore this can be viewed as a Petrov-Galerkin projection with $\mathbf{W}^n \in \mathbb{R}^{n\times k} = \frac{\mathbf{V}}{\Delta t} - \mathbf{J}(\mathbf{q}_r^n)\mathbf{V}$.

Such projections can be determined for other time integrators also. Thus, the LSPG (Least Squares Petrov-Galerkin) ROM is equivalent to applying a Petrov-Galerkin projection to the FOM ODE.

It can be proved formally that if the reduced basis is orthogonal, then the Galerkin ROM (Eq. 3.6) is continuous optimal in the sense that the approximated velocity minimizes the 2-norm of the FOM ODE residual (Eq. 3.1) over the range of $\mathbf{V}$. i.e.,

$$\frac{d\mathbf{q}(t)}{dt} = arg \min_{\mathbf{g} \in range(\mathbf{V})} ||\mathbf{g} - \mathbf{f}(\mathbf{V}\mathbf{q}_r(t), t)||_2 \tag{3.59}$$

Due to the above optimality property of the Galerkin ROM, adding vectors to the trial basis which enriches the trial subspace $range(\mathbf{V})$ results in a monotonic decrease in the minimum-residual objective function in Eq. 3.59, which is simply the 2-norm of the FOM ODE residual. This implies that the 2-norm of the error in the ROM $\frac{d\mathbf{q}(t)}{dt}$ will monotonically decrease as the trial subspace is enriched.
On the other hand, enriching the discrete residual minimization (Eq. 3.55) with more bases will ensure a monotonic decrease in the discrete FOM residual.

A more general LSPG ROM can be defined if we frame Eq. 3.55 as

$$\mathbf{q}_r^n = arg \min_{\hat{\mathbf{q}}_r \in range(\mathbf{V})} \|\mathbf{A}\mathbf{r}(\hat{\mathbf{q}}_r)\|_2^2. \tag{3.60}$$

This is equivalent to a Petrov Galerkin projection,

$$\mathbf{W}(\mathbf{q}_r^n)^T[\mathbf{r}(\mathbf{q}_r^n)] = 0, \tag{3.61}$$

where

$$\mathbf{W}(\mathbf{q}_r^n) = \left[\mathbf{A}^T\mathbf{A}\frac{\partial \mathbf{r}(\mathbf{q}_r^n)}{\partial \mathbf{q}_r^n}\right] = \left[\mathbf{A}^T\mathbf{A}\frac{\partial \mathbf{r}(\mathbf{q}^n)}{\partial \mathbf{q}^n}\right]\mathbf{V}. \tag{3.62}$$

Thus, LSPG is the same as Galerkin (and Galerkin is discrete optimal) if
- $\Delta t \to 0$
- Scheme is explicit ($\mathbf{A}$ can be used to normalize)
- $\mathbf{A}^T\mathbf{A} = \left[\frac{\partial \mathbf{r}(\mathbf{q})}{\partial \mathbf{q}}\right]^{-1}$

For the standard LSPG model $\mathbf{A} = \mathbf{I}$ and for the sparse version (GNAT) $\mathbf{A} = [\mathbf{P}\mathbf{\Phi}]^+\mathbf{P}$ (see next section).

It can be shown that the Backward Euler-LSPG gives a smaller error than Backward Euler-Galerkin.

3.7.1 Solving the Backward Euler-LSPG system

Let's write Eq. 3.58 as

$$[\mathbf{W}^n]^T\left[\frac{\mathbf{V}\mathbf{q}_r^n - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t}\right] = [\mathbf{W}^n]^T\mathbf{f}(\mathbf{V}\mathbf{q}_r^n).$$

To solve this at every physical time step, let's introduce a pseudo time step iteration p. When $p = 0$, $\mathbf{q}_r^p = \mathbf{q}_r^{n-1}$.

$$\begin{aligned}
\left[\mathbf{W}^{p-1}\right]^T \left[\frac{\mathbf{V}\mathbf{q}_r^p - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t}\right] &= \left[\mathbf{W}^{p-1}\right]^T \left[\mathbf{f}(\mathbf{V}\mathbf{q}_r^{p-1}) + \mathbf{J}(\mathbf{q}_r^{p-1})\mathbf{V}(\mathbf{q}_r^p - \mathbf{q}_r^{p-1})\right] \\
\left[\mathbf{W}^{p-1}\right]^T \left[\frac{\mathbf{V}\mathbf{q}_r^p - \mathbf{V}\mathbf{q}_r^{p-1}}{\Delta t}\right] &= \left[\mathbf{W}^{p-1}\right]^T \left[-\frac{\mathbf{V}\mathbf{q}_r^{p-1} - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} + \mathbf{f}(\mathbf{V}\mathbf{q}_r^{p-1}) + \mathbf{J}(\mathbf{q}_r^{p-1})\mathbf{V}(\mathbf{q}_r^p - \mathbf{q}_r^{p-1})\right] \\
\left[\mathbf{W}^{p-1}\right]^T \left[\frac{\mathbf{V}\mathbf{q}_r^p - \mathbf{V}\mathbf{q}_r^{p-1}}{\Delta t} - \mathbf{J}(\mathbf{q}_r^{p-1})\mathbf{V}(\mathbf{q}_r^p - \mathbf{q}_r^{p-1})\right] &= \left[\mathbf{W}^{p-1}\right]^T \left[-\frac{\mathbf{V}\mathbf{q}_r^{p-1} - \mathbf{V}\mathbf{q}_r^{n-1}}{\Delta t} + \mathbf{f}(\mathbf{V}\mathbf{q}_r^{p-1})\right] \\
\left[\mathbf{W}^{p-1}\right]^T \mathbf{W}^{p-1}(\mathbf{q}_r^p - \mathbf{q}_r^{p-1}) &= -\left[\mathbf{W}^{p-1}\right]^T \mathbf{r}(\mathbf{q}_r^{p-1})
\end{aligned}$$

Thus, the Backward Euler-LSPG update is:

$$\mathbf{q}_r^p = \mathbf{q}_r^{p-1} - \mathbf{W}^{(p-1)^{+}}\mathbf{r}(\mathbf{q}_r^{p-1}). \tag{3.63}$$

NOTE: The Backward Euler-Galerkin update would be:

$$\mathbf{q}_r^p = \mathbf{q}_r^{p-1} - [\mathbf{V}^T\mathbf{W}^{(p-1)}]^{-1}\mathbf{V}^T\mathbf{r}(\mathbf{q}_r^{p-1}). \tag{3.64}$$

There is another way of getting to Eqn. 3.63. Starting from Eq. 3.56, and applying Newton's method, we have

$$\frac{\partial}{\partial \mathbf{q}_r^{p-1}}\left[\left[\mathbf{W}^{p-1}\right]^T \mathbf{r}(\mathbf{q}_r^{p-1})\right](\mathbf{q}_r^p - \mathbf{q}_r^{p-1}) = -\left[\mathbf{W}^{p-1}\right]^T [\mathbf{r}(\mathbf{q}_r^{p-1})]. \tag{3.65}$$

Using the Gauss-Newton assumption $\frac{\partial}{\partial \mathbf{q}_r^{p-1}}\left[\left[\mathbf{W}^{p-1}\right]^T \mathbf{r}(\mathbf{q}_r^{p-1})\right] \approx \left[\mathbf{W}^{p-1}\right]^T \mathbf{W}^{p-1}$, we have

$$\left[\mathbf{W}^{p-1}\right]^T \mathbf{W}^{p-1}(\mathbf{q}_r^p - \mathbf{q}_r^{p-1}) = -\left[\mathbf{W}^{p-1}\right]^T \mathbf{r}(\mathbf{q}_r^{p-1}). \tag{3.66}$$

3.7.2 Sparse sampling and residual minimization

The goal of Eqn. 3.55 is to determine $\mathbf{q}_r^n$ by minimizing the residual for the entire state $\mathbf{V}\mathbf{q}_r^n$. Instead, we could aim to minimize the residual at s selected locations $\mathbf{P}\mathbf{r}(\mathbf{q}_r^n) = \mathbf{r}_s(\mathbf{q}_r^n) \in \mathbb{R}^s$. So the goal could be:

$$\mathbf{q}_r^n = arg \;\; min\|\mathbf{P}\mathbf{r}(\mathbf{q}_r^n)\|_2^2 = arg \;\; min\|\mathbf{r}_s(\mathbf{q}_r^n)\|_2^2. \tag{3.67}$$

This could be enabled, for instance, by doing some offline work to build a basis $\mathbf{\Phi}$ by considering a snapshot matrix

$$\mathbf{R} = [\mathbf{r}(\mathbf{q}^1) \;\; \mathbf{r}(\mathbf{q}^2) \;\; ....\mathbf{r}(\mathbf{q}^m)],$$

and thus, the minimization problem can be posed as [4]

$$\mathbf{q}_r^n = arg \;\; min\|\mathbf{\Phi}[\mathbf{P\Phi}]^+\mathbf{Pr}(\mathbf{q}_r^n)\|_2^2 = arg \;\; min\|[\mathbf{P\Phi}]^+\mathbf{Pr}(\mathbf{q}_r^n)\|_2^2 = arg \;\; min\|\mathbf{BPr}(\mathbf{q}_r^n)\|_2^2. \tag{3.68}$$

where $\mathbf{B} \in \mathbb{R}^{s\times s} = [\mathbf{P\Phi}]^+$ is a pre-computed matrix.

If we consider Euler implicit, then we have

$$\left[\mathbf{BP}\frac{\mathbf{V}}{\Delta t} - \mathbf{BPJ}(\mathbf{q}_r^n)\mathbf{V}\right]^T [\mathbf{BPr}(\mathbf{q}_r^n)] \;\; = \;\; 0. \tag{3.69}$$

$$\tag{3.70}$$

Defining

$$\mathbf{W} \in \mathbb{R}^{s\times k} = \left[\mathbf{BP}\frac{\mathbf{V}}{\Delta t} - \mathbf{BPJ}(\mathbf{q}_r^n)\mathbf{V}\right] = \left[\mathbf{B}\frac{\mathbf{V}_s}{\Delta t} - \mathbf{BJ}_s(\mathbf{q}_r^n)\mathbf{V}\right],$$

We have our ROM

$$\mathbf{W}^T\mathbf{BV}_s\left[\frac{\mathbf{q}_r^n - \mathbf{q}_r^{n-1}}{\Delta t}\right] = \mathbf{W}^T\mathbf{Bf}_s(\mathbf{Vq}_r^n).$$

To solve this, we can use

$$\mathbf{q}_r^p \;\; = \;\; \mathbf{q}_r^{p-1} - \mathbf{W}^+\mathbf{Br}_s(\mathbf{q}_r^{p-1}). \tag{3.71}$$

3.8 Non-intrusive Reduced Order Models

In this section, we will introduce the non-intrusive ROM technique, which is a purely data-driven technique and does not use the governing equations.

The first step is to find a low-dimensional space and this step can be the same as in the intrusive ROM. For instance, one can use POD to extract a low-dimensional manifold. For one quantity of interest, e.g. a velocity component in our case, defining a set of POD projection bases with k modes $\mathbf{V} \in \mathbb{R}^{n\times k}$ spanning a subspace $\mathcal{V} \subset \mathbb{R}^n$, and a complementary basis $\mathbf{V}_\perp$ spanning $\mathcal{V}_\perp$, such that $\mathcal{V} \oplus \mathcal{V}_\perp = \mathbb{R}^n$, then the following decomposition can be derived:

$$\mathbf{q}(t) = \mathbf{V}\mathbf{q}_r(t) + \mathbf{V}_\perp\mathbf{q}_\perp(t) \tag{3.72}$$

By discarding the low-energy modes in $\mathbf{V}_\perp$, a reduced order approximation to the quantity of interest on the reduced space $\mathcal{V}$ is given by

$$\mathbf{q}(t) \approx \mathbf{V}\mathbf{q}_r(t). \tag{3.73}$$

In contrast to the intrusive ROM in which $\mathbf{q}_r(t)$ is computed based on the information from the governing equation of th full order models, in the proposed method the dynamics of $\mathbf{q}_r(t)$ is discovered using a data-driven approach as in Chapters 3 or 4. For instance, if a

[4] $\mathbf{\Phi}$ drops out of the norm because it is orthogonal

neural network is used and time is discretized, the evolution of the dynamical system from time step n to $n+1$ is

$$\mathbf{q}_r^{n+1} = \boldsymbol{\sigma}(\boldsymbol{\Theta}^3\boldsymbol{\sigma}(\boldsymbol{\Theta}^2\boldsymbol{\sigma}(\boldsymbol{\Theta}^1\mathbf{q}_r^n + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3). \tag{3.74}$$

To recover the full field variable at any time, we can use

$$\mathbf{q}^{n+1} = \mathbf{V}\mathbf{q}_r^{n+1}.$$

Note that we have just scratched the surface of non-intrusive ROMs thus far. There are many combinations possible. As an example, check out the following paper: *Multi-level Convolutional Autoencoder Networks for Parametric Prediction of Spatio-temporal Dynamics,* by J. Xu, K. Duraisamy, arXiv:1912.11114, 2019.

3.9 Eigensystem Realization Algorithm

The eigensystem realization algorithm (ERA) is a system identification method that has been used for non-intrusive balancing transformation. Unlike the BPOD method, ERA only relies on the direct system impulse response and bypasses the adjoint system simulations that are expensive to compute in systems with a large number of outputs (e.g., full-state) and inaccessible in experiments. Another advantage of ERA is that it does not require explicit computation of the balancing transformation matrix and therefore, it does not entail the challenges that the analytical BT method faces in highly stiff systems.

For balancing transformation with ERA, first the Hankel matrix is assembled with the impulse response of the direct system,

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \dots & \mathbf{h}_{m_p} \\ \mathbf{h}_2 & \mathbf{h}_3 & \dots & \mathbf{h}_{m_p+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{m_o} & \mathbf{h}_{m_o+1} & \dots & \mathbf{h}_{m_o+m_p+1} \end{bmatrix} = \mathscr{O}\mathscr{P}. \tag{3.75}$$

Next, a shifted Hankel matrix is assembled by advancing the sequence of Markov parameters one step in time,

$$\mathbf{H}' = \begin{bmatrix} \mathbf{h}_2 & \mathbf{h}_3 & \dots & \mathbf{h}_{m_p+1} \\ \mathbf{h}_3 & \mathbf{h}_4 & \dots & \mathbf{h}_{m_p+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{m_o+1} & \mathbf{h}_{m_o+2} & \dots & \mathbf{h}_{m_o+m_p+2} \end{bmatrix} = \mathscr{O}\mathbf{A}\mathscr{P}. \tag{3.76}$$

The shifted Hankel matrix enables us to balance the Gramians without the need for adjoint system simulations. Using this matrix, and SVD of the Hankel matrix in Eqn 3.23, the balanced ROM matrices can be directly computed as,

$$\begin{aligned}\mathbf{A}_r &= \mathbf{\Sigma}_r^{-1/2}\mathbf{U}_r^*\mathbf{H}'\mathbf{V}_r\mathbf{\Sigma}_r^{-1/2}\\ \mathbf{B}_r &= \mathbf{\Sigma}_r^{1/2}\mathbf{V}_r^*\begin{bmatrix}\mathbf{I}_m & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix},\\ \mathbf{C}_r &= \begin{bmatrix}\mathbf{I}_p & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{U}_r\mathbf{\Sigma}_r^{1/2},\end{aligned} \tag{3.77}$$

where, m is the number of inputs and p is the number of outputs. The following discrete-time r-dimensional system is the balanced ROM identified by ERA,

$$\begin{aligned}\mathbf{x}_r &= \mathbf{A}_r\mathbf{x}_r + \mathbf{B}_r\mathbf{u},\\ \mathbf{y} &= \mathbf{C}_r\mathbf{x}_r.\end{aligned} \tag{3.78}$$

Using the direct system impulse response we can compute the direct balancing modes with Eqn 3.24 and reconstruct the FOM state.

Note that the ROM identified by ERA is balanced only if the impulse response snapshots are collected with a high enough frequency and for a long enough time, otherwise, Gramians of the transformed system are not equal and diagonal and the theoretical error bounds are not satisfied. In fact, sensitivity to the sampling properties is a critical aspect of data-driven methods like ERA that needs to be carefully addressed when using these methods. In practice, impulse response snapshots need to be collected until the sequence of Markov parameters captures the decay in lightly damped structures.

3.10 References

This chapter draws heavily from:

1. M Rathinam, LR Petzold, A new look at proper orthogonal decomposition, SIAM Journal on Numerical Analysis 41 (5), 1893-1925, 2004.

2. D. Amsallem, Interpolation on manifolds of CFD-based fluid and FEM-based structural ROMs for on-line aeroelastic predictions, Stanford PhD thesis, 2010.

3. K Carlberg, M Barone, H Antil, Galerkin v. Least-squares Petrov-Galerkin projection in nonlinear model reduction, Journal of Computational Physics 330, 693–734, 2016.

4. Antoulas, A.C., 2005. Approximation of large-scale dynamical systems. Society for Industrial and Applied Mathematics.

5. Benner, P., Ohlberger, M., Cohen, A. and Willcox, K. eds., 2017. Model reduction and approximation: theory and algorithms. Society for Industrial and Applied Mathematics.

6. Brunton, S.L. and Kutz, J.N., 2019. Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press.

7. Rowley, C.W., 2005. Model reduction for fluids, using balanced proper orthogonal decomposition. International Journal of Bifurcation and Chaos, 15(03), pp.997-1013.

8. Ma, Z., Ahuja, S. and Rowley, C.W., 2011. Reduced-order models for control of fluids using the eigensystem realization algorithm. Theoretical and Computational Fluid Dynamics, 25(1), pp.233-247.

9. Willcox, K. and Peraire, J., 2002. Balanced model reduction via the proper orthogonal decomposition. AIAA journal, 40(11), pp.2323-2330.

Chapter 4

Appendix: Basic Linear Algebra

4.1 Notations for vectors and matrices

The notation will be:
Scalars: Normal fonts (example: λ)
Vector: Bold-face (example: $\mathbf{e}$). An element of a vector will be denoted by a subscript (example: e_i). Note that there is no bold face for the element.
Matrix: Bold-face and capital letters (example: $\mathbf{A}$). An element of a matrix will be denoted by two subscripts (example a_{ij}). A column of a matrix will be denoted by a subscript (example: $\mathbf{a}_i$).
Transpose and Complex conjugate transpose: $\mathbf{A}^T$ represents the transpose of a real matrix and $\mathbf{A}^*$ represents the complex conjugate transpose (or the Hermitian transpose).
Note: $[\mathbf{AB}]^* = \mathbf{B}^*\mathbf{A}^*$
When $\mathbf{A}$ and $\mathbf{B}$ are non-singular matrices,

$$[\mathbf{AB}]^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1} \;\; ; \; [\mathbf{A}^*]^{-1} = [\mathbf{A}^{-1}]^* = \mathbf{A}^{-*}.$$

4.2 Some definitions

Consider a set of vectors $\mathbf{v}_1, \mathbf{v}_2, ...\mathbf{v}_n$, where each $\mathbf{v}_i \in \mathbb{C}^m$.
Linear independence: These vectors are linearly independent if $\sum_i \alpha_i \mathbf{v}_i = 0$, only when $\alpha_i = 0$.
Span: The span of the collection of these vectors is the set of all possible linear combinations and can define a subspace V.
Basis: A set of linearly independent vectors whose span is a subspace V can represent a basis for that subspace.

Consider a matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$.
Rank of $\mathbf{A}$: Maximum number of linearly independent rows or columns of $\mathbf{A}$. Note that the row rank and column rank are both the same.

If $rank(\mathbf{A}) = min(m, n)$, $\mathbf{A}$ has full rank.
If $rank(\mathbf{A}) = m$, $\mathbf{A}$ has full row rank.
If $rank(\mathbf{A}) = n$, $\mathbf{A}$ has full column rank.
If $rank(\mathbf{A}) < min(m, n)$, $\mathbf{A}$ is rank-deficient.
$rank(\mathbf{A}) = rank(\mathbf{A}^*)$.

Range of $\mathbf{A}$: Set of all linear combinations of its columns $range(\mathbf{A}) = \mathbf{Ax} : \mathbf{x} \in \mathbb{C}^n$. $rank(\mathbf{A}) = dim(range(\mathbf{A}))$.
Null space or Kernel of $\mathbf{A}$: Set of vectors annihilated by $\mathbf{A}$. $null(\mathbf{A}) = \{\mathbf{x} \in \mathbb{C}^n : \mathbf{Ax} = \mathbf{0}\}$.
The four fundamental subspaces of a matrix $\mathbf{A}$:

The column space of $\mathbf{A}$ is the range of $\mathbf{A}$.
The null space of $\mathbf{A}$ is its kernel.
The row space of $\mathbf{A}$ is the range of $\mathbf{A^T}$.
The left null space of $\mathbf{A}$ is the kernel of $\mathbf{A}^T$.

The fundamental theorem of Linear Algebra:

- $dim(range(\mathbf{A})) + dim(null(\mathbf{A})) = n.$

  Intuition: $r + (n - r) = n$.

- $dim(range(\mathbf{A}^T)) + dim(null(\mathbf{A}^T)) = m.$

  Intuition: $r + (m - r) = m$.

- $dim(range(\mathbf{A})) = dim(range(\mathbf{A}^T)).$

  Intuition: column rank = row rank.

- $null(\mathbf{A}) \perp range(\mathbf{A}^T).$

  Intuition: $\mathbf{A}\mathbf{x}_{null} = \mathbf{0}$ means every row of $\mathbf{A}$ is orthogonal to $\mathbf{x}_{null}$.

- $null(\mathbf{A}^T) \perp range(\mathbf{A}).$

  Intuition: $\mathbf{A}^T\mathbf{y}_{null} = \mathbf{0}$ means every column of $\mathbf{A}$ is orthogonal to $\mathbf{y}_{null}$.

If we have a non-trivial null-space, then a linear system spanned by $\mathbf{A}$ has a non-unique solution.

Consider square matrices $\mathbf{A} \in \mathbb{C}^{m\times m}$.
Unitary/Orthogonal Matrix : $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^* = \mathbf{I}$. If $\mathbf{A} \in \mathbb{R}^{m\times m}$, then a unitary matrix is referred to as an Orthogonal Matrix.

Normal Matrix : $\mathbf{A}^*\mathbf{A} = \mathbf{A}\mathbf{A}^*$. Note: Normal matrices include Hermitian, skew-Hermitian and Unitary matrices.
Trace of $\mathbf{A}$: Sum of its diagonal elements (only defined for square matrix).

4.3 Matrix and Vector Norms

Given a scalar x its absolute value $|x|$ is a representation of length. To represent the "length" of a vector or a matrix and to define the convergence of sequences of matrices or vectors, we use the notion of a norm.

For a vector of dimension N, an L-p Norm is defined as:

$$\|\mathbf{x}\|_p = \left[\sum_{j=1}^{N} |x_j|^p\right]^{1/p} \tag{4.1}$$

Note: The L-2 Norm is nothing but the Euclidean distance.
Note: As a special case, the $\mathrm{L} - \infty$ Norm is

$$\|\mathbf{x}\|_\infty = \max\{|x_1|, |x_2|, ...|x_n|\} \tag{4.2}$$

For matrices, the induced L-p Norm (or the L-p norm) is

$$\|\mathbf{A}\|_p = \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{Ax}\|_p}{\|\mathbf{x}\|_p} = \max_{\|\mathbf{x}\|_p=1} \|\mathbf{Ax}\|_p \tag{4.3}$$

Proof of the above equivalence:

$$\max_{\|x\|=1} \|Ax\| = \max_{\|x\|=1} \frac{\|Ax\|}{\|x\|} \leq \max_{x\neq 0} \frac{\|Ax\|}{\|x\|}.$$

For the reverse inequality, fix $x \neq 0$. Then

$$\frac{\|Ax\|}{\|x\|} = \left\| A\frac{x}{\|x\|} \right\| \leq \max_{\|y\|=1} \|Ay\|.$$

Thus

$$\max_{x\neq 0} \frac{\|Ax\|}{\|x\|} \leq \max_{\|y\|=1} \|Ay\|.$$

Note: As special cases (for an $M \times N$ matrix),

$$\|\mathbf{A}\|_1 = \max_{1\leq j\leq N} \sum_{i=1}^{M} |a_{ij}| \tag{4.4}$$

which is simply the maximum absolute column sum of the matrix.

$$\|\mathbf{A}\|_\infty = \max_{1\leq i\leq M} \sum_{j=1}^{N} |a_{ij}| \tag{4.5}$$

which is simply the maximum absolute row sum of the matrix.

Another useful norm is the Frobenius norm

$$\|\mathbf{A}\|_F = \left[\sum_{i=1}^{M}\sum_{j=1}^{N} |a_{ij}|^2\right]^{1/2} = trace(\mathbf{A}^*\mathbf{A})^{1/2}. \tag{4.6}$$

4.3.1 Norm inequalities

The expressions below follow directly from the definition of norms

$$\begin{aligned}
\|\mathbf{x}\| &> 0 \;\; when \;\; \mathbf{x} \neq 0 && (4.7)\\
\|\mathbf{0}\| &= 0 && (4.8)\\
\|\mathbf{x}+\mathbf{y}\| &\leq \|\mathbf{x}\| + \|\mathbf{y}\| && (4.9)\\
\|\mathbf{x}^T\mathbf{y}\| &\leq \|\mathbf{x}\|\|\mathbf{y}\| && (4.10)\\
\|\mathbf{A}\mathbf{B}\| &\leq \|\mathbf{A}\|\|\mathbf{B}\| && (4.11)\\
\|\mathbf{A}\mathbf{x}\| &\leq \|\mathbf{A}\|\|\mathbf{x}\| && (4.12)
\end{aligned}$$

The last inequality is easy to prove. Consider the definition of $\|\mathbf{A}\|$:

$$\|\mathbf{A}\| = \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{A}\mathbf{v}\|}{\|\mathbf{v}\|} \tag{4.13}$$

Now if we take any vector $\mathbf{x}$, we have

$$\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \leq \max_{\mathbf{v}\neq 0} \frac{\|\mathbf{A}\mathbf{v}\|}{\|\mathbf{v}\|} \tag{4.14}$$

Thus

$$\frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\| \tag{4.15}$$

Therefore

$$\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|\|\mathbf{x}\|. \tag{4.16}$$

4.3.2 Some basic facts about norms

1. The spectral radius of a matrix is defined as $\rho(\mathbf{A}) = \max\{|\lambda_1|, \cdots, |\lambda_n|\}$, where λ_i are the eigenvalues. This holds true for both real and complex eigenvalues.
2. For any matrix and matrix norm, the spectral radius satisfies $\rho(\mathbf{A}) \leq \|\mathbf{A}\|_p$.
3. For a normal matrix, the spectral radius satisfies $\rho(\mathbf{A}) = \|\mathbf{A}\|_2$.
4. If $\mathbf{U}$ is a unitary matrix, $\|\mathbf{U}\mathbf{x}\| = \|\mathbf{x}\|$.

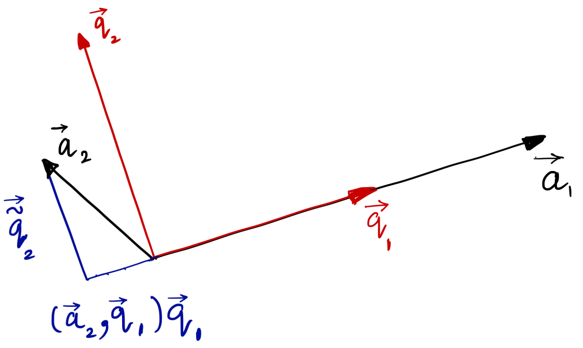


Figure 4.1: Gram-schmidt procedure illustrated for 2 vectors in $\mathbb{R}^2$.

4.4 Gram-Schmidt Orthonormalization

Given a linearly independent set of vectors $S \equiv \{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, ...\mathbf{a}_n\}$, where $\mathbf{a}_i \in \mathbb{R}^m$, the Gram-Schmidt procedure finds an orthonormal set of vectors $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, ...\mathbf{q}_n\}$ that spans the same subspace as S. The procedure is as follows:

1. $\tilde{\mathbf{q}}_1 = \mathbf{a}_1$
2. $\mathbf{q}_1 = \frac{\tilde{\mathbf{q}}_1}{\|\tilde{\mathbf{q}}_1\|}$
3. $\tilde{\mathbf{q}}_2 = \mathbf{a}_2 - (\mathbf{q}_1^T \mathbf{a}_2)\mathbf{q}_1$
4. $\mathbf{q}_2 = \frac{\tilde{\mathbf{q}}_2}{\|\tilde{\mathbf{q}}_2\|}$
5. $\tilde{\mathbf{q}}_3 = \mathbf{a}_3 - (\mathbf{q}_1^T \mathbf{a}_3)\mathbf{q}_1 - (\mathbf{q}_2^T \mathbf{a}_3)\mathbf{q}_2$
6. $\mathbf{q}_3 = \frac{\tilde{\mathbf{q}}_3}{\|\tilde{\mathbf{q}}_3\|}$
7. Repeat recursively

4.5 Eigenvalues and Eigenvectors

The eigensystem $\{\Lambda, \mathbf{S}\}$ of a square matrix $\mathbf{A} \in \mathbb{C}^{N \times N}$ can be obtained as follows: A right eigenvector $\mathbf{s}_i$ and the corresponding eigenvalue λ_i satisfy the following relationship

$$\mathbf{A}\mathbf{s}_i = \lambda_i \mathbf{s}_i$$

The eigenvalues can be obtained by solving

$$det\left[\mathbf{A} - \lambda_i \mathbf{I}\right] = 0$$

The eigenvectors can be combined to form the eigenvector matrix

$$\mathbf{S} = \left[\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, ...\mathbf{s}_N\right],$$

and the eigenvalues can be combined to form a diagonal matrix $\boldsymbol{\Lambda}$.

In matrix form, the eigenvalue-vector relationships can be written as

$$\mathbf{AS} = \mathbf{S}\boldsymbol{\Lambda}.$$

If a matrix $\mathbf{A}$ is symmetric, *i.e.*, if $\mathbf{A}^T = \mathbf{A}$, then we have the following:

• Fact: The eigenvalues of $\mathbf{A}$ are real

• Fact: Eigenvectors corresponding to distinct eigenvalues are orthogonal.

• Definition: A symmetric matrix with all positive eigenvalues is symmetric positive definite.

In the general case, a matrix is positive definite if $\mathbf{x}^*\mathbf{A}\mathbf{x} > 0$ for all non-zero $\mathbf{x}$.

A necessary and sufficient condition for the positive definiteness of a general matrix $\mathbf{A}$ is that its Hermitian part $\frac{1}{2}(\mathbf{A} + \mathbf{A}^*)$ is positive definite.

The inverse of a positive definite matrix is positive definite.

The adjoint (dual) eigenvalue problem is defined as

$$\mathbf{A}^*\hat{\mathbf{S}} = \hat{\mathbf{S}}\hat{\mathbf{\Lambda}}.$$

The primal and dual eigenvectors are related by $\mathbf{S}^T\hat{\mathbf{S}} = \mathbf{I}$.

Also, $\mathbf{A}\mathbf{A}^* = det(\mathbf{A})\mathbf{I}$.

If $\mathbf{A}$ is non-singular, then

1. Eigenvalues of $\mathbf{A}^{-1}$ are $1/\lambda_i$.
2. Eigenvalues of $\mathbf{A}^*$ are $det(\mathbf{A})/\lambda_i$.

4.6 Diagonalization

If the eigensystem is complete (*i.e.* if the eigenvectors are linearly independent), then the matrix $\mathbf{A}$ can be diagonalized in the following manner:

$$\mathbf{S}^{-1}\mathbf{A}\mathbf{S} = \mathbf{\Lambda}, \quad or \quad = \mathbf{A} = \mathbf{S}\mathbf{\Lambda}\mathbf{S}^{-1}$$

where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues *i.e.* $\Lambda_{ii} = \lambda_i$.

If $\mathbf{A}$ is a real symmetric matrix then it is diagonalized by $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, where $\mathbf{Q}$ is an orthonormal matrix.

If $\mathbf{A}$ is a normal matrix, then it is diagonalizable by a unitary matrix $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^*$.

It is not possible to diagonalize a matrix that has an incomplete or defective eigensystem. We can, however, use the Jordan decomposition

$$\mathbf{A} = \mathbf{P}\mathbf{J}\mathbf{P}^{\mathbf{-1}},$$

to block diagonalize such a system.

$$\mathbf{J} = \mathbf{P}^{\mathbf{-1}}\mathbf{A}\mathbf{P} = \begin{bmatrix} J_1 & 0 & .... & 0 \\ 0 & J_2 & .... & 0 \\ . & . & .... & . \\ . & . & .... & . \\ 0 & 0 & .... & J_K \end{bmatrix},$$

where there are K independent eigenvectors and

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & 0 & .... & 0 \\ 0 & \lambda_i & 1 & .... & 0 \\ . & . & .... & . & \\ . & . & .... & . & \\ 0 & 0 & .... & . & \lambda_i \end{bmatrix}$$

is a Jordan sub-block. A repeated root in a Jordan block is referred to as a defective eigenvalue. For each $J_i \in \mathbb{C}^{r\times r}$, associated with an eigenvalue λ_i of multiplicity r, there exists one eigenvector. The other $r-1$ vectors associated with this eigenvalue are referred to as principal vectors. The complete set of principal vectors are all linearly independent.

4.7 Singular Value Decomposition

Given a matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ of full rank n, the SVD is given by

$$\underbrace{\mathbf{A}}_{m\times n} = \underbrace{\hat{\mathbf{U}}}_{m\times r}\underbrace{\hat{\mathbf{\Sigma}}}_{r\times r}\underbrace{\hat{\mathbf{V}}^*}_{r\times n}.$$

The columns of the left $\hat{\mathbf{U}}$ and right $\hat{\mathbf{V}}$ singular vectors are orthonormal. $\hat{\mathbf{\Sigma}}$ is a diagonal matrix with entries $\{\sigma_1, \sigma_2, ..., \sigma_r\}$ (the singular values of $\mathbf{A}$, typically arranged in descending order.) We can write this as,

$$\mathbf{A} = \Sigma_{j=1}^{r} \sigma_j \hat{\mathbf{u}}_j \hat{\mathbf{v}}_j^*,$$

where r is the rank of $\mathbf{A}$.
We also have

$$\mathbf{A}\hat{\mathbf{V}} = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}.$$

This can be written as

$$\mathbf{A}\hat{\mathbf{v}}_j = \sigma_j \hat{\mathbf{u}}_j.$$

Thus a geometric interpretation is that the singular values are the lengths of the semi-axes of the hyper-ellipsoid which is a consequence of the transformation of unit hypersphere operated on by $\mathbf{A}$. This is called **reduced SVD** of $\mathbf{A}$.

The **full SVD** of $\mathbf{A}$ is given by

$$\underbrace{\mathbf{A}}_{m\times n} = \underbrace{\mathbf{U}}_{m\times m}\underbrace{\mathbf{\Sigma}}_{m\times n}\underbrace{\mathbf{V}^*}_{n\times n},$$

where $\hat{\mathbf{U}}$ is augmented with $m-r$ orthonormal columns and $\hat{\mathbf{V}}$ is augmented with $n-r$ orthonormal columns to complete their respective spaces. A corresponding number of zeros are added to $\hat{\mathbf{\Sigma}}$. Note that $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices.
For a matrix of rank r,

$\mathbf{u}_1, \ldots, \mathbf{u}_r$ are orthonormal bases for $Range(\mathbf{A})$ and $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ are orthonormal bases for $Null(\mathbf{A})$.

$\mathbf{u}_{r+1}, \ldots, \mathbf{u}_m$ are orthonormal bases for $Null(\mathbf{A}^*)$ and $\mathbf{v}_1, \ldots, \mathbf{v}_r$ are orthonormal bases for $Range(\mathbf{A}^*)$.

4.7.1 SVD facts

1. Every matrix has a SVD and the singular values are uniquely determined (but they are not necessarily distinct). If the matrix is square and singular values are distinct, the left and right singular vectors are uniquely determined up to complex signs (i.e., complex scalar factors of absolute value 1).
2. If $\mathbf{A}$ is rank-deficient (of rank r), then $\mathbf{\Sigma}$ will contain r positive diagonal entries and $n-r$ zero elements.
3.

$$\begin{aligned} \mathbf{A}^*\mathbf{A} &= [\mathbf{U\Sigma V}^*]^* [\mathbf{U\Sigma V}^*] && (4.17) \\ &= \mathbf{V\Sigma}^2\mathbf{V}^*, && (4.18) \end{aligned}$$

thus σ_i are the square-roots of the eigenvalues of $\mathbf{A}^*\mathbf{A}$.
4. Similarly,

$$\mathbf{AA}^* = \mathbf{U\Sigma}^2\mathbf{U}^*, \qquad (4.19)$$

thus σ_i are the square-roots of the eigenvalues of $\mathbf{AA}^*$.
5. The rank r of $\mathbf{A}$ is equal to the number of non-zero singular values of $\mathbf{A}$.
6. The range and null space of $\mathbf{A}$ are spanned by the first r columns of $\mathbf{U}$ and the last $n-r$ columns of $\mathbf{V}$.
7. The range and null space of $\mathbf{A}^*$ are spanned by the first r columns of $\mathbf{V}$ and the last $m-r$ columns of $\mathbf{U}$.
8. $\mathbf{x} = \mathbf{Ay} = \mathbf{U\Sigma V}^*\mathbf{y}$ can be interpreted as

$$\mathbf{x} = \sum_{i=1}^{r} (\mathbf{v}_i \cdot \mathbf{y}\sigma_i)\mathbf{u}_i.$$

In other words, $\mathbf{x}$ is represented in $\mathbf{U}$ space and the projection coefficients are obtained by projecting $\mathbf{y}$ onto $\mathbf{V}$ and scaling by the singular value.
9. $\mathbf{y} = \mathbf{V\Sigma}^{-1}\mathbf{U}^*\mathbf{x}$ can be interpreted as

$$\mathbf{y} = \sum_{i=1}^{r} (\mathbf{u}_i \cdot \mathbf{x}/\sigma_i)\mathbf{v}_i.$$

In other words, $\mathbf{y}$ is represented in $\mathbf{V}$ space and the projection coefficients are obtained by projecting $\mathbf{x}$ onto $\mathbf{u}$ and scaling by the inverse singular value.
10. Given $\mathbf{A} = \Sigma_{j=1}^{r}\sigma_j\mathbf{u}_j\mathbf{v}_j^*$, then for any $0 \leq q \leq r$, the matrix $\mathbf{A}_q = \Sigma_{j=1}^{q}\sigma_j\mathbf{u}_j\mathbf{v}_j^*$ satisfies the following properties:

$$\|\mathbf{A} - \mathbf{A}_q\|_2 = \sigma_{q+1} \quad ; \quad \|\mathbf{A} - \mathbf{A}_q\|_F = \sqrt{\sum_{i=q+1}^{r} \sigma_i^2}.$$

$$\|\mathbf{A}^{+} - \mathbf{A}_q^{+}\|_F = \sqrt{\sum_{i=q+1}^{r} \frac{1}{\sigma_i^2}}.$$

Note: $\mathbf{A}_q$ is also the best rank q approximation to $\mathbf{A}$. In other words $\|\mathbf{A} - \mathbf{A}_q\| \leq \|\mathbf{A} - \mathbf{B}\|$, for any $\mathbf{B}$ of rank q.
11. If $\mathbf{A} \in \mathbb{R}^{n\times n}$, then the "closest" orthogonal matrix $\mathbf{A}$ is given by $\mathbf{Q} = \mathbf{UIV}^T$.

$$\|\mathbf{A} - \mathbf{Q}\|_F = \|\mathbf{U}(\mathbf{\Sigma} - \mathbf{I})\mathbf{V}^T\|_F = \sqrt{\sum_{i=q+1}^{r} (\sigma_i - 1)^2}.$$

12.

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^*\mathbf{A})} = \sigma_{\max}(\mathbf{A})$$

.
Proof:

$$\sup_{\|x\|_2=1} \|Ax\|_2 = \sup_{\|x\|_2=1} \|U\Sigma V^T x\|_2 = \sup_{\|x\|_2=1} \|\Sigma V^T x\|_2$$

Then let $y = V^T x$. $\|y\|_2 = \|V^T x\|_2 = \|x\|_2 = 1$ since V is unitary.

$$\sup_{\|x\|_2=1} \|\Sigma V^T x\|_2 = \sup_{\|y\|_2=1} \|\Sigma y\|_2$$

Since $\Sigma = \mathrm{diag}(\sigma_1, \cdots, \sigma_n)$, where σ_1 is the largest singular value. The max for the above, σ_1, is attained when $y = (1, \cdots, 0)^T$.
13.

$$\|\mathbf{A}^{-1}\|_2 = \frac{1}{\sqrt{\lambda_{\min}(\mathbf{A}^*\mathbf{A})}} = \frac{1}{\sigma_{\min}(\mathbf{A})}$$

.
14. $\sum_{i=1}^{r} \sigma_i^2 = \|\mathbf{A}\|_F$.

4.7.2 Example

Listing 4.2 gives a demo of a simple use of SVD in image compression. The results of the compression are shown in Figures 4.2 and 4.3. Note that if the image was greyscale, there will only be one matrix (instead of three) to process.

Listing 4.1: A sample code to compress an image using SVD.

```
clear all

% Input image
image=imread('ball.jpg');

% Check size
```

```
size(image)

% Decompose RGB into three distinct matrices
A=image(:,:,1);
AD=double(A);
B=image(:,:,2);
BD=double(B);
C=image(:,:,3);
CD=double(C);

% Compute SVD
[UA,SA,VA]=svd(AD);
[UB,SB,VB]=svd(BD);
[UC,SC,VC]=svd(CD);


% Number of singular values to keep
N=100

    % Truncate the SVD
    SA(N+1:end,:)=0;
    SA(:,N+1:end)=0;

    SB(N+1:end,:)=0;
    SB(:,N+1:end)=0;

    SC(N+1:end,:)=0;
    SC(:,N+1:end)=0;

    % Low resolution image
    D(:,:,1)=UA*SA*VA';
    D(:,:,2)=UB*SB*VB';
    D(:,:,3)=UC*SC*VC';

    % write the file
    imwrite(uint8(D),'ballreduced.jpg','jpg');

    % display and compute error

    subplot(1,2,1)
    imshow(image);
    buffer = sprintf('Image input with %d singular values', length(SA))
    title(buffer);

    subplot(1,2,2)
```

```
imshow(uint8(D));
buffer = sprintf('Image output using %d singular values', N)
title(buffer);

error=sum(sum(sum(((double(image)-D).^2))))
```

Note on SVD
The SVD can be used to diagonalize any matrix. For instance, consider $\mathbf{Ax} = \mathbf{b}$.

$$\begin{aligned} \mathbf{U}^*\mathbf{Ax} &= \mathbf{U}^*\mathbf{b} \qquad (4.20)\\ \mathbf{\Sigma V}^*\mathbf{x} &= \mathbf{U}^*\mathbf{b} \qquad (4.21)\\ \mathbf{\Sigma}\hat{\mathbf{x}} &= \hat{\mathbf{b}}, \qquad (4.22) \end{aligned}$$

where $\hat{\mathbf{x}} = \mathbf{V}^*\mathbf{x}$ and $\hat{\mathbf{b}} = \mathbf{U}^*\mathbf{b}$.
Note that eigendecompositions are used when the domain and range spaces are the same (i.e. when $\mathbf{A}$ is a square matrix with certain properties). SVD is used when these spaces are different (when $\mathbf{A}$ is rectangular).

4.8 Other matrix factorizations

4.8.1 QR Factorization

If $\mathbf{A} \in \mathbb{C}^{m\times n}$ and $\mathbf{A}$ has full column rank, then the QR decomposition is given by $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{C}^{m\times n} = \{\mathbf{q}_1, \mathbf{q}_2, \dots \mathbf{q}_n\}$ are orthonormal vectors and $\mathbf{R} \in \mathbb{C}^{n\times n}$ is an upper-triangular matrix with non-zero diagonal elements.

There are many ways of computing the QR decomposition, but the Gram-Schmidt procedure is most intuitive. Since we know that
$\mathbf{a}_i = (\mathbf{q}_1^T\mathbf{a}_i)\mathbf{q}_1 + (\mathbf{q}_2^T\mathbf{a}_i)\mathbf{q}_2 .. + (\mathbf{q}_{i-1}^T\mathbf{a}_i)\mathbf{q}_{i-1} + \|\tilde{\mathbf{q}}_i\|\mathbf{q}_i$,
we can directly determine the elements of $\mathbf{R}$ by inspection.

In the general case, the full QR decomposition is given by $\mathbf{A} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{C}^{m\times m}$ is a unitary matrix.

When $m > n$

$\mathbf{R} \in \mathbb{C}^{m\times n}$ is such that

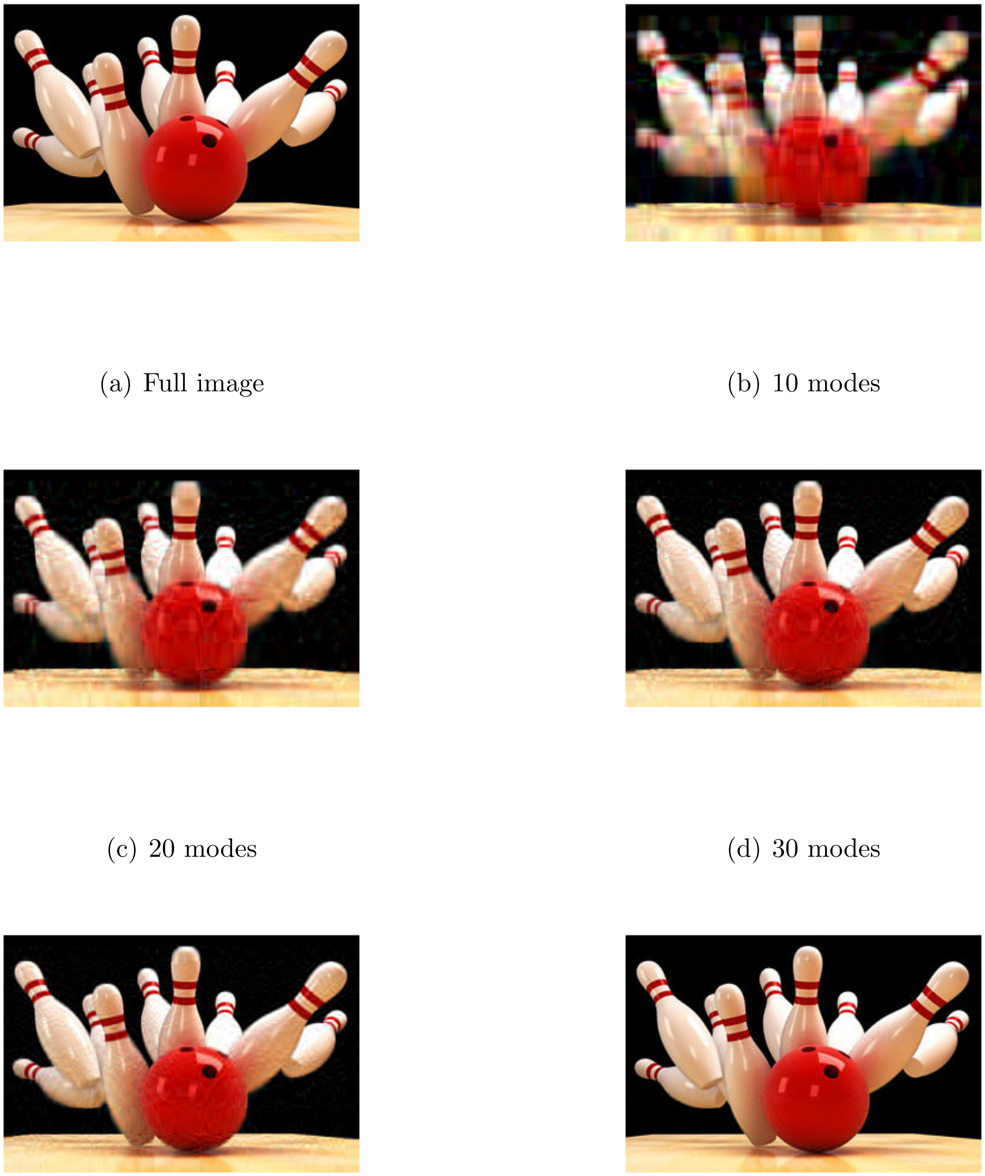
$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix},$$

where $\mathbf{R}_1 \in \mathbb{C}^{n\times n}$ is an upper triangular matrix and $\mathbf{R}_2 \in \mathbb{C}^{m-n\times n}$ is a matrix of zeroes.

$$\mathbf{Q} = [\mathbf{Q}_1 \;\; \mathbf{Q}_2],$$

where $\mathbf{Q}_1 \in \mathbb{C}^{m\times n}$ and $\mathbf{Q}_2 \in \mathbb{C}^{m\times m-n}$.

Note:If $rank(A) = n$, then



(a) Full image

(b) 10 modes

(c) 20 modes

(d) 30 modes

(e) 40 modes

(f) 100 modes

Figure 4.2: Image compression using SVD

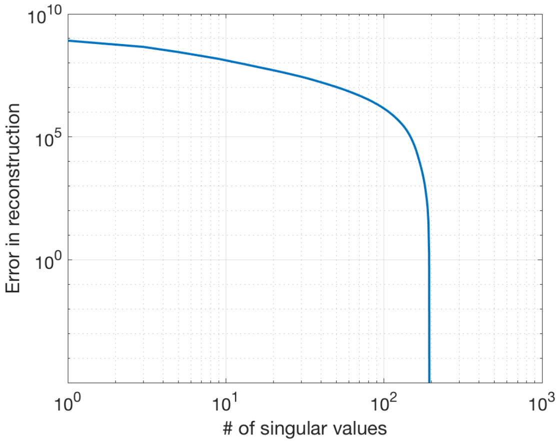


Figure 4.3: L2 error in reconstruction

Columns of $\mathbf{Q}_1$ are an orthonormal basis for $range(\mathbf{A})$,
Columns of $\mathbf{Q}_2$ are an orthonormal basis for $null(\mathbf{A}^T)$,
$\mathbf{R}_1$ is non-singular.

When $n > m$
$\mathbf{R} \in \mathbb{C}^{m \times n}$ is such that

$$\mathbf{R} = [\mathbf{R}_1 \;\; \mathbf{R}_2]\,,$$

where $\mathbf{R}_1 \in \mathbb{C}^{m \times m}$ is an upper triangular matrix and $\mathbf{R}_2 \in \mathbb{C}^{m \times n-m}$.

4.8.2 Cholesky Factorization

If $\mathbf{A} \in \mathbb{C}^{n \times n}$ is a Hermitian positive definite matrix, then $\mathbf{A} = \mathbf{R}^*\mathbf{R}$, where $\mathbf{R}$ is upper triangular with positive diagonal elements. This factorization is unique.

4.8.3 Connection between Cholesky and QR

If $\mathbf{A} \in \mathbb{C}^{m \times n}$ and $m \geq n$, then $\mathbf{A} = \mathbf{QR}$, then $\mathbf{A}^*\mathbf{A} = \mathbf{R}^*\mathbf{R}$.

4.9 Pseudoinverse

The pseudoinverse $\mathbf{A}^+$ of a $m \times n$ matrix $\mathbf{A}$ is a $n \times m$ matrix satisfying all of the following four criteria:

$$\begin{aligned} \mathbf{A}\mathbf{A}^+\mathbf{A} &= \mathbf{A} && (4.23)\\ \mathbf{A}^+\mathbf{A}\mathbf{A}^+ &= \mathbf{A}^+ && (4.24)\\ (\mathbf{A}\mathbf{A}^+)^T &= \mathbf{A}\mathbf{A}^+ && (4.25)\\ (\mathbf{A}^+\mathbf{A})^T &= \mathbf{A}^+\mathbf{A} && (4.26) \end{aligned}$$

$\mathbf{A}^+$ exists for any matrix, $\mathbf{A}$, but under the following conditions explicit expressions can be developed:

$\mathbf{A}$ has linearly independent columns (rank($\mathbf{A}$)=n): $\mathbf{A}^T\mathbf{A}$ is invertible and thus $\mathbf{A}^+ = (\mathbf{A}^*\mathbf{A})^{-1}\mathbf{A}^*$. This is a *left inverse*, since, in this case, $\mathbf{A}^+\mathbf{A} = \mathbf{I}$. Note: In this case, the pseudoinverse can also be computed using the QR decomposition : $\mathbf{A}^+ = \mathbf{R}^{-1}\mathbf{Q}^T$.

$\mathbf{A}$ has linearly independent rows (rank($\mathbf{A}$)=m): $\mathbf{A}\mathbf{A}^T$ is invertible, and thus $\mathbf{A}^+ = \mathbf{A}^*(\mathbf{A}\mathbf{A}^*)^{-1}$. This is a *right inverse*, as $\mathbf{A}\mathbf{A}^T = \mathbf{I}$. Note: In this case, the pseudoinverse can also be computed using the QR decomposition : $\mathbf{A}^+ = \mathbf{Q}\mathbf{R}^{-T}$.

If rank($\mathbf{A}$)=r and $\mathbf{A} = \mathbf{U\Sigma V}^*$, then

$$\mathbf{A}^+ = \mathbf{V}diag(\sigma_1^{-1}, \sigma_2^{-1}, ..., \sigma_r^{-1}, 0, 0...0)\mathbf{U}^*,$$

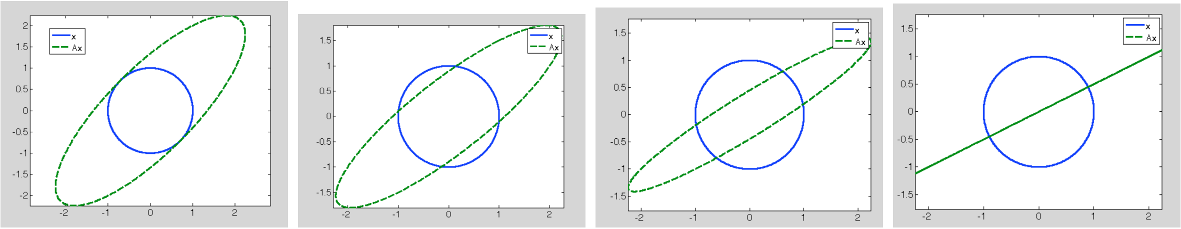
where $\mathbf{A} = \mathbf{U\Sigma V}^*$ is the SVD. This is a very important application of the SVD.

4.10 Condition number

The condition number (with respect to a $p - norm$) of a matrix is defined as $\kappa_p(\mathbf{A}) = \|\mathbf{A}\|_p\|\mathbf{A}^{-1}\|_p$. The condition number is a measure of the ratio of the relative output to a relative input perturbation. This can be visually understood by considering four matrices as shown below:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}; \quad \mathbf{A}_2 = \begin{bmatrix} 1 & 2 \\ 1.5 & 1 \end{bmatrix}; \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 2 \\ 1.0 & 1 \end{bmatrix}; \quad \mathbf{A}_4 = \begin{bmatrix} 1 & 2 \\ 0.5 & 1 \end{bmatrix} \tag{4.27}$$

The 2-norm condition number for these matrices is $3, 3.8664, 6.8541, \infty$, respectively. This can be understood by looking at the transformation of the unit circle by the action of these matrices.



(a) $\mathbf{A}_1$ (b) $\mathbf{A}_2$ (c) $\mathbf{A}_3$ (d) $\mathbf{A}_4$

Figure 4.4: Transformation of unit circle.

Consider a matrix equation $\mathbf{Ax} = \mathbf{b}$. Consider an input perturbation $\Delta\mathbf{x}$ that leads to an output perturbation $\Delta\mathbf{b}$:

$$\begin{aligned}
\mathbf{A}(\mathbf{x}+\Delta\mathbf{x}) &= \mathbf{b}+\Delta\mathbf{b} && (4.28)\\
\mathbf{A}\Delta\mathbf{x} &= \Delta\mathbf{b} && (4.29)\\
\Delta\mathbf{x} &= \mathbf{A}^{-1}\Delta\mathbf{b} && (4.30)\\
\|\Delta\mathbf{x}\| &= \|\mathbf{A}^{-1}\Delta\mathbf{b}\| && (4.31)\\
\|\Delta\mathbf{x}\| &\leq \|\mathbf{A}^{-1}\|\|\Delta\mathbf{b}\| && (4.32)\\
\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{A}\mathbf{x}\|} &\leq \frac{\|\mathbf{A}^{-1}\|\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} && (4.33)\\
\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{A}\|\|\mathbf{x}\|} &\leq \frac{\|\mathbf{A}^{-1}\|\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} && (4.34)\\
\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} &\leq \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} && (4.35)\\
\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} &\leq \kappa(\mathbf{A})\frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} && (4.36)\\
\frac{\|\Delta\mathbf{x}\|_2}{\|\mathbf{x}\|_2} &\leq \frac{\sigma_{max}(\mathbf{A})}{\sigma_{min}(\mathbf{A})}\frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} && (4.37)
\end{aligned}$$

4.11 Least squares regression

Consider an overdetermined linear system ($m > n$) of equations

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}. \qquad (4.38)$$

We can take $\mathbf{y}$ to be the *data* or *observations*, and $\mathbf{x}$ to be the unknown *model parameters.*

Such a system usually has no solution, so the goal is instead to find the parameters $\mathbf{x}$ which fit the equations "best," in the sense of solving the minimization problem:

$$\hat{\mathbf{x}} = arg \min_{\mathbf{x}} S(\mathbf{x}),$$

where the objective function is given by:

$$S(\mathbf{x}) = \left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2.$$

This minimization problem has a unique solution, provided that the n columns of the matrix $\mathbf{A}$ are linearly independent. The solution is given by the normal equations

$$S(\mathbf{x}) = \left\|\mathbf{y} - \mathbf{A}\mathbf{x}\right\|_2^2 = (\mathbf{y} - \mathbf{A}\mathbf{x})^{\mathrm{T}}(\mathbf{y} - \mathbf{A}\mathbf{x}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{y} - \mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x}.$$

Note that: $(\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{y})^{\mathrm{T}} = \mathbf{y}^{\mathrm{T}}\mathbf{A}\mathbf{x}$ and the quantity to minimize becomes:

$$S(\mathbf{x}) = \mathbf{y}^{\mathrm{T}}\mathbf{y} - 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{y} + \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x}.$$

Differentiating this with respect to $\mathbf{x}$ and equating to zero to satisfy the first-order conditions gives:

$$-\mathbf{A}^{\mathrm{T}}\mathbf{y} + (\mathbf{A}^{\mathrm{T}}\mathbf{A})\mathbf{x} = 0,$$

which is equivalent to the above-given normal equations. A sufficient condition for satisfaction of the second-order conditions for a minimum is that $\mathbf{A}$ have full column rank, in which case $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is positive definite. The matrix $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ is known as the Gramian matrix of $\mathbf{A}$, which possesses several nice properties such as being a positive semi-definite matrix, and the matrix $\mathbf{A}^{\mathrm{T}}\mathbf{y}$ is known as the moment matrix.

Thus

$$\hat{\mathbf{x}} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y} = \mathbf{A}^{+}\mathbf{y}$$

$\mathbf{A}\hat{\mathbf{x}}$ is the orthogonal projection of $\mathbf{y}$ onto range($\mathbf{A}$). In other words

$$\mathcal{P}_{range(\mathbf{A})}(\mathbf{y}) = \mathbf{A}\hat{\mathbf{x}} = \mathbf{A}(\mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{T}\mathbf{y}.$$

The residual $\mathbf{r} = \mathbf{A}\hat{\mathbf{x}} - \mathbf{y}$ is orthogonal to the range of $\mathbf{A}$. This means $\mathbf{r}$ is in the null space of $\mathbf{A}^{T}$. In other words, $(\mathbf{r}, \mathbf{A}\mathbf{z}) = 0 \;\; \forall \mathbf{z} \in \mathbb{R}^{n}$ and $\mathbf{A}^{T}(\mathbf{A}\hat{\mathbf{x}} - \mathbf{y}) = 0$.

In summary, if $\mathbf{y} \in range(\mathbf{A})$, then we can satisfy $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}$ precisely. If not, we can satisfy $\mathcal{P}_{range(\mathbf{A})}(\mathbf{y}) = \mathbf{A}\hat{\mathbf{x}}$.

4.12 Least Norm regression

In under-determined systems, we have $\mathbf{Ax} = \mathbf{y}$, but with $m < n$. Thus, any parameter vector $\mathbf{x}$ that satisfies $\mathbf{Ax} = \mathbf{y}$ is consistent with the above system, and thus we have multiple solutions. To pin down a useful solution, a "Least-norm" solution is often desired. Least norm problems have the form

$$\tilde{\mathbf{x}} = arg \min_{\mathbf{x}} \|\mathbf{x}\|_2 \ \ ; \ \ subject \ \ to \ \ \mathbf{Ax} = \mathbf{y}.$$

For under-determined systems with full row rank, the least norm solution is given by

$$\tilde{\mathbf{x}} = \mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{y} = \mathbf{A}^{+}\mathbf{y}.$$

Let's prove this via standard constrained optimization. The problem is:

$$min \ \ \mathbf{x}^T\mathbf{x} \ \ s.t. \ \ \mathbf{Ax} = \mathbf{y}$$

.

Define a Lagrangian

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{x}^T\mathbf{x} + \boldsymbol{\lambda}^T(\mathbf{Ax} - \mathbf{y}).$$

The first order optimality conditions are:

$$\nabla_{\mathbf{x}} L = 0 \ \ ; \ \ \nabla_{\boldsymbol{\lambda}} L = 0.$$

These translate to

$$2\mathbf{x} + \mathbf{A}^T\boldsymbol{\lambda} = \mathbf{0} \ \ ; \ \ \mathbf{Ax} - \mathbf{y} = \mathbf{0}.$$

Thus $\mathbf{x} = -\frac{1}{2}\mathbf{A}^T\boldsymbol{\lambda}$. Plugging this into $\mathbf{Ax} - \mathbf{y} = \mathbf{0}$ we get $\boldsymbol{\lambda} = -2\mathbf{y}(\mathbf{AA}^T)^{-1}$, and thus $\tilde{\mathbf{x}} = \mathbf{A}^T(\mathbf{AA}^T)^{-1}$.

Now let's check this. Let's take any $\mathbf{x}$ that satisfies $\mathbf{Ax} = \mathbf{y}$. We know that $\mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}) = \mathbf{0}$, and so

$$\begin{aligned} (\mathbf{x} - \tilde{\mathbf{x}})^{\mathbf{T}}\tilde{\mathbf{x}} &= (\mathbf{x} - \tilde{\mathbf{x}})^T\mathbf{A}^T(\mathbf{AA}^T)^{-1}\mathbf{y} && (4.39) \\ &= (\mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}))^T(\mathbf{AA}^T)^{-1}\mathbf{y} && (4.40) \\ &= 0. && (4.41) \end{aligned}$$

Since $(\mathbf{x} - \tilde{\mathbf{x}}) \perp \tilde{\mathbf{x}}$,

$$\|\mathbf{x}\|^{\mathbf{2}} = \|\mathbf{x} - \tilde{\mathbf{x}} + \tilde{\mathbf{x}}\|^{\mathbf{2}} = \|\mathbf{x} - \tilde{\mathbf{x}}\|^{\mathbf{2}} + \|\tilde{\mathbf{x}}\|^{\mathbf{2}} \geq \|\tilde{\mathbf{x}}\|^{\mathbf{2}}.$$

Thus $\tilde{\mathbf{x}}$ is the minimizer.

The least-norm chooses the smallest plausible $\mathbf{x}$ (which may represent prior knowledge).

4.13 Rank-deficient systems

Thus far, we assumed that the Least squares problem involved a matrix of full column rank and the least norm problem involved a matrix of full Row rank. Now, let's relax the least squares problem and assume that $rank(\mathbf{A}) = r < n$. There are many ways of solving this problem and here we pursue the SVD approach. One can perform an SVD

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = [\mathbf{U}_1 \;\; \mathbf{U}_2] \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_1 \;\; \mathbf{V_2}]^T = \mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T,$$

where $\mathbf{U} \in \mathbb{R}^{m\times m}$, $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$. $\mathbf{U}_1, \mathbf{V}_1$ have r columns and $\mathbf{\Sigma}_1$ contains all the non-zero singular values.

$$\begin{aligned} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 &= \|\mathbf{U}^T(\mathbf{A}\mathbf{V}\mathbf{V^T}\mathbf{x} - \mathbf{y})\|_2^2 && (4.42) \\ &= \|\mathbf{\Sigma}\mathbf{V}^T\mathbf{x} - \mathbf{U}^T\mathbf{y}\|_2^2 && (4.43) \\ &= \sum_{i=1}^{r}(z_i\sigma_i - \mathbf{u}_i^T\mathbf{y})^2 + \sum_{i=r+1}^{m}(\mathbf{u}_i^T\mathbf{y})^2, && (4.44) \end{aligned}$$

where $\mathbf{z} = \mathbf{V}^T\mathbf{x}$.

Thus, to minimize the norm, we can set $z_i = \frac{\mathbf{u}_i^T\mathbf{y}}{\sigma_i}$ for $1 \leq i \leq r$. The rest of the components of $\mathbf{z}_i$ can be completely arbitrary.

Therefore we have $\hat{\mathbf{x}} = \sum_{i=1}^{r} \frac{\mathbf{u}_i \cdot \mathbf{y}}{\sigma_i}\mathbf{v}_i$. With this choice, $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 = \sum_{i=r+1}^{m}(\mathbf{u}_i^T\mathbf{y})^2$.

Note: The solution can also be written as

$$\hat{\mathbf{x}} = \mathbf{V}_1\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^T\mathbf{y}.$$

4.14 Improving robustness

There are many reasons to pursue more advanced minimization techniques including robustness.

Let's consider the general problem

$$min_{\mathbf{x}} \;\; \mathbf{A}\mathbf{x} = \mathbf{y},$$

where as before, $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{x} \in \mathbb{R}^n, \mathbf{y} \in \mathbb{R}^m$.

Defining $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{y}$, we have

$$\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p = [|r_1|^p + |r_2|^p + ...|r_m|^p]^{1/p}$$

4.14.1 p-norm minimization and penalty function approximation

Consider the following minimization problem

$$min_{\mathbf{x}} \;\; \phi(r_1) + ... + \phi(r_m),$$

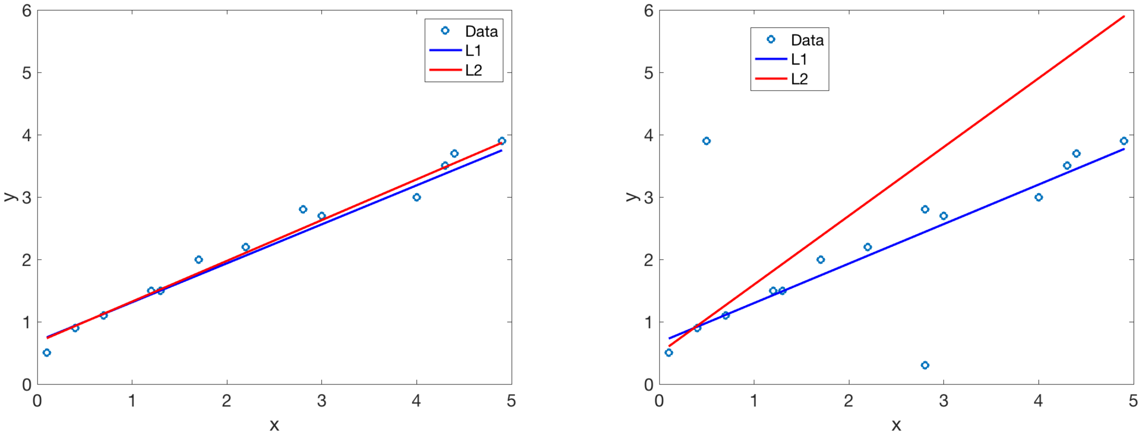
where $\phi(r_i)$ is a penalty function.

Boyd notes the following: “Let’s compare L1-norm and L2-norm approximation, associated with the penalty functions $\phi_1(u) = |u|$ and $\phi_2(u) = u^2$, respectively. For $|u| = 1$, the two penalty functions assign the same penalty. For small u we have $\phi_1(u) >> \phi_2(u)$, so L1-norm approximation puts relatively larger emphasis on small residuals compared to L2-norm approximation. For large u, we have $\phi_2(u) >> \phi_1(u)$, so L1-norm approximation puts less weight on large residuals, compared to L2-norm approximation. This difference in relative weightings for small and large residuals is reflected in the solutions of the associated approximation problems. The amplitude distribution of the optimal residual for the L1-norm approximation problem will tend to have more zero and very small residuals, compared to the L2-norm approximation solution. In contrast, the L2-norm solution will tend to have relatively fewer large residuals (since large residuals incur a much larger penalty in L2-norm approximation than in L1-norm approximation).”

This is quite clear in the code below and Figure 4.5, which shows the performance of L1 and L2 norm-based regression on data generated by perturbing $y = a_0 + a_1 x$.

Listing 4.2: A sample code to compress an image using SVD.

```
% Simple code to compare L2 vs L1 regression with a linear model

% clean data
x=[0.1 0.4 0.7 1.2 1.3 1.7 2.2 2.8 3.0 4.0 4.3 4.4 4.9];
y=[0.5 0.9 1.1 1.5 1.5 2.0 2.2 2.8 2.7 3.0 3.5 3.7 3.9];



coeff_L2=fminsearch('line_L2_fit',[1,1],[],x,y)
coeff_L1=fminsearch('line_L1_fit',[1,1],[],x,y)

figure(1)
plot(x,y,'o',x,coeff_L1(1)+coeff_L1(2)*x,'b-', ...
 x,coeff_L2(1)+coeff_L2(2)*x,'r-','LineWidth',2)
legend('Data','L1','L2')
set(gca,'FontSize',16)
xlabel('x')
ylabel('y')
ylim([0 6])



% corrupted data
x=[x 0.5 2.8];
y=[y 3.9 0.3];

coeff_L2=fminsearch('line_L2_fit',[1,1],[],x,y)
coeff_L1=fminsearch('line_L1_fit',[1,1],[],x,y)

figure(2)
```



(a) 'Good' data

(b) 'Corrupted' data

Figure 4.5: L1 and L2 regression example

```
plot(x,y,'o',x,coeff_L1(1)+coeff_L1(2)*x,'b-', ...
 x,coeff_L2(1)+coeff_L2(2)*x,'r-','LineWidth',2)
legend('Data','L1','L2')
set(gca,'FontSize',16)
xlabel('x')
ylabel('y')
ylim([0 6])

end

%========
%Functions
%=======


 function E=line_L1_fit(x0,x,y)
E=sum( abs( x0(1)*x+x0(2) - y ) )
end


 function E=line_L2_fit(x0,x,y)
E=sum( ( x0(1)*x+x0(2) - y ).^2 )
end
```

We can design other types of penalty functions such as
a) the "deadzone-linear" penalty function

$$\phi(u) = 0 \quad |u| \leq a \quad ; \quad \phi(u) = |u| - a \quad |u| > a$$

which puts no penalty on small residuals, or

b) the ‘Huber” penalty function

$$\phi(u) = u^2 \ \ |u| \leq a \ \ ; \ \ \phi(u) = a(2|u| - a) \ \ |u| > a$$

This penalty function agrees with the least-squares penalty function for residuals smaller than a, and then reverts to L1-like linear growth for larger residuals, thus robust to outliers.

4.15 Improving robustness

The above example was a clear case where the model was too sensitive to outliers. In general learning problems, we encounter other difficulties : The model may be can be “too complex” for the available data and may “overfit” to the data. This can happen especially when there is a lot of noise in the data. In this scenario, we may have to choose from different models (e.g. quadratic, cubic, quartic, quintic etc.)

The common sense way to do this, is to separate the data into a training and validation set. Start with a set of candidate models. Train each model on the training set and evaluate the error of each model on the validation set. The model with the least validation error is potentially a good model. This is an example of the so-called cross-validation approach.

Instead of explicitly evaluating several model classes, regularization offers a more elegant route to improve robustness.

Regularized problems combine both of the above aspects, seeking to determine

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p^p + \lambda \|\mathbf{x}\|_q^q,$$

where $\lambda > 0$ is a regularization parameter. When $\mathbf{A}$ is poorly conditioned, regularization gives a compromise between solving the equations and keeping $\mathbf{x}$ to a reasonable size. When $p = 2$, there is a very clear and rigorous connection to Bayesian inference, which we will explore later. $p = 2$ corresponds to a Gaussian likelihood. In this setting, $q = 2$ is formally equivalent to the use of a Gaussian prior, and $q = 1$ is formally equivalent to the use of a Laplace prior.

The geometric interpretation is also apparent. Consider the following constrained optimization problem

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \ \ s.t. \|\mathbf{x}\|_2^2 \leq \eta.$$

The Lagrangian for this problem can be written as

$$\mathcal{L} = \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda(\mathbf{x}^T\mathbf{x} - \eta),$$

It can be shown that for every choice of λ, there is a unique η such that the unconstrained minimization of the Lagrangian and the constrained minimization preceding it has the same solution. Thus, we are minimizing the ‘loss function’ $\|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$ under the constraint that $\mathbf{x}$ resides in a ball (in the q norm) of radius δ. Thus, all possible models residing outside of the ball of radius η is discarded.

4.15.1 L2 minimization with L2 regularization

When $p = q = 2$, the problem is

$$min_{\mathbf{x}} \frac{1}{2}\|\mathbf{Ax} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2.$$

$$\frac{1}{2}\|\mathbf{Ax} - \mathbf{y}\|_2^2 + \frac{\lambda}{2}\|\mathbf{x}\|_2^2 \;=\; \| \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix} \mathbf{x} - \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} \|_2^2, \tag{4.45}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$. Regardless of the rank of $\mathbf{A}$,

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}$$

is always of full rank n.

Therefore, the normal equation corresponding to this system is

$$\begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix} \mathbf{x} \;=\; \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{I} \end{bmatrix}^T \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix}^T \tag{4.46}$$

$$(\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})\mathbf{x} \;=\; \mathbf{A}^T\mathbf{y}. \tag{4.47}$$

Therefore,

$$\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{y}.$$

Thus, regularization improves the conditioning of the problem. In fact, it is possible to show that $\hat{\mathbf{x}} = \sum_{i=1}^{r} \frac{\sigma_i \mathbf{u}_i \cdot \mathbf{y}}{\sigma_i^2 + \lambda} \mathbf{v}_i$.

The above procedure is also called ridge regression. This is also a special case of Tikhonov regularization.

4.15.2 Other forms of regularization

When $p = 2$ and $q = 1$, the problem is

$$min_{\mathbf{x}}\|\mathbf{Ax} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}\|_1.$$

This is called LASSO regression. LASSO is more robust to outliers than ridge and also yields more parsimonious models (i.e. less non-zero x_i than ridge). However, unlike ridge, analytical solutions to LASSO are generally intractable and computational solutions are typically much more expensive. Perhaps most importantly, ridge is a strictly convex problem (thus there are unique solutions) whereas LASSO is not (unless $\mathbf{A}$ is of full column rank).

Another type of regularization involves smoothing in term of the form $\|\Delta\mathbf{x}\|$. This typically amounts to second-order differentiation, so $\|\Delta\mathbf{x}\|$ represents a measure of the smoothness of $\mathbf{x}$. For instance, $(\Delta\mathbf{x})_j = c(x_{j+1} - 2x_j + x_{j-1})$.

So advanced regularization could involve the minimization of

$$\|\mathbf{Ax} - \mathbf{y}\|_p + \lambda\|\mathbf{x}\|_q + \delta\|\Delta\mathbf{x}\|_r.$$

Insight on Lagrangian

Consider the following problem

$$\min_{\mathbf{x}} \ \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 \ \ s.t. \ \ \|\mathbf{x}\|_2^2 \leq \eta, \tag{4.48}$$

where $\eta > 0$. We are thus interested in minimizing $f(\mathbf{x}) \triangleq \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2$ in a feasible region defined by $g(\mathbf{x}) \triangleq \|\mathbf{x}\|_2^2 \leq \eta$.

Now, let us break free of the specific problem above and generalize to a strictly convex optimization problem

$$\min_{\mathbf{x}} \ f(\mathbf{x}) \ \ s.t. \ \ g(\mathbf{x}) \leq \eta. \tag{4.49}$$

If the global minima of $f(\mathbf{x})$ lies inside the feasible region, then the problem is solved by $\min_{\mathbf{x}} f(\mathbf{x})$, which in the case of eq. 4.48 is the (unregularized) least squares solution. If the global minima of $f(\mathbf{x})$ lies outside the feasible region, then we can seek a solution on the boundary of the feasible region. This is because of the convexity of problem which guarantees that, inside any convex region, the extrema will be at the boundaries. In seeking the minima on the boundary $g(\mathbf{x}) = \eta$, it is clear that the gradients $\nabla_{\mathbf{x}} f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ should have opposing signs. This is because of all of the following reasons:

1. $f(\mathbf{x})$ increases inside the feasible region.
2. $g(\mathbf{x})$ increases outside the feasible region.
3. If we want to find an optimal solution, $f(\mathbf{x})$ should not increase along the constraint boundary $g(\mathbf{x}) = \eta$, and so we seek the maximum change to be aligned with the normal to the constraint surface.

Therefore, it is sensible to seek

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = -\alpha \nabla_{\mathbf{x}} g(\mathbf{x}) \ \ \text{and thus} \ \ \nabla_{\mathbf{x}}(f(\mathbf{x}) + \alpha g(\mathbf{x})) = 0.$$

where $0 \leq \alpha \in \mathbb{R}$ is a parameter.

Hence, one can define a Lagrangian

$$\mathcal{L}(\mathbf{x}, \alpha) \triangleq f(\mathbf{x}) + \alpha g(\mathbf{x}),$$

and attempt an unconstrained optimization problem, for which the optimum is determined by

$$\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}^*, \alpha) = 0$$
$$\alpha^*(g(\mathbf{x}^*) - \eta) = 0.$$

The second of the above equations is also intuitive. If the optima is inside the feasible region, $\alpha^* = 0$. If not, we seek the optima on the constraint boundary, and therefore $g(\mathbf{x}^*) - \eta = 0$.

Equivalence of constrained and unconstrained optimization solutions
Consider the two problems:

$$\min_{\mathbf{x}_1} \ \mathcal{F}(\mathbf{x}_1, \lambda) = \|\mathbf{A}\mathbf{x_1} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{x}_1\|_2^2 \tag{4.50}$$

$$\min_{\mathbf{x}_2} \ \|\mathbf{A}\mathbf{x_2} - \mathbf{y}\|_2^2 \ \ s.t. \ \ \|\mathbf{x}_2\|_2^2 \leq \eta \tag{4.51}$$

Given $\lambda \geq 0$, the optimal solution for the first problem is given by:

$$\nabla_{\mathbf{x}_1}\mathcal{F}(\mathbf{x}_1^*, \lambda) = 0. \tag{4.52}$$

Let us call this as $\mathbf{x}_1^*(\lambda)$.
For the second problem, the optimal conditions are:

$$\nabla_{\mathbf{x}_2}\mathcal{L}(\mathbf{x}_2^*, \alpha^*) = 0 \ \text{ and } \ \alpha^*(\|\mathbf{x}_2^*\|_2^2 - \eta) = 0,$$

where $\mathcal{L}(\mathbf{x}_2, \alpha) = \|\mathbf{A}\mathbf{x_2} - \mathbf{y}\|_2^2 + \alpha(\|\mathbf{x}_2\|_2^2 - \eta)$.
Define $\eta \triangleq \|\mathbf{x}_1^*(\lambda)\|_2^2$. Then $\alpha^* \triangleq \lambda$ and $\mathbf{x}_2^* \triangleq \mathbf{x}_1^*(\lambda)$ can be verified to satisfy the optimality conditions for problem 2.
Thus, problems have the same solution (under the above definitions).

An example

Fig. 4.6 corresponds to

$$\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \ \ s.t. \ \ \|\mathbf{x}\|_2^2 \leq 1,$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 1 \\ 6 & 2 \end{bmatrix} \ \text{ and } \ \mathbf{y} = \begin{bmatrix} 6 \\ 7 \\ 4 \end{bmatrix}.$$

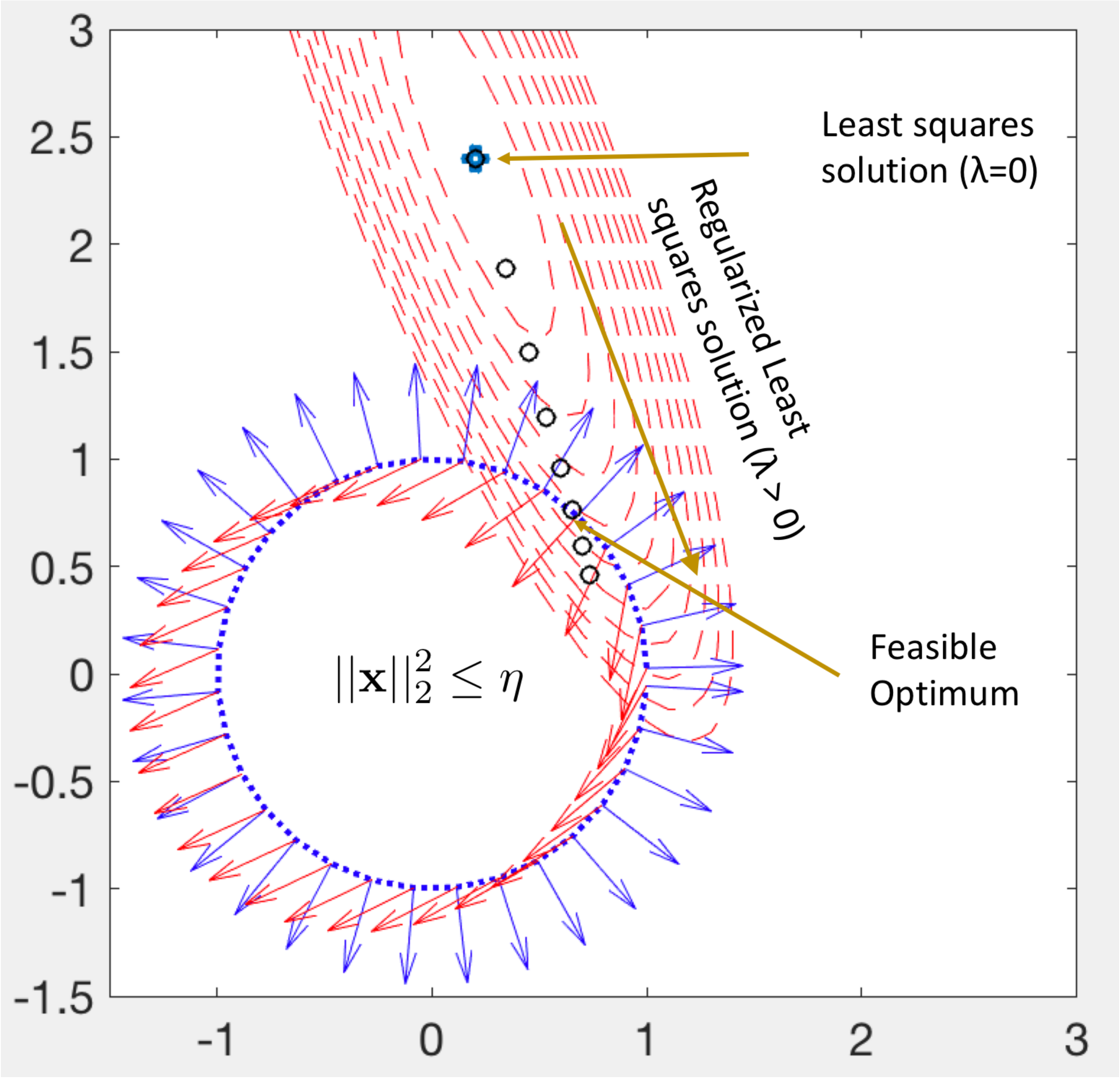


Figure 4.6: Demonstration of constrained and unconstrained optimization problems in $\mathbf{x}$ space. Dashed red lines: Contours of $f(\mathbf{x})$. Dashed blue lines: Constraint boundary $g(\mathbf{x}) - \eta = 0$. Blue arrows: $\nabla_{\mathbf{x}} g(\mathbf{x})$ (normalized) on the constraint boundary. Red arrows: $\nabla_{\mathbf{x}} f(\mathbf{x})$ (normalized) on the constraint boundary. Black circles: $\min_{\mathbf{x}} f(\mathbf{x}) + \lambda g(\mathbf{x})$ for a few select λ.
$f(\mathbf{x}) \triangleq \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2$, $g(\mathbf{x}) \triangleq \|\mathbf{x}\|_2^2$, and $\eta \triangleq 1$.

A further note on the Lagrangian dual and constrained optimization
Consider the following constrained optimization problem:

$$\min_{\mathbf{x}} \; f(\mathbf{x}) \quad s.t. \quad g(\mathbf{x}) \leq \eta. \tag{4.53}$$

Assuming a feasible minima exists for this problem (which is guaranteed for a strictly convex problem), let's call it $\mathbf{x}_p^*$. Let's form a Lagrangian

$$\mathcal{L}(\mathbf{x}, \alpha) = f(\mathbf{x}) + \alpha(g(\mathbf{x}) - \eta), \tag{4.54}$$

where $\alpha \geq 0$. For convenience, define another function $\mathcal{M}(\alpha) = \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \alpha)$.
For a given $\mathbf{x}$, if the constraints are satisfied, then $\mathcal{L}(\mathbf{x}, \alpha) \leq f(\mathbf{x})$. Clearly, $\mathcal{M}(\alpha) \leq f(\mathbf{x})$ and $\mathcal{M}(\alpha) \leq f(\mathbf{x}_p^*)$.
If the constraints are satisfied, we can go about minimizing $\mathcal{L}(\mathbf{x}, 0)$. If the constraints are not satisfied, then the penalty term can be activated by having $\alpha > 0$. Therefore, one could solve the following problem:

$$\min_{\mathbf{x}} \max_{\alpha} \mathcal{L}(\mathbf{x}, \alpha).$$

This is perhaps not a practical choice as the maximization in the inner loop may want to go to $\alpha \to \infty$ when the constraints are violated.
What if we switch to the following so-called dual problem?

$$\max_{\alpha} \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \alpha) = \max_{\alpha} \mathcal{M}(\alpha).$$

If the solution to the above dual problem is α_d^*, then

$$\mathcal{M}(\alpha_d^*) \leq f(\mathbf{x}_p^*).$$

It can be shown that equality is achieved if the optimization problem is convex, and a strictly feasible point exists.
The dual formulation is easier in situations where obtaining $\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \alpha)$ is easier. $\max_{\mathbf{x}} \mathcal{M}(\alpha)$ is easy because it is a concave function.
Further details can be obtained in books *Convex Optimization, Boyd* or the more formal *Convex Optimization Theory, Bertsekas.*

4.16 Least Squares with QR

Consider a full column rank Least squares problem.

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \tag{4.55}$$

$$
\begin{aligned}
\|\mathbf{Ax}-\mathbf{y}\|_2^2 &= \|\mathbf{Q}^T(\mathbf{Ax}-\mathbf{y})\|_2^2 && (4.56)\\
&= \|\mathbf{Q}^T(\mathbf{APP^Tx}-\mathbf{y})\|_2^2 && (4.57)\\
&= \| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{P^Tx} - \mathbf{Q}^T\mathbf{y}\|_2^2 && (4.58)\\
&= \| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{b} - \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \|_2^2, && (4.59)
\end{aligned}
$$

where $\mathbf{c}$ are the first n rows of $\mathbf{Q}^T\mathbf{y}$, and $\mathbf{d}$ are the last $m-n$ rows of $\mathbf{Q}^T\mathbf{y}$, and $\mathbf{b}=\mathbf{P}^T\mathbf{x}$. Therefore, we have

$$\|\mathbf{Ax}-\mathbf{y}\|_2^2 = \| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{b} - \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \|_2^2 = \|\mathbf{R}_1\mathbf{b}-\mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2. \tag{4.60}$$

Therefore,

$$\min_{\mathbf{x}} \|\mathbf{Ax}-\mathbf{y}\|_2^2 \text{ is equivalent to } \min_{\mathbf{b}} \|\mathbf{R}_1\mathbf{b}-\mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2. \tag{4.61}$$

Therefore, $\mathbf{b}=\mathbf{R}_1^{-1}\mathbf{c}$.

4.17 References

This chapter heavily draws from

1. *The Princeton Companion to Applied Mathematics*, Higham et al., Princeton, 2015.
2. *Linear Algebra and Learning from Data*, Strang, Wellesley-Cambridge, 2019.
3. Wikipedia and the internet in general.