Model Reduction

Hans Zwart, Bernard Geurts

Department of Applied Mathematics, University of Twente,
PO Box 217, 7500 AE Enschede, The Netherlands

January 31, 2025

Contents

Chapter 1

Introduction

The field of model reduction can be motivated concisely in three steps:

1. Many problems in Science and Technology can be modeled in mathematical terms using ordinary and partial differential equations . Often, these models involve a number of variables, whose evolution depends on their dynamic coupling. This leads to considerable complexity even in rather simple applications and very quickly implies that the problem can not be solved analytically.

2. The field of numerical mathematics provides important answers to this dilemma in which we know all about the mathematical model but cannot solve the problem in closed form. In fact, using discretization in space and time, approximate discrete models can be derived and discrete solutions can be obtained which represent the unknown analytical solution with high precision. In a growing number of cases the error in the numerical solution can be estimated explicitly, yielding a corresponding level of confidence in this approach to a solution.

3. While the computational approach is highly effective in a large number of applications, also here crucial limitations need to be faced. This concerns the computational costs the numerical solution would generate in terms of computing power and data storage. The sheer number of space-time points needed to represent a relevant solution can grow beyond feasible limits, rendering the direct computational approach inadequate. This calls for a further step in which the complexity of the model is reduced deliberately to yield feasible computations on the one hand, which, on the

other hand, are nevertheless of sufficient accuracy to yield useful answers to the application-related questions at hand.

In this perspective, **model reduction** aims to formulate simpler models that provide 'enough for less', rather than to aim for much more expensive all-inclusive models that yield 'all the original model has to offer'. This illustrates the 'balancing act' that also characterizes model reduction where computational costs are weighed against a reduced outcome at reduced costs. It is part of the considerations that underpin model reduction to incorporate an intuition about the relevant balance that can be struck.

General, finite-dimensional dynamical systems can be represented in a phase space which is a subset of $\mathbb{R}^n$ where n denotes the dimension of the system, i.e., the number of grid points times the number of unknown variables in the formulation. To represent the state of the system a state-vector $\mathbf{y}$ is introduced which contains, e.g., the values of the discrete solution attained on a grid. In case of autonomous systems, the evolution can be expressed by

$$\dot{\mathbf{y}}(t) = \mathbf{F}(\mathbf{y}) \tag{1.1}$$

where $\dot{\mathbf{y}} = d\mathbf{y}/dt$ and the flow is characterized by the discrete 'flux' $\mathbf{F}$. If external forcing would be included or other effects that depend explicitly on time t one may express the dynamical system as $\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}, t)$. To complete the specification of a time-dependent solution, the initial condition $\mathbf{y}(0) = \mathbf{y}_0$ is provided. If the solution to eq. (1.1) is unique, then the above specification suffices to describe $\mathbf{y}$ for all $t \geq 0$. This deterministic view is somewhat simplistic since in actual nonlinear systems an element of chaotic behavior, due to a sensitive dependence on initial conditions, can arise which has large repercussions on the interpretation of the numerically attained solution. We illustrate this next using a famous example.

Sensitive dependence: Lorenz system

It is well known that systems of ordinary differential equations can be solved analytically only in very special cases. Most nonlinear problems are in fact non-integrable and may display sensitive dependence on initial conditions. This implies that two solutions which initially differ by only a small amount, may develop large separations already over small time intervals. Also, effects arising from the finite number of digits used to represent variables at discrete locations can contribute to such sensitive dependencies. A well known

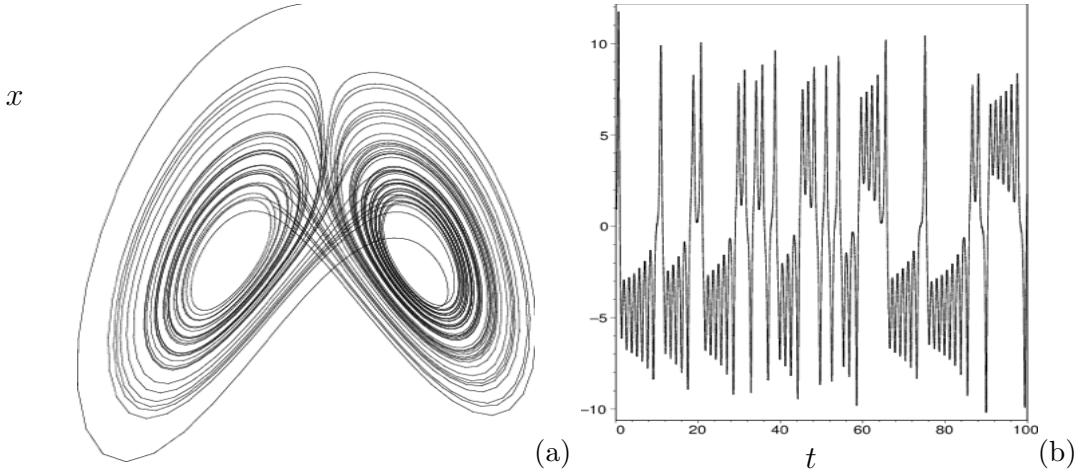


Figure 1.1: An impression of the Lorenz attractor at $a = 3$, $b = 25$ and $c = 1$; the trajectory which emanates from $x(0) = 1,\ y(0) = 0,\ z(0) = 2$ is presented in (a) and the history of the component $x(t)$ is shown in (b).

example which is illustrative for our purposes here is the Lorenz system [46] described by:

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= bx - y - xz \\ \dot{z} &= xy - cz \end{aligned} \tag{1.2}$$

with parameters a, b and c. Lorenz attempted to model the motion of a particle subject to atmospheric forces with these equations. When this system is solved numerically, it is found that the modeled particle moves in an irregular fashion. After a transient period, though, a dynamically important attractor becomes clear from the particle's motion in phase space. This attractor determines the long-time properties of all solution trajectories that originate from its basin of attraction. The individual solution components $x(t)$, $y(t)$ and $z(t)$ still behave quite erratically as functions of time, but seen as an orbit in phase space, the motion displays a striking pattern. For suitable values of the parameters, the pattern resembles a butterfly-shaped object with two 'wings', known as the Lorenz attractor. An impression is shown in figure 1.1(a) and the corresponding history of the x-component of the solution is shown in figure 1.1(b). After a sufficiently long time the trajectories of solutions to (1.2) approach the attractor arbitrarily closely while their motion continues to be unsteady.

A typical solution to (1.2) will cycle around one of the wings of the attractor for a

certain period and then quickly travel to the other wing where it again may spend some cycles before 'hopping' back to the first wing. This process continues indefinitely and is characterized by a strong sensitive dependence on initial conditions. If we consider the trajectories corresponding to two different but very close initial conditions then the distance between the solutions at later times increases very rapidly. In fact, it becomes impossible to predict with certainty on which of the two wings the solution will be located at some time $t_* \gg 1$, due to the finite accuracy of any numerical evaluation of the solution and the accumulated 'phase-error'.

Attractors and structural stability

In view of the sensitive dependence of trajectories on initial conditions and accumulated round-off errors, one may at best hope to approximate properties of the entire attractor instead of simulating individual solutions for all times. Also, initial conditions that correspond to a particular problem are typically not known in great detail and, coupled to the sensitive dependencies, this by itself creates significant uncertainty about the interpretation of a specific simulated solution, especially concerning details of its long-time behavior.

The accumulated round-off errors may cause deviations that tend to take the solution away from the attractor. However, the inherent attracting properties of the Lorenz attractor will limit these 'excursions' to a small band around this object. Thus, although individual exact trajectories are not likely to be available, the numerically generated impression will remain close to the attractor and consequently one may still accurately obtain certain average properties of the entire attractor. Likewise, perturbations of the parameters $a - c$ will give rise to changes in the shape of the corresponding attractor. However, as long as these perturbations are sufficiently small, the dynamics will typically remain structurally stable. By this we imply that properties of the perturbed attractor such as its dimension, its location and size or the fraction of time spent in certain regions on the attractor, will also deviate only by a limited amount. Even though individual trajectories obtained from the same initial condition will deviate tremendously in the perturbed system after some time, the concept of a structurally stable attractor provides a setting in which robust predictions of global properties of a nonlinear dynamical system are still possible. In contrast, there are also certain perturbations that may alter the dynamical behavior considerably. By analogy, this provides some 'hope' for successful model reduction in which it is not even the aim to approximate details of individual trajectories, but rather capture important features of the underlying structurally stable attractor.

The general philosophy behind model reduction provides a modeling framework for a wide variety scientific and technological problems. An important class of such problems is related to heat transfer , e.g., where cooling of electronics or food items is concerned.

- **Keeping food cool**

  After harvesting, the harvest products have to be kept fresh till the next harvest. Many farming produce are stored in large barns. In these barns, the temperature and humidity has to controlled to stay within bounds. This is done placing ventilators and cooling elements. The ventilators are mounted on a refrigerators element, and in this way the air inside the barns can be controlled, see also Figure 1.2. The control variables are the temperature of the refrigerator element and the ventilation speed. Note that we are storing lying organisms and so they produce heat.

Figure 1.2: Storing onions

It is clear that the temperature inside the produce will vary with time and position. Closer to the ventilators the onions/potatoes will be cooler than at the bottom of the containers. Given the spatial and temporal dependence of our to-be-controlled variable, it may come as no surprise that a model describing the temperature distribution in the barn and produce will be a partial differential equation. However, as with the temperature control in your house, you don't need to know the temperature everywhere, and it is even impossible to measure the temperature of every potato/onion. Moreover, the control actions are limited. For instance, in the barn, you cannot switch the fan on and off every minute. To design a implementable controller, a

simpler model suffices. As a first step, it was chosen to simplify the situation. The adjusted set-up is schematically given in Figure 1.3.

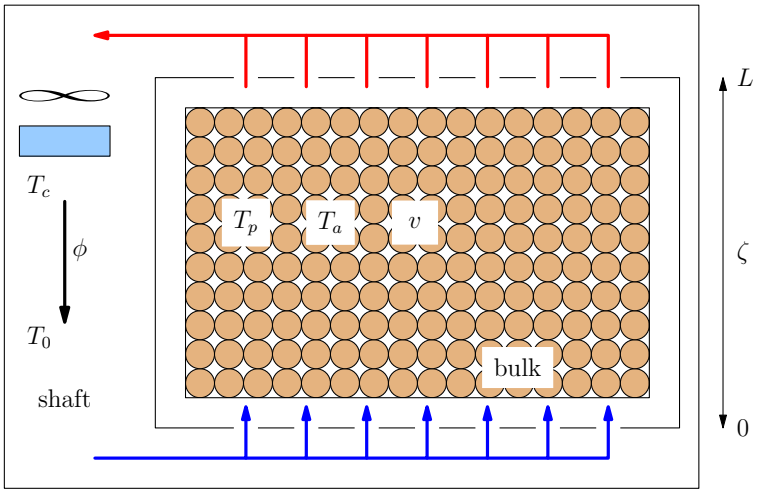


Figure 1.3: Schematic representation

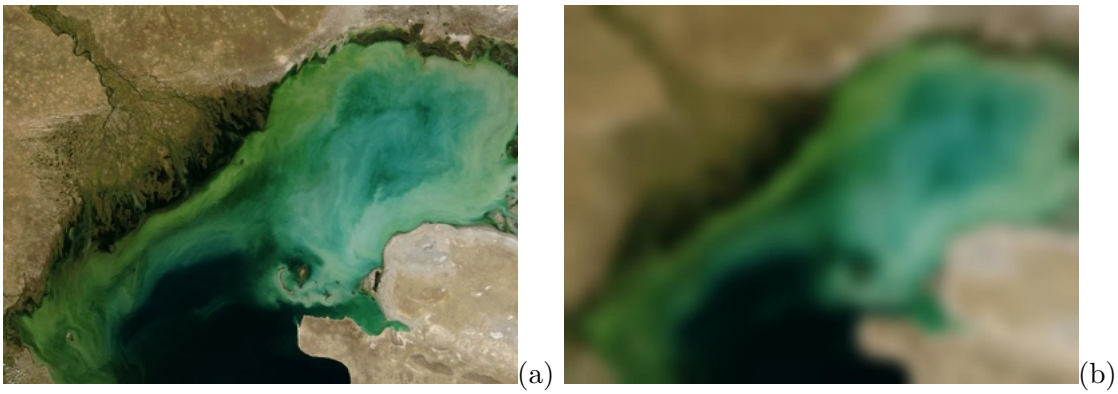
So from a 3-D configuration we went to 1-D, by considering only the height as spatial variable. Even the partial differential equation associated to this simplified set-up was not needed to design a controller. With system theoretic techniques the model was further simplified. This (very) simplified model was used to construct a controller.

Other, highly challenging problems in Science and Technology are related to fluid mechanics . The flow of air and water has long intrigued scientists and engineers. Mastering such flow problems is crucial for many applications, such as in aerodynamics, the design of our waterways, the development of industrial processes, and the long-time development of our climate. Developing models that are 'fit for purpose' for these crucial societal problems is often all that is achievable, as full resolution of all flow physics is infeasible. An important example is

- **Turbulence**

  Turbulent flow is all around us, having beneficial as well as life-threatening consequences, depending on the circumstances and application. While turbulence enhances mixing, which often is highly desired, it also may induce large forces on buildings, airplanes and bridges which could lead to failure of these structures. In figure 1.4

satellite images of algea bloom in the Caspian Sea, at high and at low spatial resolutions are depicted. Obviously, much of the small-scale details can be appreciated only at high resolution. However, the similarity between the high-resolution image and the spatially filtered one is so striking, that it is conceivable that various important questions related to this algae bloom can be answered with confidence also when only the filtered representation would be available. This sets the stage for model reduction in turbulence, often known as 'large-eddy simulation' (LES) .



(a) (b)

Figure 1.4: Resolved (a) and coarsened (b) impressions of algae blooms in the Caspian sea.

The organization of these lecture notes is as follows. In Chapter 2 the method of balancing to arrive at a coarsened model approximation is introduced and discussed in detail. This concerns model reduction for linear problems. Chapter 3 is devoted to the method of spatial filtering to achieve coarsened models for partial differential equations (PDEs). The filtering approach can be applied to linear PDEs, in which case the initial and boundary conditions need investigation. Filtering can, however, also be applied to nonlinear PDEs, leading to a so-called closure problem that needs to be addressed. Finally, spatial and temporal discretisations are introduced for the coarsened models.

Chapter 2

Model approximation using balancing

2.1 Introduction

In the course *Linear System Theory* we have seen the following model class

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad x(0) = x_0, \tag{2.1}$$

$$y(t) = Cx(t). \tag{2.2}$$

Here[1] $x(t) \in \mathbb{C}^n$ is the *state*, $u(t) \in \mathbb{C}^m$ the *input*, and $y(t) \in \mathbb{C}^p$ is the *output*. We call (2.1)–(2.2) a *state linear system*. All linear, time-invariant ordinary differential equations can be written in this standard form, and so this is a general model class. The solution of (2.1) is given by

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau. \tag{2.3}$$

From this and (2.2) it follows directly that

$$y(t) = Ce^{At}x_0 + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau. \tag{2.4}$$

When we define the function

$$h(t) = \begin{cases} Ce^{At}B & t \geq 0 \\ 0 & t < 0, \end{cases} \tag{2.5}$$

[1] For inconvenience we take our spaces complex-valued. However, we could have taken them real-valued.

then for $x_0 = 0$ equation (2.4) can be written (please check)

$$y(t) = \int_0^\infty h(t-\tau)u(\tau)d\tau.$$

This is known as the *convolution product* of h and u. The function h as defined in (2.5) is called the *impulse response* of the system (2.1)–(2.2), or simply the impulse response.

In *Linear Systems Theory* there are some concepts treated which we need in the sequel. The first one being stability, see also [2, Section 2.3].

Definition 2.1.1 *Consider the differential equation (2.1) with u (identically) zero. We say that the solutions of this differential equation are* stable *if they converge to zero as $t \to \infty$.*

With abuse of notation, we often speak of *the system* (2.1) or *the system* (2.1)–(2.2) *being stable*. By which we mean that the solutions of the differential equation $\dot{x}(t) = Ax(t)$ are stable.

Hence the differential equation $\dot{x}(t) = Ax(t)$ is stable if and only if for all $x_0 \in \mathbb{C}^n$ there holds $e^{At}x_0 \to 0$ as $t \to \infty$. We have the following characterisation of stability.

Theorem 2.1.1 *The differential equation $\dot{x}(t) = Ax(t)$ is stable if and only if the matrix A is* Hurwitz. *That is all its eigenvalues have negative real part.*

The other two fundamental concepts we need are controllability and observability, see [2, Chapter 3].

Definition 2.1.2 *The system (2.1) is* controllable *if and only if for any two elements in the state space $\mathbb{C}^n$, x_0 and x_1, there exists a $t_1 > 0$ and an input u such that the solution of (2.1) satisfies $x(0) = x_0$ and $x(t_1) = x_1$.*

Definition 2.1.3 *The system (2.1)–(2.2) is* observable *if for u identically zero the output trajectory $\{y(t) \mid t \geq 0\}$ uniquely determines the initial state x_0. Thus if for all $t \geq 0$ there holds that $Ce^{At}x_0 = Ce^{At}\tilde{x}_0$, then $x_0 = \tilde{x}_0$.*

For controllability and observability there are simple matrix tests to check them.

Theorem 2.1.2 *The following equivalences holds;*

- *The system (2.1) is controllable if and only if the matrix*

$$\begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$$

  has full rank.

- *The system (2.1)–(2.2) is observable if and only if the matrix*

$$\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has full rank.

For a system which is stable, we can formulate two other equivalent conditions.

Theorem 2.1.3 *If the system (2.1)–(2.2) is stable, then the following holds*

- *The system (2.1) is controllable if and only if the matrix*

$$P := \int_0^\infty e^{At} B B^* e^{A^* t} dt \tag{2.6}$$

is invertible.

- *The system (2.1)–(2.2) is observable if and only if the matrix*

$$Q := \int_0^\infty e^{A^* t} C^* C e^{At} dt \tag{2.7}$$

is invertible.

The matrix P of (2.6) is called the *controllability gramian*, the matrix Q of (2.7) is called the *observability gramian*.

It may seem that to calculate P and Q we have first to solve the differential equation, or at least calculate e^{At}. This is not the case. They can be calculated by solving linear equations, known as *Lyapunov equations*.

Theorem 2.1.4 *Assume that the system is stable, i.e, all eigenvalues of A have negative real part. Then the following hold:*

- *The controllability gramian of (2.6) is the unique solution of the Lyapunov equation*

$$AP + PA^* = -BB^*. \tag{2.8}$$

- *The observability gramian of (2.7) is the unique solution of the Lyapunov equation*

$$A^*Q + QA^* = -C^*C. \tag{2.9}$$

Proof We will only prove the first part, since the second goes very similar. We have to prove two assertions, namely the P as defined in (2.6) satisfies (2.8), and conversely when a matrix $\tilde{P}$ satisfies (2.8), then $\tilde{P}$ equals P from (2.6).

We begin by showing that the controllability gramian satisfies (2.8). Using (2.6) we find

$$\begin{aligned} AP + PA^* &= A\int_0^\infty e^{At}BB^*e^{A^*t}dt + \int_0^\infty e^{At}BB^*e^{A^*t}dtA^* \\ &= \int_0^\infty \left[Ae^{At}BB^*e^{A^*t} + e^{At}BB^*e^{A^*t}A^*\right]dt. \end{aligned} \tag{2.10}$$

We know that $\frac{d}{dt}e^{At} = Ae^{At}$ and, by taken the complex conjugate, that $\frac{d}{dt}e^{A^*t} = e^{A^*t}A^*$. By the product rule, we find

$$\frac{d}{dt}\left[e^{At}BB^*e^{A^*t}\right] = Ae^{At}BB^*e^{A^*t} + e^{At}BB^*e^{A^*t}A^*.$$

Substituting this in (2.10), we obtain

$$AP + PA^* = \int_0^\infty \frac{d}{dt}\left[e^{At}BB^*e^{A^*t}\right]dt = \left[e^{At}BB^*e^{A^*t}\right]_0^\infty = -BB^T,$$

where we have used that the system is stable. So we have proved that P satisfies the Lyapunov equation (2.8).

Let now $\tilde{P}$ be a solution of the Lyapunov equation (2.8), then for all $t \geq 0$

$$e^{At}\left[A\tilde{P} + \tilde{P}A^*\right]e^{A^*t} = -e^{At}BB^*e^{A^*t}.$$

Similar as in the first part, we have that

$$e^{At}\left[A\tilde{P} + \tilde{P}A^*\right]e^{A^*t} = \frac{d}{dt}\left[e^{At}\tilde{P}e^{A^*t}\right].$$

So

$$-\int_0^{t_1} e^{At}BB^Te^{A^*t}dt = \int_0^{t_1}\frac{d}{dt}\left[e^{At}\tilde{P}e^{A^*t}\right]dt = e^{At_1}\tilde{P}e^{A^*t_1} - \tilde{P}.$$

Using once more that the system is stable gives, by letting t_1 approach infinity, that $\tilde{P} = \int_0^\infty e^{At}BB^*e^{A^*t}dt = P$. □

2.1.1 Questions

1. In this exercise we investigate how the systems matrices, etc. change over a basis transformation. Consider the system (2.1)–(2.2) on which we apply the state transformation $z(t) = Tx(t)$, with T an invertible matrix.

   (a) Formulate the system (2.1)–(2.2) with this (transformed) state.

   (b) Find the solutions to the Lyapunov equation (2.8) and (2.9) for the transformed systems. Note that these gramians transform differently.

2.2 Transfer functions

In *Signals and Transforms*, [3, Section 3.7] the concept of a transfer function is introduced. There it is introduced for an ordinary differential equation. Since we are working with the state-differential equation (2.1)–(2.2), we have to define transfer functions for a larger class. Instead of defining it directly for (2.1)–(2.2), we do it for a very general class of systems. Namely, we see a system as a set of trajectories, in the following way. Let $\mathbb{T} \subseteq \mathbb{R}$ be the time axis, U, X, and Y the input, state, and output space, respectively. A *general system* is a subset of $\{(u, x, y) \mid u : \mathbb{T} \mapsto U, x : \mathbb{T} \mapsto X, y : \mathbb{T} \mapsto Y\}$. The elements of this subset are called solutions.

Definition 2.2.1 *Consider a general system with input u (taking values in U), state x and output y, defined on the time-axis $\mathbb{T}$. Let $\Omega \subseteq \mathbb{C}$. If for every $u_0 \in U$ and $s \in \Omega$ there exists a solution $(u(t), x(t), y(t)), t \in \mathbb{T}$ of the form $(u(t), x(t), y(t)) = (u_0 e^{st}, x_0 e^{st}, y_0 e^{st})$ and this solution is unique, then the map $u_0 \mapsto y_0$ is called the* transfer function *on Ω. We normally denote this transfer function by $G(s)$.*

Note that normally, Ω is chosen to be as large as possible. We will calculate the transfer function of our state linear system (2.1)–(2.2). However, before we do so, we will do it for the ordinary differential equation

$$y^{(n)}(t) + p_{n-1}y^{(n-1)}(t) + \cdots + p_1 y^{(1)}(t) + p_0 y(t) = q_m u^{(m)}(t) + q_{m-1}u^{(m-1)}(t) + \cdots + q_1 u^{(1)}(t) + q_0 u(t). \tag{2.11}$$

We begin by writing this as a general system. We choose as time axis $\mathbb{T} = [0, \infty)$.

$$\mathcal{B}_{ode} := \{(u, y) \mid u : [0, \infty) \mapsto \mathbb{C}, y : [0, \infty) \mapsto \mathbb{C}, \text{ s.t. } u \text{ is } m\text{-times differentiable}, \\ y \text{ is } n \text{ times differentiable, and (2.11) is satisfied.}\}$$

So in this equation there is no "x". Note that we could as well have chosen $\mathbb{R}$ as our time axis.

To calculate the transfer function, we choose an $u_0 \in \mathbb{C}$, and try to find a solution of the form $(u(t), y(t)) = (u_0 e^{st}, y_0 e^{st})$ in $\mathcal{B}_{ode}$. These are C^∞-functions, so we only have to check that $(u_0 e^{st}, y_0 e^{st})$ satisfies the equation (2.11). Since the k-th derivative of e^{st} equals $s^k e^{st}$, this is equivalent to

$$y_0 s^n e^{st} + p_{n-1} y_0 s^{n-1} e^{st} + \cdots + p_1 y_0 s e^{st} + p_0 y_0 e^{st} = \\ q_m u_0 s^m e^{st} + q_{m-1} u_0 s^{m-1} e^{st} + \cdots + q_1 u_0 s e^{st} + q_0 u_0 e^{st}.$$

Since e^{st} is never equal to zero, we may divide by it. Furthermore, we may collect the y_0 and u_0-terms. By doing so, we find the equation

$$\left[s^n + p_{n-1} s^{n-1} + \cdots + p_1 s + p_0\right] y_0 = \left[q_m s^m + q_{m-1} s^{m-1} + \cdots + q_1 s + q_0\right] u_0.$$

Recall that u_0 was given, and we want to find y_0. From the above this is possible if and only if $s \in \mathbb{C}$ is such that $s^n + p_{n-1} s^{n-1} + \cdots + p_1 s + p_0 \neq 0$. So for those s we find

$$y_0 = \frac{q_m s^m + q_{m-1} s^{m-1} + \cdots + q_1 s + q_0}{s^n + p_{n-1} s^{n-1} + \cdots + p_1 s + p_0} u_0 = G(s) u_0. \tag{2.12}$$

This equals the formula for the transfer function of the ordinary differential equation (2.11) given in [2, Theorem 3.7.1].

Using a similar construction, we can find the transfer function of the linear system (2.1)–(2.2).

Theorem 2.2.1 *The transfer function of the linear system (2.1)–(2.2) is given by*

$$G(s) = C(sI - A)^{-1} B, \tag{2.13}$$

for all $s \in \mathbb{C}$ not an eigenvalue of A.

Proof We begin by writing this as a general system. We choose as time axis $\mathbb{T} = [0, \infty)$.

$$\mathcal{B}_{ls} := \{(u, x, y) \mid u : [0,\infty) \mapsto \mathbb{C}^m, x : [0,\infty) \mapsto \mathbb{C}^n, y : [0,\infty) \mapsto \mathbb{C}^p, \text{ with } x \text{ differentiable,} \\ \text{and there exists an } x_0 \text{ s.t. (2.1) and (2.2) are satisfied.}\}$$

To determine the transfer function we have, for a given $s \in \mathbb{C}$ and $u_0 \in \mathbb{C}^m$, to find a triple (u, x, y) of the form $(u(t), x(t), y(t)) = (u_0 e^{st}, x_0 e^{st}, y_0 e^{st})$ in $\mathcal{B}_{ls}$. It is easy to see that this is satisfied if and only if for all $t \geq 0$

$$x_0 s e^{st} = A x_0 e^{st} + B u_0 e^{st}, \quad y_0 e^{st} = C x_0 e^{st}.$$

Since e^{st} is never equal to zero, this can equivalently be written as

$$x_0 s = Ax_0 + Bu_0, \quad y_0 = Cx_0.$$

From the second equation, we see that once x_0 is found y_0 will follow. The first equation is rewritten as

$$(sI - A)x_0 = Bu_0$$

which is solvable when the matrix $sI - A$ is invertible. Since that holds if and only if s is not an eigenvalue, we find for those s

$$x_0 = (sI - A)^{-1}Bu_0$$

and thus $y_0 = Cx_0 = C(sI - A)^{-1}Bu_0$, which proves the assertion. □

In the following theorem we show that properties of the system lead to properties of the transfer function.

Theorem 2.2.2 *Let P be a symmetric non-negative matrix, i.e., $x^*Px \geq 0$ for all $x \in \mathbb{C}^n$. Suppose that along solutions of (2.1)–(2.2) the following inequality holds*

$$\frac{d}{dt}\left[x(t)^*Px(t)\right] \leq -\alpha\|y(t)\|^2 + \beta\|u(t)\|^2, \tag{2.14}$$

for some $\alpha, \beta \in [0, \infty)$. Then for $u_0 \in \mathbb{C}^m$, $s \in \mathbb{C}$ with $Re(s) \geq 0$ and s not an eigenvalue of A,

$$\alpha\|G(s)u_0\|^2 \leq \beta\|u_0\|^2. \tag{2.15}$$

Proof The transfer function is constructed from solutions of the form $(u(t), x(t), y(t)) = (u_0e^{st}, x_0e^{st}, y_0e^{st})$. We know that for s not an eigenvalue of A, the transfer function exists for that s. Substitution, this exponential solution in the inequality (2.14) gives for $t \geq 0$

$$\frac{d}{dt}\left[x_0^*e^{\bar{s}t}Px_0e^{st}\right] \leq -\alpha\|G(s)u_0e^{st}\|^2 + \beta\|u_0e^{st}\|^2,$$

Now $|e^{st}|^2 = e^{\bar{s}t}e^{st} = e^{2\mathrm{Re}(s)t}$ and using that, the above relation becomes

$$2\mathrm{Re}(s)e^{2\mathrm{Re}(s)t}x_0^*Px_0 \leq -\alpha\|G(s)u_0\|^2e^{2\mathrm{Re}(s)t} + \beta\|u_0\|^2e^{2\mathrm{Re}(s)t}.$$

Since $e^{2\mathrm{Re}(s)t}$ is always unequal zero, we can simplify this to

$$2\mathrm{Re}(s)x_0^*Px_0 \leq -\alpha\|G(s)u_0\|^2 + \beta\|u_0\|^2,$$

or

$$2\mathrm{Re}(s)x_0^*Px_0 + \alpha\|G(s)u_0\|^2 \leq \beta\|u_0\|^2.$$

Since P is a non-negative matrix, we find for $\mathrm{Re}(s) \geq 0$ that the first term is always non-negative, and so (2.15) follows. □

2.2.1 Questions

1. Consider the state linear systems

$$\dot{x}_1(t) = A_1x_1(t) + B_1u(t), \quad y_1(t) = C_1x_1(t), \text{ and} \tag{2.16}$$

$$\dot{x}_2(t) = A_2x_2(t) + B_2u(t), \quad y_2(t) = C_2x_2(t) \tag{2.17}$$

with $u \in \mathbb{C}^m$, $y_1, y_2 \in \mathbb{C}^p$ and x_1, x_2 in $\mathbb{C}^{n_1}$ and $\mathbb{C}^{n_2}$, respectively. The corresponding transfer function are $G_1(s)$ and $G_2(s)$.

Consider the linear state space system

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t), \tag{2.18}$$

$$y(t) = \begin{bmatrix} C_1 & -C_2 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}. \tag{2.19}$$

Prove that its transfer function equals $G_1(s) - G_2(s)$ for $s \in \mathbb{C}$ not an eigenvalue of A_1 or A_2.

2.3 The idea behind balancing

If a system is not observable, then there exists a non-zero $v \in \mathbb{C}^n$ such that $Ce^{At}v = 0$ for all $t \geq 0$. So you cannot observe this initial state if the only information you get from the system is via the outputs. When your goal is to reduce the state space, then you would kick this state out. Now assume that the system is observable, but there is a state v such that the corresponding output $y(t) = Ce^{At}v$ is very small, then you would also like to exclude this state from your state space as well. This is half the idea behind balancing. Balancing is that you do it also for the input side.

So assume that the system is not controllable, then there exits a state which you cannot reach from the origin. From *Linear Systems*, we know that the solution of

$$\dot{x}(t) = Ax(t) + Bu(t), \qquad x(t_0) = x_0 \tag{2.20}$$

(so with initial time t_0) is given by

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau. \tag{2.21}$$

Let in this equation the initial time t_0 be negative, the initial state $x(t_0) = 0$, and let $t = 0$. Then

$$x(0) = \int_{t_0}^{0} e^{A(-\tau)} B u(\tau) d\tau. \tag{2.22}$$

So if the system is not controllable, then there exists a state, let us call this x_1, which you cannot reach from the origin. Hence for all inputs $u : [t_0, 0] \mapsto \mathbb{C}^m$ you will have that the $x(0) \neq x_1$. Again if you look to the system from an input-output point of view, you would exclude this state.

If the system is controllable, but to reach the given state x_1 would require a lot of input energy, then it is very unlikely that in a practical situation this state would be reached, and so you would like to exclude it as well.

So we see that we would like to exclude states that are either hard to see, i.e., the corresponding output $y(t) = Ce^{At}x_0$ is very small, or take a lot of input energy to reach.

The idea behind balancing is precisely that; remove states for which the output is small and/or which need a lot of energy to reach. This may sound convincing, but what do we mean by "small" and "a lot"? These terms have to make more precise.

With the integral substitution, $s = -\tau$, the integral (2.22) becomes (note that we have chosen $t_0 < 0$)

$$x(0) = \int_{0}^{-t_0} e^{As} B u(-s) ds.$$

Since we want to have an expression independent of the final time t_0, we write this as

$$x(0) = \int_{0}^{\infty} e^{As} B v(s) ds \tag{2.23}$$

with

$$v(s) = \begin{cases} u(-s) & s \in [0, -t_0] \\ 0 & s > -t_0. \end{cases}$$

From now on we work with the expression (2.23). Next we have to give a mathematical meaning to "input energy". We do that by taking the $L^2(0, \infty)$-norm on u, see also [3, Section 1.5]. So

$$\|u\| := \sqrt{\int_{0}^{\infty} \|u(t)\|^2 dt}, \tag{2.24}$$

where the norm within the integral is the normal (Euclidean) $\mathbb{C}^m$-norm.

So using (2.23) and (2.24) we can now say that state x_0 is harder to reach than state x_1. Namely, when for every input v satisfying $x_0 = \int_0^\infty e^{As}Bv(s)ds$ there exists an input u satisfying $x_1 = \int_0^\infty e^{As}Bu(s)ds$ and $\|u\| < \|v\|$.

It remains to define what we mean with a "small output". We know that the output (for $u = 0$) is given as $y(t) = Ce^{At}x_0$. Similar as for the input we take the $L^2(0,\infty)$-norm as a measure of "how large" y is, so

$$\|y\| := \sqrt{\int_0^\infty \|y(t)\|^2 dt}, \tag{2.25}$$

where the norm within the integral is the normal (Euclidian) $\mathbb{C}^p$-norm.

So the idea is to remove states that give little output energy $\|y\|$ and states that need a lot of input energy $\|u\|$ to reach.

2.4 Balancing

In the previous section we explained the idea behind balancing, and in this section we will do it. However, there are a few things that we have to realise first. Namely, with zero input, the output corresponding to $2x_0$ is always twice the output corresponding to x_0. Hence the energy will also be twice as large. Since we want to keep a linear state space, we cannot remove $2x_0$ and not x_0. Furthermore, it would not make sense. It is logical that larger states produce more output energy than smaller states. So we normalise the states, and will only look at states of norm one, i.e., $\|x_0\| = 1$.

Similarly, since $2u$ will reach $2x_0$ when u reaches x_0, we also want to normalise the states. Here "u reaches x_0" means that $x_0 = \int_0^\infty e^{As}Bu(s)ds$.

On the input side, there is another complicating factor. Like you can go to the university via a detour, there are almost always more inputs that reach the same state. Given $x_0 \in \mathbb{C}^n$ we define the set of all inputs reaching that state as

$$V_{x_0} = \{u \in L^2((0,\infty);\mathbb{C}^m) \mid x_0 = \int_0^\infty e^{As}Bu(s)ds\}. \tag{2.26}$$

Like you would say that you live close to the university, when the shortest path from your home to the university takes little time, we say that the state x_0 needs little input energy to reach when there exists an input $u \in V_{x_0}$ for which $\|u\|$ is small. So we are interested in the infimum norm of all $u \in V_{x_0}$.

Theorem 2.4.1 *Assume that the system (2.1) is controllable and that A is Hurwitz. By P we denote the controllability gramian, see (2.6). Then for every $x_0 \in \mathbb{C}^n$, the following hold;*

1. *The set V_{x_0}, see (2.26), is non-empty;*
2. *The function u_{opt} defines as $u_{opt}(t) = B^* e^{A^* t} P^{-1} x_0$ is in V_{x_0}, and $\|u_{opt}\|^2 = x_0^* P^{-1} x_0$;*
3. *The input u_{opt} satisfies $\|u_{opt}\| \leq \|u\|$ for all $u \in V_{x_0}$. Furthermore, it is unique input with this optimality property.*

Proof *Item 1.* Since the system (2.1) is controllable, there exists a time $t_1 > 0$ and a u_1 such that x_0 can be reached from zero, i.e., $x_0 = \int_0^{t_1} e^{A(t_1-\tau)} B u_1(\tau) d\tau$. Next define $u \in L^2((0,\infty); \mathbb{C}^m)$ as

$$u(t) = \begin{cases} u_1(t_1 - t) & t \in [0, t_1] \\ 0 & t > t_1. \end{cases}$$

Then it is easy to see that $u \in V_{x_0}$.

Item 2. We have to show that $u_{opt}(t) \in V_{x_0}$. Thus that $\int_0^\infty e^{As} B u_{opt}(s) ds = x_0$. Using the definition of u_{opt} and P, see (2.6), we find

$$\begin{aligned} \int_0^\infty e^{As} B u_{opt}(s) ds &= \int_0^\infty e^{As} B B^* e^{A^* s} P^{-1} x_0 ds \\ &= \int_0^\infty e^{As} B B^* e^{A^* s} ds P^{-1} x_0 = P P^{-1} x_0 = x_0, \end{aligned}$$

which proves the first assertion. It remains to calculate its norm.

$$\begin{aligned} \|u_{opt}\|^2 &= \int_0^\infty \|u_{opt}(t)\|^2 dt = \int_0^\infty u_{opt}(t)^* u_{opt}(t) dt \\ &= \int_0^\infty \left[B^* e^{A^* t} P^{-1} x_0 \right]^* B^* e^{A^* t} P^{-1} x_0 dt = \int_0^\infty x_0^* P^{-1} e^{At} B B^* e^{A^* t} P^{-1} x_0 dt \\ &= x_0^* P^{-1} \int_0^\infty e^{At} B B^* e^{A^* t} dt P^{-1} x_0 = x_0^* P^{-1} P P^{-1} x_0 = x_0^* P^{-1} x_0, \end{aligned}$$

where we have used that P, and thus P^{-1}, is symmetric, and its definition.

Item 3. Now we prove the main part of this theorem. Instead of minimising $\|u\|$ we minimise

$\|u\|^2$. Using the definition of the $L^2(0,\infty)$-norm the following equalities follow.

$$\begin{aligned}\|u\|^2 &= \|u - u_{opt} + u_{opt}\|^2 \\ &= \|u - u_{opt}\|^2 + \|u_{opt}\|^2 + \\ &\quad \int_0^\infty (u(t) - u_{opt}(t))^* u_{opt}(t)dt + \int_0^\infty u_{opt}(t)^*(u(t) - u_{opt}(t))dt. \end{aligned} \tag{2.27}$$

Now it is interesting to look at the two last terms. In particular, we will look at $u_{opt}(t)^*(u(t)-u_{opt}(t))$. Using the expression for u_{opt} we find

$$u_{opt}(t)^*(u(t) - u_{opt}(t)) = \left(B^* e^{A^* t} P^{-1} x_0\right)^* (u(t) - u_{opt}(t)) = x_0^* P^{-1} e^{At} B(u(t) - u_{opt}(t)),$$

where we have used that P is symmetric. So

$$\begin{aligned}\int_0^\infty u_{opt}(t)^*(u(t) - u_{opt}(t))dt &= \int_0^\infty x_0^* P^{-1} e^{At} B(u(t) - u_{opt}(t))dt \\ &= x_0^* P^{-1} \int_0^\infty e^{At} B(u(t) - u_{opt}(t))dt \\ &= x_0^* P^{-1} \int_0^\infty e^{At} B u(t)dt - x_0^* P^{-1} \int_0^\infty e^{At} B u_{opt}(t))dt \\ &= x_0^* P^{-1} x_0 - x_0^* P^{-1} x_0 = 0,\end{aligned}$$

where we have used that both u and u_{opt} are in V_{x_0}.

Since

$$\int_0^\infty (u(t) - u_{opt}(t))^* u_{opt}(t)dt = \left[\int_0^\infty u_{opt}(t)^*(u(t) - u_{opt}(t))dt\right]^*,$$

we have that this term is zero as well. Using this in (2.27) we find that for any $u \in V_{x_0}$ we have that

$$\|u\|^2 = \|u - u_{opt}\|^2 + \|u_{opt}\|^2.$$

From this we directly conclude that the norm is minimal for $u = u_{opt}$ and that this is the only one. □

So from this theorem we conclude that the minimal energy needed to steer the state from zero to x_0 equals $\sqrt{x_0^* P^{-1} x_0}$, where P is the controllability gramian. For the energy coming out of x_0 via the output a similar result holds.

Theorem 2.4.2 *For $x_0 \in \mathbb{C}^n$ the energy of the corresponding output satisfies*

$$\|y\|^2 = \int_0^\infty \|Ce^{At} x_0\|^2 dt = x_0^* Q x_0, \tag{2.28}$$

where Q is the observability gramian, see (2.7).

Proof This follows directly from the definition of the observability gramian.

$$\begin{aligned}\|y\|^2 &= \int_0^\infty \|Ce^{At}x_0\|^2 dt = \int_0^\infty [Ce^{At}x_0]^* Ce^{At}x_0 dt \\ &= \int_0^\infty x_0^* e^{A^*t} C^* Ce^{At} x_0 dt = x_0^* Q x_0,\end{aligned}$$

where we have used (2.7). □

From the above two theorems we see that our aim to remove states, of norm one, that are hard to reach or hard to observe can now by formulated equivalently as to remove states (of norm one) for which $x_0^*Qx_0$ is small or for which $x_0^*P^{-1}x_0$ is large. This can still be contradictory, because states for which $x_0^*Qx_0$ is small, the energy $x_0^*P^{-1}x_0$ could be small as well. Thus how to decide?

It would be much easier if Q and P were the same and diagonal. To illustrate this, assume that $P = Q = \text{diag}_{i=1,\cdots,n}(\sigma_i)$. Let us consider $x_0 = e_1$, where e_1 is the first basis vector of $\mathbb{C}^n$. Then $x_0^*Qx_0 = \sigma_1$, and $x_0^*P^{-1}x_0 = e_1^* \left[\text{diag}_{i=1,\cdots,n}(\sigma_i)\right]^{-1} e_1 = \sigma_1^{-1}$. So checking whether σ_1 is small is the same as checking that σ_1^{-1} is large. This we can do for all basis vectors of $\mathbb{C}^n$, and remove those basis vectors for we find the σ_i's too small.

In Exercise 1 we saw that the solutions of the Lyapunov equation (2.8), P, and (2.9), Q, transform differently, and that is a reason why we can find a state transformation such that after this transformation they are the same and diagonal. This will be shown in the following theorem.

Theorem 2.4.3 *Consider the system (2.1), (2.2) for which we assume that A is Hurwitz, and that it is controllable and observable. We denote the solutions of the Lyapunov equations (2.8) and (2.9) by P and Q, respectively.*

Let R be a Cholesky factor *of Q, i.e.,*

$$Q = R^*R. \tag{2.29}$$

Next diagonalise the positive definite matrix, RPR^*, i.e.,

$$RPR^* = V\Sigma^2 V^*, \tag{2.30}$$

where V is unitary, i.e., $V^*V = I$ and Σ is diagonal, with decreasing diagonal elements, $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n), \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$. Then the following holds

1. *After the state transformation $T = \Sigma^{-\frac{1}{2}} V^* R$, the Lyapunov gramians are the same and equal to Σ.*

2. The σ_i^2, $i = 1, \cdots, n$ are the eigenvalues of PQ.

Proof *Item 1* After the state transformation $T = \Sigma^{-\frac{1}{2}} V^* R$, the solution P of (2.8) becomes

$$\begin{aligned} TPT^* &= \Sigma^{-\frac{1}{2}} V^* R P R^* V \Sigma^{-\frac{1}{2}} \\ &= \Sigma^{-\frac{1}{2}} V^* V \Sigma^2 V^* V \Sigma^{-\frac{1}{2}} = \Sigma, \end{aligned}$$

where we have used (2.30) and the fact that V is unitary.

After the state transformation $T = \Sigma^{-\frac{1}{2}} V^* R$, the solution Q of (2.9) becomes

$$\begin{aligned} T^{-*} Q T^{-1} &= \Sigma^{\frac{1}{2}} V^* R^{-*} Q R^{-1} V \Sigma^{\frac{1}{2}} \\ &= \Sigma^{\frac{1}{2}} V^* R^{-*} R^* R R^{-1} V \Sigma^{\frac{1}{2}} = \Sigma, \end{aligned}$$

where we have used (2.29) and the fact that V is unitary. Hence we have proved the assertion.

Item 2 From the previous part we can see that PQ undergoes a state transformation, and so the eigenvalues do not change. It is clear that after this state transformation the eigenvalues equal σ_i^2. Alternatively, we can reformulate PQ using (2.29) and (2.30), and obtain

$$PQ = PR^*R = R^{-1} V \Sigma^2 V^* R.$$

From this it is clear that the eigenvalues of PQ equal those of Σ^2, and so the proof is complete. □

This theorem leads to some notation.

Definition 2.4.1 *The σ_i's are called the* Hankel singular values. *The realisation which you find after the state space transformation is called the* balanced realisation.

2.5 Reduction using balancing

In Theorem 2.4.3, we defined the the following transformation matrix

$$T = \Sigma^{-\frac{1}{2}} V^* R, \tag{2.31}$$

where R, V and Σ are constructed via (2.29) and (2.30). With this matrix we transformed the system with matrices A, B, C into A_{bal}, B_{bal}, and C_{bal} given by

$$A_{bal} = TAT^{-1}, \quad B_{bal} = TB, \text{ and } C_{bal} = CT^{-1}. \tag{2.32}$$

This is known as the *balanced realisation* of our system (2.1)–(2.2). From Theorem 2.4.3 we know that the corresponding solutions to the Lyapunov equation, i.e.,

$$A_{bal}P_{bal} + P_{bal}A_{bal}^* = -B_{bal}B_{bal}^*, \text{ and } A_{bal}^*Q_{bal} + Q_{bal}A_{bal} = -C_{bal}^*C_{bal} \tag{2.33}$$

are given by $P_{bal} = Q_{bal} = \Sigma$. As discussed before we would like to remove states, x_0, for which $x_0^*Qx_0$ is small and/or for which $x_0^*P^{-1}x_0$. Now after the state transformation (2.31) this becomes the same condition. Furthermore, since $P_{bal} = Q_{bal}$ are diagonal, we want to remove those states for which σ_i are small. If we have ordered the Hankel singular values σ_i's, then the reduction is done as follows. Choose a threshold $\varepsilon > 0$ and remove the states corresponding to those σ_i below this value. So given $\varepsilon > 0$ and let n_0 be such that

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{n_0} \geq \varepsilon > \sigma_{n_0+1} \geq \ldots \geq \sigma_n. \tag{2.34}$$

Next we split A_{bal}, B_{bal} and C_{bal} correspondingly

$$A_{bal} = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right], \quad B_{bal} = \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right], \text{ and } C_{bal} = \left[\begin{array}{c|c} C_1 & C_2 \end{array}\right]. \tag{2.35}$$

So A_{11} is a $n_0 \times n_0$ matrix, B_1 a $n_0 \times m$ matrix, and C_1 is a $p \times n_0$ matrix.

Definition 2.5.1 *The* reduced system *is given by the system matrices A_{11}, B_1 and C_1 in (2.35). Thus*

$$\dot{z}(t) = A_{red}z(t) + B_{red}u(t), \qquad y(t) = C_{red}z(t) \tag{2.36}$$

with $A_{red} = A_{11}$, $B_{red} = B_1$, $C_{red} = C_1$, on the state space $\mathbb{C}^{n_0}$

Using the fact that $P_{bal} = Q_{bal} = \Sigma$ and the notation in (2.35), the Lyapunov equations (2.33) become

$$\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]\left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] + \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right]\left[\begin{array}{c|c} A_{11}^* & A_{21}^* \\ \hline A_{12}^* & A_{22}^* \end{array}\right] = -\left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right]\left[\begin{array}{c|c} B_1^* & B_2^* \end{array}\right], \tag{2.37}$$

$$\left[\begin{array}{c|c} A_{11}^* & A_{21}^* \\ \hline A_{12}^* & A_{22}^* \end{array}\right]\left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right] + \left[\begin{array}{c|c} \Sigma_1 & 0 \\ \hline 0 & \Sigma_2 \end{array}\right]\left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = -\left[\begin{array}{c} C_1^* \\ \hline C_2^* \end{array}\right]\left[\begin{array}{c|c} C_1 & C_2 \end{array}\right], \tag{2.38}$$

where

$$\Sigma_1 = \text{diag}(\sigma_1, \cdots, \sigma_{n_0}) \text{ and } \Sigma_2 = \text{diag}(\sigma_{n_0+1}, \cdots, \sigma_n).$$

Expanding these expressions, and considering only the upper left part of the resulting matrices, we find

$$A_{11}\Sigma_1 + \Sigma_1 A_{11}^* = -B_1 B_1^*, \text{ and } A_{11}^*\Sigma_1 + \Sigma_1 A_{11} = -C_1^* C_1. \tag{2.39}$$

These equations will be very useful in proving our main theorem. We begin by studying the stability of the system (2.36).

We started with a stable system, meaning that A is a Hurwitz matrix. Since A_{bal} is obtained after a basis transformation on A, it is Hurwitz as well. However, will the reduced system be stable, i.e., will all eigenvalues of A_{11} have negative real part? Note that the matrix

$$A = \begin{bmatrix} 1 & 2 \\ -3 & -2 \end{bmatrix}$$

is Hurwitz, but its upper 1×1 block is not. Hence the question could have a negative answer. The following theorem shows that all eigenvalues of $A_{red} = A_{11}$ have real part less than or equal to zero.

Theorem 2.5.1 *The reduced system (2.36) has the following properties;*

1. *The eigenvalues of A_{red} are all in the closed left half plane, i.e., they lie in $\overline{\mathbb{C}^-} = \{s \in \mathbb{C} \mid \mathrm{Re}(s) \leq 0\}$;*
2. *If v is an eigenvector corresponding to an eigenvalue of A_{11} on the imaginary axis, then $C_1 v = 0$;*
3. *If w is an eigenvector corresponding to an eigenvalue of A_{11}^* on the imaginary axis, then $B_1^* w = 0$;*

Proof The proof is strongly based on the Lyapunov equations after the state space transformation, i.e., equation (2.39).

Item 1. and 2. Let $\lambda \in \mathbb{C}$ be an eigenvalue of $A_{red} = A_{11}$ with corresponding eigenvector $v \in \mathbb{C}^{n_0}$, then the second equality in (2.39) gives

$$\begin{aligned} -v^* C_1^* C_1 v &= v^* A_{11}^* \Sigma_1 v + v^* \Sigma_1 A_{11} v \\ &= \overline{\lambda} v^* \Sigma_1 v + \lambda v^* \Sigma_1 v = 2\mathrm{Re}(\lambda) v^* \Sigma_1 v. \end{aligned}$$

Now $v^* C_1^* C_1 v = (C_1 v)^* C_1^* v = \|C_1 v\|^2$, and so

$$-\|C_1 v\|^2 = 2\mathrm{Re}(\lambda) v^* \Sigma_1 v. \tag{2.40}$$

Since $v^*\Sigma_1 v > 0$ (why?) we find that $\mathrm{Re}(\lambda) \leq 0$.

When λ lies on the imaginary axis, then using once more that $v^*\Sigma_1 v > 0$, we conclude from (2.40) that $C_1 v = 0$, and thus proving item 2.

Item 3. Similar as in the proof of item 1, we can show by using the first equation in (2.39) that when $w \in \mathbb{C}^{n_0}$ is an eigenvector corresponding to the eigenvalue μ of $A_{red}^* = A_{11}^*$, then

$$-\|B_1^* w\|^2 = 2\mathrm{Re}(\mu) w^* \Sigma_1 w. \tag{2.41}$$

Thus when μ lies on the imaginary axis, then $B_1^* w = 0$. □

We may wonder if A_{red} could have an eigenvalue on the imaginary axis. After all, the big matrix A is stable. The following example shows that this case can happen, even in the "smallest" case, $n = 2$ and $n_0 = 1$.

Example 2.5.1 *We choose*

$$A_{bal} = \left[\begin{array}{c|c} A_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right] = \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}.$$

This matrix has eigenvalues, $-\frac{1}{2} \pm \frac{1}{2}\sqrt{3}i$, and is thus stable. It is easy to see that for

$$\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \left[\begin{array}{c} B_1 \\ \hline B_2 \end{array}\right] = \begin{bmatrix} 0 \\ -\sqrt{2} \end{bmatrix}, \quad \left[\begin{array}{c|c} C_1 & C_2 \end{array}\right] = \begin{bmatrix} 0 & \sqrt{2} \end{bmatrix},$$

the Lyapunov equations (2.37) and (2.38) are satisfied. Since $A_{red} = A_{11} = 0$, this clearly has an eigenvalue on the imaginary axis.

So the above example shows that we can have eigenvalues of A_{red} on the imaginary axis. However, we see that the splitting of the Σ did not satisfy (2.34). Namely, in our example $\sigma_{n_0} = \sigma_1 = 1 = \sigma_2 = \sigma_{n_0+1}$. We will show in Theorem 2.5.2 that if $\sigma_{n_0} > \sigma_{n_0+1}$, then A_{red} is Hurwitz. For this we need the following lemma.

Lemma 2.5.1 *Let $\lambda = i\omega$ be an eigenvalue of A_{red} on the imaginary axis. Then*

$$\Sigma_1 \ker(A_{red} - i\omega I) \subseteq \ker(A_{red}^* + i\omega I), \text{ and}$$

$$\Sigma_1 \ker(A_{red}^* + i\omega I) \subseteq \ker(A_{red} - i\omega I).$$

Hence $\Sigma_1^2 \ker(A_{red} - i\omega I) \subseteq \ker(A_{red} - i\omega I)$, or equivalently the linear subspace $\ker(A_{red} - i\omega I)$ is Σ_1^2-invariant.

Proof Let $0 \neq v \in \ker(A_{red} - i\omega I)$, or equivalently let v be an eigenvector corresponding to the eigenvalue $i\omega$ of A_{red}. Then by Theorem 2.5.1, we know that $C_1 v = 0$. Multiplying the second equation in (2.39) by v, we find

$$A_{red}^* \Sigma_1 v + \Sigma_1 (i\omega) v = 0$$

Or equivalently, $\Sigma_1 v \in \ker(A_{red}^* + i\omega I)$.

Similarly, but now applying the first equation in (2.39), we find that $\Sigma_1 w \in \ker(A_{red} - i\omega I)$ for any $w \in \ker(A_{red}^* + i\omega I)$.

Combining these inclusions, we find that $\ker(A_{red} - i\omega I)$ is Σ_1^2-invariant. □

Theorem 2.5.2 *If Σ_1 and Σ_2 are chosen such that $\sigma_{n_0} > \sigma_{n_0+1}$ (see (2.34)), then A_{11} is a Hurwitz matrix. Furthermore, the system (2.36) is controllable and observable.*

Proof Assume that A_{red} has an eigenvalue on the imaginary axis. We denote this eigenvalue by $i\omega$. By Lemma 2.5.1 we have that the linear subspace $V := \ker(A_{red} - i\omega I) \subset \mathbb{C}^{n_0}$ is Σ_1^2-invariant. In particular, this implies that there is an eigenvector of Σ_1^2 in the kernel of $A_{red} - i\omega I$. We denote this eigenvector by v, and so $\Sigma_1^2 v = \sigma^2 v$ and $A_{red} v = i\omega v$. Since the eigenvalues of Σ_1^2 are known, we have that $\sigma^2 = \sigma_i^2$ for some $i \in \{1, \cdots, n_0\}$.

Next we define $x \in \mathbb{C}^n$ as $x = \left[\begin{smallmatrix} v \\ 0 \end{smallmatrix}\right]$. Using (2.38) and Theorem 2.5.1.2, we have

$$\begin{aligned}
0 = & -\begin{bmatrix} C_1^* \\ C_2^* \end{bmatrix} \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} \\
= & \begin{bmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} + \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} \\
= & \begin{bmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{bmatrix} \begin{bmatrix} \sigma v \\ 0 \end{bmatrix} + \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} A_{11} v \\ A_{21} v \end{bmatrix} \\
= & \begin{bmatrix} \sigma A_{11}^* v \\ \sigma A_{12}^* v \end{bmatrix} + \begin{bmatrix} \Sigma_1 A_{11} v \\ \Sigma_2 A_{21} v \end{bmatrix}.
\end{aligned}$$

In particular, we find

$$0 = \sigma A_{12}^* v + \Sigma_2 A_{21} v. \tag{2.42}$$

Similarly, using (2.37) and Theorem 2.5.1.3, we have

$$\begin{aligned} 0 = & -\begin{bmatrix} B_1 \\ B_2 \end{bmatrix}\begin{bmatrix} B_1^* & B_2^* \end{bmatrix}\begin{bmatrix} v \\ 0 \end{bmatrix} \\ = & \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}\begin{bmatrix} v \\ 0 \end{bmatrix} + \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}\begin{bmatrix} A_{11}^* & A_{21}^* \\ A_{12}^* & A_{22}^* \end{bmatrix}\begin{bmatrix} v \\ 0 \end{bmatrix} \\ = & \begin{bmatrix} \sigma A_{11} v \\ \sigma A_{21} v \end{bmatrix} + \begin{bmatrix} \Sigma_1 A_{11}^* v \\ \Sigma_2 A_{12}^* v \end{bmatrix}. \end{aligned}$$

From which we obtain that

$$0 = \sigma A_{21} v + \Sigma_2 A_{12}^* v. \tag{2.43}$$

Combining (2.42) and (2.43) we find

$$\sigma^2 A_{12}^* v = -\sigma \Sigma_2 A_{21} v = -\Sigma_2 (\sigma A_{21} v) = \Sigma_2^2 A_{12}^* v$$

In other words, if $A_{12}^* v$ would be a non-zero vector, then σ^2 is an eigenvalue of Σ_2^2. However, we know that $\sigma = \sigma_i$ for some $i \in \{1, \cdots, n_0\}$ and these sigma's are all larger than the eigenvalues of Σ_2. So $A_{12}^* v$ is the zero vector. Equation (2.43) gives that

$$A_{21} v = 0.$$

Applying this, and the fact that v is an eigenvector of $A_{red} = A_{11}$ we find that

$$A_{bal}\begin{bmatrix} v \\ 0 \end{bmatrix} = \begin{bmatrix} A_{11} v \\ A_{21} v \end{bmatrix} = \begin{bmatrix} i\omega v \\ 0 \end{bmatrix} = i\omega \begin{bmatrix} v \\ 0 \end{bmatrix}.$$

However, A and thus A_{bal} is Hurwitz, and so v must be zero, which provides the contradiction. □

When reducing a system/equation, we would like to know how large/small the error is. The following theorem gives the estimate for the balanced truncation.

Theorem 2.5.3 *Given a stable, controllable and observable system (2.1)–(2.2). Let the reduced system (2.36) be constructed via the balanced realisation, see Definition 2.5.1 and (2.35). Furthermore, let the Hankel singular values be ordered as in (2.34). Then the following estimate holds*

$$\sup_{s \in \mathbb{C}^+} \|C(sI - A)^{-1} B - C_{red}(sI - A_{red})^{-1} B_{red}\| \leq 2 \sum_{k=n_0+1}^{n} \sigma_k. \tag{2.44}$$

where $\mathbb{C}_0^+ = \{s \in \mathbb{C} \mid \mathrm{Re}(s) \geq 0\}$.

This theorem is not so simple to prove, and one of the biggest system theoretic results in the eighties of the last century. In the next section we will give the proof. However, before doing so, we make some remarks.

Remark 2.5.1 *Regarding the estimate (2.44) two remarks can be made.*

1. *If two or more singular values are the same, then it only has to be counted once in the sum. So the the right hand side of (2.44) can be made sharper by replacing it by*

$$2 \sum_{k=n_0+1, \sigma_k \neq \sigma_\ell}^{n} \sigma_k.$$

   Since it is very rare that two singular values are the same, we almost always see the right hand side in (2.44). However, the proof gives directly the sharper estimate.

2. *A result in complex function theory gives that when a scalar transfer function $h(s)$ satisfies that*

$$\sup_{s \in \mathbb{C}^+} |h(s)| < \infty,$$

   then

$$\sup_{\omega \in \mathbb{R}} |h(i\omega)| = \sup_{s \in \mathbb{C}^+} |h(s)|.$$

   So the maximum of h over the right half plane equals that over the imaginary axis. This result extends to matrix valued functions, and so to see how close the reduced model approximates the original system we only have to calculate

$$\sup_{\omega \in \mathbb{R}} \|C(i\omega I - A)^{-1}B - C_{red}(i\omega I - A_{red})^{-1}B_{red}\|.$$

2.6 Proof of Theorem 2.5.3

The proof is separated into two parts. The first part is the longest and assumes that Σ_2 is a multiple of the identity. The second part explains how this special case helps to solve the general case.

Part 1. We choose n_0 the largest index for which $\sigma_{n_0} > \sigma_n$, and take $\Sigma_2 = \sigma_n I$, i.e., $\sigma_{n_0+1} = \sigma_{n_0+2} = \cdots = \sigma_n$. Note that in general this n_0 will be equal to $n-1$, and then Σ_2 will just be the 1×1 matrix with entry σ_n.

We write the state space equation of the balanced system as, see (2.35),

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u(t), \text{ and } y(t) = \begin{bmatrix} C_1 & C_2 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}.$$

and that of the reduced system as, see (2.36),

$$\dot{x}_r(t) = A_{11}x_r(t) + B_1u(t), y_r(t) = C_1x_r(t)$$

The "error" system has a state space representation, see Exercise 1 of Section 2.2.

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_r(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ A_{21} & A_{22} & 0 \\ 0 & 0 & A_{11} \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_r(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \\ B_1 \end{bmatrix} u(t), \text{ and}$$

$$e(t) = \begin{bmatrix} C_1 & C_2 & -C_1 \end{bmatrix}\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_r(t) \end{bmatrix},$$

where $e(t) = y(t) - y_r(t)$. Introducing the new state coordinates $z(t) = x_1(t) - x_r(t)$ and $w(t) = x_1(t) + x_r(t)$, gives the state space representation

$$\frac{d}{dt}\begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & 0 \\ \frac{1}{2}A_{21} & A_{22} & \frac{1}{2}A_{21} \\ 0 & A_{12} & A_{11} \end{bmatrix}\begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 2B_1 \end{bmatrix} u(t), \text{ and}$$

$$e(t) = \begin{bmatrix} C_1 & C_2 & 0 \end{bmatrix}\begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix}.$$

Consider next the following (Lyapunov) function

$$\begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix}^* \mathcal{P} \begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix} := z(t)^*\Sigma_1 z(t) + 2x_2(t)^*\sigma_n x_2(t) + \sigma_n^2 w(t)^*\Sigma_1^{-1}w(t).$$

Thus $\mathcal{P}$ is block diagonal, i.e., $\mathcal{P} = \text{diag}(\Sigma_1, 2\sigma_n I, \sigma_n^2\Sigma_1^{-1})$. It is clear that it is a positive,

symmetric matrix. Furthermore, along solutions, we have (we omit the t)

$$
\begin{aligned}
&\frac{d}{dt}\left(\begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \mathcal{P} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}\right) = \\
&\left(\begin{bmatrix} A_{11} & A_{12} & 0 \\ \frac{1}{2}A_{21} & A_{22} & \frac{1}{2}A_{21} \\ 0 & A_{12} & A_{11} \end{bmatrix} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 2B_1 \end{bmatrix} u\right)^* \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & 2\sigma_n I & 0 \\ 0 & 0 & \sigma_n^2 \Sigma_1^{-1} \end{bmatrix} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \\
&\begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & 2\sigma_n I & 0 \\ 0 & 0 & \sigma_n^2 \Sigma_1^{-1} \end{bmatrix} \left(\begin{bmatrix} A_{11} & A_{12} & 0 \\ \frac{1}{2}A_{21} & A_{22} & \frac{1}{2}A_{21} \\ 0 & A_{12} & A_{11} \end{bmatrix} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ B_2 \\ 2B_1 \end{bmatrix} u\right) \\
&= \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \left(\begin{bmatrix} \Sigma_1 A_{11} & \Sigma_1 A_{12} & 0 \\ \sigma_n A_{21} & 2\sigma_n A_{22} & \sigma_n A_{21} \\ 0 & \sigma_n^2 \Sigma_1^{-1} A_{12} & \sigma_n^2 \Sigma_1^{-1} A_{11} \end{bmatrix} + \right. \\
&\qquad \left. \begin{bmatrix} A_{11}^* \Sigma_1 & \sigma_n A_{21}^* & 0 \\ A_{12}^* \Sigma_1 & 2\sigma_n A_{22}^* & \sigma_n^2 A_{12}^* \Sigma_1^{-1} \\ 0 & \sigma_n A_{21}^* & \sigma_n^2 A_{11}^* \Sigma_1^{-1} \end{bmatrix}\right) \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \\
&\begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \begin{bmatrix} 0 \\ 2\sigma_n B_2 \\ 2\sigma_n^2 \Sigma_1^{-1} B_1 \end{bmatrix} u + u^* \begin{bmatrix} 0 \\ 2\sigma_n B_2 \\ 2\sigma_n^2 \Sigma_1^{-1} B_1 \end{bmatrix}^* \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} \\
&= \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \left(\begin{bmatrix} \Sigma_1 A_{11} & \Sigma_1 A_{12} & 0 \\ \sigma_n A_{21} & \sigma_n A_{22} & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} A_{11}^* \Sigma_1 & \sigma_n A_{21}^* & 0 \\ A_{12}^* \Sigma_1 & \sigma_n A_{22}^* & 0 \\ 0 & 0 & 0 \end{bmatrix} \right. \\
&\qquad \left. \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_n A_{22} & \sigma_n A_{21} \\ 0 & \sigma_n^2 \Sigma_1^{-1} A_{12} & \sigma_n^2 \Sigma_1^{-1} A_{11} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_n A_{22}^* & \sigma_n^2 A_{12}^* \Sigma_1^{-1} \\ 0 & \sigma_n A_{21}^* & \sigma_n^2 A_{11}^* \Sigma_1^{-1} \end{bmatrix}\right) \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \\
&2\sigma_n \left[x_2^* B_2 u + u^* B_2^* x_2\right] + 2\sigma_n^2 \left[w^* \Sigma_1^{-1} B_1 u + u^* B_1^* \Sigma_1^{-1} w\right].
\end{aligned}
$$

Using (2.38) and the fact that $\Sigma_2 = \sigma_n I$, we write the above as

$$
\begin{aligned}
&\frac{d}{dt}\left(\begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \mathcal{P} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}\right) = \\
&= \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \begin{bmatrix} -C_1^*C_1 & -C_1^*C_2 & 0 \\ -C_2^*C_1 & -C_2^*C_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \\
&\quad \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \begin{bmatrix} 0 & 0 & 0 \\ 0 & \sigma_n A_{22} + \sigma_n A_{22}^* & \sigma_n A_{21} + \sigma_n^2 A_{12}^* \Sigma_1^{-1} \\ 0 & \sigma_n A_{21}^* + \sigma_n^2 \Sigma_1^{-1} A_{12} & \sigma_n^2(\Sigma_1^{-1} A_{11} + A_{11}^* \Sigma_1^{-1}) \end{bmatrix} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix} + \\
&\quad 2\sigma_n \left[x_2^* B_2 u + u^* B_2^* x_2\right] + 2\sigma_n^2 \left[w^* \Sigma_1^{-1} B_1 u + u^* B_1^* \Sigma_1^{-1} w\right] \\
&= -\|e(t)\|^2 + \\
&\quad \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \end{bmatrix}^* \begin{bmatrix} \sigma_n A_{22} + \sigma_n A_{22}^* & A_{21}\Sigma_1 + \sigma_n A_{12}^* \\ \Sigma_1 A_{21}^* + \sigma_n A_{12} & A_{11}\Sigma_1 + \Sigma_1 A_{11}^* \end{bmatrix} \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \end{bmatrix} + \\
&\quad 2\sigma_n \left[x_2^* B_2 u + u^* B_2^* x_2\right] + 2\sigma_n^2 \left[w^* \Sigma_1^{-1} B_1 u + u^* B_1^* \Sigma_1^{-1} w\right],
\end{aligned}
$$

where we have used the definition of e.

Using (2.37) and the fact that $\Sigma_2 = \sigma_n I$, we write the above as

$$
\begin{aligned}
&\frac{d}{dt}\left(\begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}^* \mathcal{P} \begin{bmatrix} z \\ x_2 \\ w \end{bmatrix}\right) \\
&= -\|e(t)\|^2 + \\
&\quad \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \end{bmatrix}^* \begin{bmatrix} -B_2 B_2^* & -B_2 B_1^* \\ -B_1 B_2^* & -B_1 B_1^* \end{bmatrix} \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \end{bmatrix} + \\
&\quad 2\sigma_n \left[x_2^* B_2 u + u^* B_2^* x_2\right] + 2\sigma_n^2 \left[w^* \Sigma_1^{-1} B_1 u + u^* B_1^* \Sigma_1^{-1} w\right] \\
&= -\|e(t)\|^2 + \\
&\quad \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \\ u \end{bmatrix}^* \begin{bmatrix} -B_2 B_2^* & -B_2 B_1^* & 2\sigma_n B_2 \\ -B_1 B_2^* & -B_1 B_1^* & 2\sigma_n B_1 \\ 2\sigma_n B_2^* & 2\sigma_n B_1^* & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ \sigma_n \Sigma_1^{-1} w \\ u \end{bmatrix}.
\end{aligned}
$$

Since

$$\begin{bmatrix} B_2 \\ B_1 \\ -2\sigma_n I \end{bmatrix} \begin{bmatrix} B_2 \\ B_1 \\ -2\sigma_n I \end{bmatrix}^* = \begin{bmatrix} B_2B_2^* & B_2B_1^* & -2\sigma_n B_2 \\ B_1B_2^* & B_1B_1^* & -2\sigma_n B_1 \\ -2\sigma_n B_2^* & -2\sigma_n B_1^* & 4\sigma_n^2 I \end{bmatrix},$$

the above equality becomes

$$\begin{aligned} &\frac{d}{dt}\left(\begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix}^* \mathcal{P} \begin{bmatrix} z(t) \\ x_2(t) \\ w(t) \end{bmatrix} \right) \\ &\quad = -\|e(t)\|^2 - \begin{bmatrix} x_2 \\ \sigma_n\Sigma_1^{-1}w \\ u \end{bmatrix}^* \begin{bmatrix} B_2 \\ B_1 \\ -2\sigma_n I \end{bmatrix} \begin{bmatrix} B_2 \\ B_1 \\ -2\sigma_n I \end{bmatrix}^* \begin{bmatrix} x_2 \\ \sigma_n\Sigma_1^{-1}w \\ u \end{bmatrix} + 4\sigma_n^2\|u\|^2 \\ &\quad = -\|e(t)\|^2 - \|B_2^*x_2(t) + \sigma_n B_1^*\Sigma_1^{-1}w(t) - 2\sigma_n u(t)\|^2 + 4\sigma_n^2\|u(t)\|^2 \\ &\quad \leq -\|e(t)\|^2 + 4\sigma_n^2\|u(t)\|^2. \end{aligned}$$

By Theorem 2.2.2 it follows that the transfer function $H(s)$ from u to e satisfies the estimate

$$\|H(s)u_0\|^2 \leq 4\sigma_n^2\|u_0\|^2 \text{ for } s \in \mathbb{C} \text{ s.t. } \mathrm{Re}(s) \geq 0. \tag{2.45}$$

The transfer function $H(s)$ equals the transfer function of the full order plant minus that of the reduced order plant. Thus with Theorem 2.2.1

$$H(s) = C(sI - A)^{-1}B - C_{red}(sI - A_{red})^{-1}B_{red}.$$

Combining this with (2.45) we taking the supremum over all $u_0 \in U$, we find (2.44).

Part 2. The "first" reduced system found in the previous part is stable, controllable and observable, according to Theorem 2.5.2. Furthermore it is balanced with controllability and observability gramian

$$\tilde{\Sigma} = \mathrm{diag}\left(\sigma_1, \cdots, \sigma_{n-p_n}\right),$$

where p_n is the multiplicity of the Hankel singular value σ_n. For this system we can repeat the process as done in the first part, and approximate it with an error of at most $2\sigma_{n-n_p}$. Note that the new A_{red} is just a part of the original A_{red}. Similar for B_{red} and C_{red}. So we get the same as if we would have removed the states corresponding to σ_n and σ_{n-p_0} in one reduction step. Furthermore, by the triangular inequality the error between the original and second reduced system is less than or equal to the sum of the errors of the first and second reduction step, $2\sigma_n + 2\sigma_{n-p_n}$.

This process we can repeat, and gives us the estimate (2.44). □

Chapter 3

Discretising coarsened PDE models

3.1 Introduction

The formulation of a model for a physical/engineering problem may often involve partial differential equations (PDEs) to describe how the properties of the solution evolve in time and space. Important examples include fluid mechanics, process engineering and electromagnetism. These fields of application are characterized by well-established concepts and basic models. Despite the 'mature' status of the conceptual modeling in these areas, many challenges remain that arise from the fact that the actual solution to these problems is characterized by a huge range of spatial and temporal features. As a result, the computation of the actual solution to such multiscale problems is very costly and often not even feasible on current (super-)computers. This sets the stage for model reduction of problems governed by PDEs, to reduce the final complexity of the computational problem. In this chapter we focus on complexity reduction by spatial filtering and consider the translation of the resulting formulation into a discrete model suitable for numerical simulation., as is developed in great detail in the context of turbulent flows.

Complexity reduction of a computational PDE model may be obtained by applying a spatial or a temporal filter to the original governing equations. So-called low-pass filters can be considered for this purpose, of which the top-hat filter is a particularly simple example. In fact, top-hat filtering corresponds to local volume averaging. The filter is denoted by

$L : u \to \overline{u}$ and in one spatial dimension it is explicitly defined by

$$\overline{u}(x,t) = L(u)(x,t) = \int_{x-\Delta/2}^{x+\Delta/2} \frac{u(\xi,t)}{\Delta}\, d\xi \tag{3.1}$$

in which the parameter Δ denotes the filter-width. Application of such filter to a solution u will affect features with length scales comparable to or smaller than Δ, while leaving structures that are considerably larger than Δ virtually unaffected. This is illustrated in figure 3.1 in which a signal, composed of a number of harmonic functions, is explicitly filtered at different Δ. One may clearly see the effective smoothing of u, illustrating intuitively the external control one achieves over the complexity of the signal by filtering.

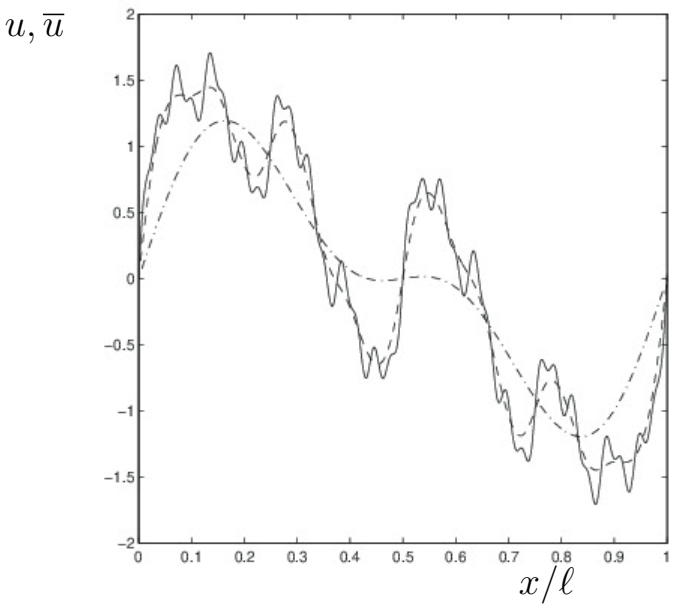


Figure 3.1: Filtering a reference signal (solid) with a top-hat filter with filter-width equal to $\ell/16$ (dashed) and $\ell/4$ (dash-dotted).

The top-hat filter can readily be extended to three spatial dimensions by consecutively applying the one-dimensional filter (3.1) in each of the three coordinate directions. An illustration of filtering a developed numerical turbulence solution is collected in figure 3.2. In this example, we used a snapshot of a turbulent mixing layer to highlight the complexity reduction achieved by increasing the filter width. The trends observed in relation to figure 3.1 clearly reappear in this three-dimensional case. With increasing filter width localized details in the solution, on the scale Δ, are strongly reduced and one can expect to

represent the filtered solution on a much coarser spatial grid while maintaining acceptable accuracy.

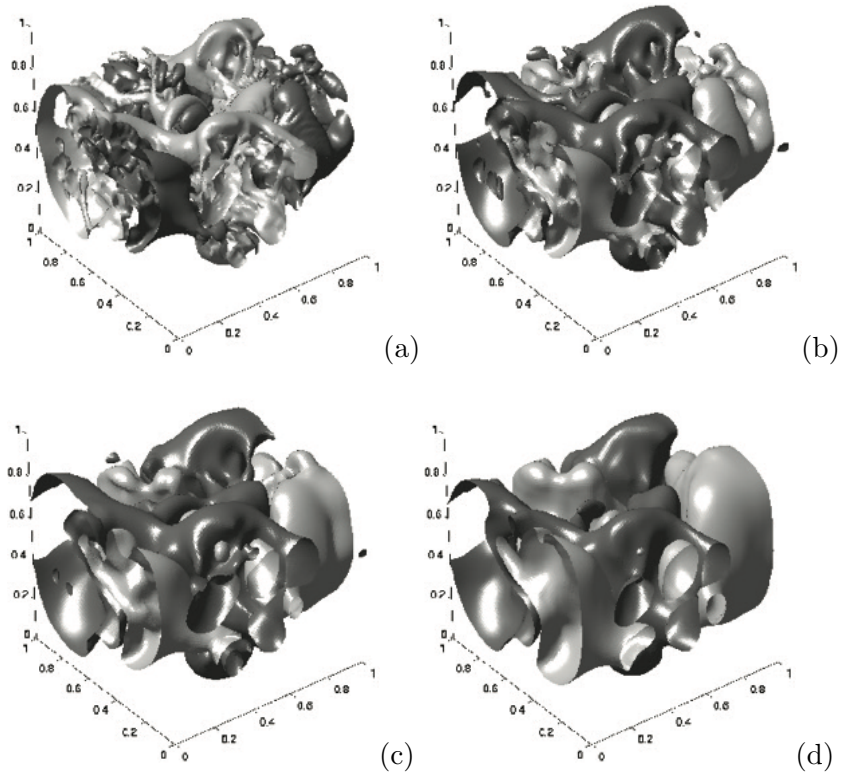


Figure 3.2: Snapshot of the vertical velocity field attained in the developed turbulent stages of temporal mixing layer flow. In (a) a fully resolved solution in a box of size ℓ^3 is shown and the corresponding filtered solution is depicted in the other graphs, using $\Delta = \ell/32$ (b), $\Delta = \ell/16$ (c) and $\Delta = \ell/8$ (d). The light (dark) level-sets correspond to upward (downward) flow.

The simple application of the top-hat filter shows that the remaining filtered signal $\overline{u}$ possesses a much reduced representation of the smaller-scale features. This hints at a 'balancing' between, on the one hand, retaining sufficient contributions of the smaller scales to approximate the original solution in adequate detail, while, on the other hand, reducing the even more localized features to make the computation of the filtered solution $\overline{u}$ much less costly. Intuitively, there is a definite limit to the amount of detail that can

be filtered away. A strong reduction of smaller scales can be achieved by selecting a wide filter width, which greatly reduces the spatial resolution needed for an accurate numerical representation. Exaggerating this reduction would, however, yield an impression of the original signal that is so incomplete that it becomes quite useless for actual predictions. Conversely, much of the details of the original signal can be maintained in case Δ is chosen sufficiently small, compared to the physical length-scale spectrum of the actual solution. This would, however, not yield any significant saving of computational resources to represent such mildly filtered solution. The aim is to find a proper balance between these two tendencies. Since a quantitative theory for this problem is lacking, the search for this balance requires problem-specific input and careful *a posteriori* judgment.

A systematic approach to the reduction of solution complexity can be formulated in a general spatial filtering approach. The organization of this chapter is as follows. For completeness, we briefly introduce partial differential equations in Section 3.2. Section 3.3 introduces convolution filters and quantifies the application of such filters to complex solutions. The closure problem that emerges when filtering nonlinear problems is studied in Section 3.3. After coarsening by filtering we study computational models that arise from discretization of the filtered continuum formulations. While the coarsening of the equations provides a clear basis for the corresponding reduced order model, errors arising from shortcomings in the adopted model used for the reduction, and errors due to the spatial and temporal discretization that was adopted, imply uncertainty in the final predictions. We discuss numerical discretization in Section 3.4, and concentrate in more detail on the accuracy of discrete derivatives in Section 3.5.

3.2 Partial differential equations

In this section we briefly introduce Partial Differential Equations (PDEs) and discuss a few well-known physical problems that are modeled in terms of these equations. The presentation will be introductory only - excellent references exist, e.g., [4].

PDEs are equations that formulate a desired solution u in terms of a relationship containing independent variables x_j, $j = 1, ..., n$ and t, as well as partial derivatives of u with respect to these variables. Often, the x_j are referred to as spatial variables and t is interpreted as a time variable. PDEs are defined on a domain Ω, delineating sets for x_j and t for which the PDE is enforced. The domain $\Omega = \Omega_x \times \Omega_t$ has a boundary $\partial\Omega = \partial\Omega_x \times \partial\Omega_t$ in case spatial (Ω_x) and temporal ($\Omega_t = [t_0, t_F]$) subdomains are identified. Here, t_0

and t_F denote the initial and final time of interest, respectively. At these boundaries conditions may be applied to specify the solution u, often motivated by additional modeling assumptions. The boundary condition corresponding to the time variable at $t = t_0$ is referred to as the initial condition, while the final time t_F may for some PDEs also be endowed with a specific condition. PDEs may be linear or nonlinear in u, and have a property referred to as its order, referring to the highest partial derivative contained in the equation.

One finds partial differential equations in practically every branch of physics, chemistry, and engineering. They are also found in other branches of the physical sciences and in the social sciences, economics, business, etc. Many parts of theoretical physics are formulated in terms of partial differential equations. In some cases, axioms uinderpinning a theory require that the states of physical systems be given by solutions of partial differential equations. In other cases, partial differential equations arise when one applies the axioms to specific situations.

First-order equations are of considerable importance to computational modeling in Science and Engineering. In general, a first-order equation may be expressed as

$$F(x_1, x_2, ..., x_n, t, u, \partial_1 u, \partial_2 u, ..., \partial_n u, \partial_t u) = 0 \tag{3.2}$$

where $\partial_j u = \partial u / \partial x_j$. The advection equation

$$u_t + a u_x = 0 \quad ; \quad x \in \mathbb{R}, \quad t > 0 \tag{3.3}$$

with constant $a > 0$ is one of the best known cases of a first-order equation. With initial condition $u(x, 0) = f(x)$, this problem is directly solvable as $u(x, t) = f(x - at)$, hence serving as a versatile reference problem with which the accuracy of certain numerical methods can be investigated.

An example of a partial differential equation of second order is

$$\frac{\partial^2 u}{\partial t^2} - c \frac{\partial^2 u}{\partial x^2} = \partial_{tt} u - c \partial_{xx} u = 0 \tag{3.4}$$

where we introduced the notations ∂_{tt} and $\partial_{xx} u$. This equation is known as the wave equation, describing the propagation of a wave at wave speed c. It is linear, second order in time t and space x. In general, linear, second-order PDEs can be classified as hyperbolic (e.g., flow problems), parabolic (e.g., heat transport) or elliptic (e.g., Laplace equation in electrostatics), referring to particular transport mechanisms underpinning the equations, and, correspondingly, particular mathematical structure of the solution.

Completion of this example problem requires the specification of the domain, e.g., $\Omega = [0, L] \times [0, t_F]$ for a domain of length L. Boundaries in x require boundary conditions which may be specified, e.g., as so-called Dirichlet conditions $u(0, t) = g(t)$ and $u(L, t) = h(t)$ in terms of arbitrary functions g and h. In view of the wave equation being of second order in time, two initial conditions require specification, e.g., $u(x, 0) = \Phi(x)$ and $\partial_t u(x, 0) = \Psi(x)$. This would correspond to the well-known example of a string of length L, clamped at $x = 0$ and $x = L$ into moving boundary locations, and with a given initial shape and velocity. Various other options can be chosen, of course, depending on the particular problem studied. Specification of the derivative at the spatial boundaries would be referred to as Neumann conditions, while also combinations of u and $\partial_x u$ at the boundaries could be enforced, known as Robin conditions.

Partial differential equations can prove to be difficult to solve. For selected problems the use of certain techniques such as the separation method, or a change of variables might be successful. More often, an analytical solution is not achievable and numerical simulation approaches need to be developed to approximate the solution to within a controllable tolerance. Even when following the simulation approach, success is not guaranteed since the problem may be computationally too expensive to treat in all details. This is where model reduction appears as a viable and practical way forward.

3.3 Coarsening by filtering

In this Section we investigate the application of a convolution filter to a PDE and introduce the central closure problem that emerges when nonlinear equations are being filtered.

3.3.1 Basic filters

In order to arrive at the desired complexity reduction of a PDE solution with a wide range of length-scales, one can apply a spatial filtering operation. This operation should be designed such that the large-scale features remain essentially unchanged by the filtering while flow features of sufficiently small scales should be strongly attenuated.

Convolution filters

We first consider basic convolution filters in one spatial dimension and extend to the three-dimensional case afterwards. A signal u may be filtered with a filter-operation L and give

rise to the filtered solution $\overline{u}$ which is defined as:

$$\overline{u}(x) = L(u) = \int_{-\infty}^{\infty} G(x-\xi)u(\xi)\ d\xi = G * u \tag{3.5}$$

in which we introduced the filter-kernel G associated with the filter L. The filter L is a linear operation and we require all admissible filters to be normalized , i.e.,

$$L(1) = \int_{-\infty}^{\infty} G(z)\ dz = 1 \tag{3.6}$$

which assures that if $u = const$ then it is invariant under filtering. This corresponds directly with the intuitive requirement that filtering of large-scale flow features should have only a small effect.

Next to the specification of a filter in physical space by its kernel G the characterization of the effect of a filter in spectral space by the Fourier-transform $H(k)$ of $G(z)$ is illustrative. By definition

$$H(k) = \int_{-\infty}^{\infty} G(z)e^{-ikz}\ dz \tag{3.7}$$

The normalization condition expresses itself as $H(0) = 1$. If we consider the signal u to be periodic with period 1, i.e.,

$$u(x) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi ikx} \tag{3.8}$$

then the effect of filtering can directly be written with the help of $H(k)$:

$$\overline{u}(x) = \sum_{k=-\infty}^{\infty} \Big(H(k)c_k\Big)e^{2\pi ikx} \tag{3.9}$$

Thus, in order for small-scale features (i.e., large k) to be strongly attenuated it is required that $|H(k)| \ll 1$ if k becomes large. Consequently, the filters L are so-called low-pass filters .

Filter-width

The filter kernels G which are considered here are functions which are 'peaked' around the origin. A certain effective width Δ can be associated with the filter L, which also defines the notion of 'small' versus 'large' scales in a solution. All length-scales will be interpreted in terms of the width of the adopted filter. We can define Δ in a number of different ways:

1. If the kernel G is square integrable then a characteristic size is given by

$$\frac{1}{\Delta} = \frac{1}{2\pi}\int_{-\infty}^{\infty} |H(k)|^2 \, dk \tag{3.10}$$

   which also corresponds to the integral of G^2 via Parseval's equality.

2. An alternative definition for the filter-width expresses the width Δ in terms of the second moment of G:

$$\Delta^2 = \int_{-\infty}^{\infty} z^2 G(z) \, dz = -H''(0) \tag{3.11}$$

3. Various spatially localized filters are characterized by a compact support, i.e., the filter-kernel G is non-zero only in a bounded interval around the origin. The width of this interval may be used to define the width.

We will consider these three filter width definitions for three commonly adopted filters; the 'top-hat' or 'box' filter, the Gaussian filter and the spectral cut-off filter.

Popular spatial filters

The top-hat filter is computationally the simplest filter. This compact support filter is defined by the filter kernel:

$$G_{th}(z) = \frac{1}{\ell}\{\mathcal{H}(z+\frac{\ell}{2}) - \mathcal{H}(z-\frac{\ell}{2})\} = \begin{cases} 1/\ell & \text{if } |z| < \ell/2 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

in which we introduced a length-parameter ℓ and $\mathcal{H}$ denotes the Heaviside function which is equal to 1 if $z \geq 0$ and 0 elsewhere. The Fourier-transform H_{th} of G_{th} is given by

$$H_{th}(k\ell) = \frac{\sin(k\ell/2)}{(k\ell/2)} \tag{3.13}$$

The width of this filter, according to the three definitions given above is: $\Delta = \ell$ according to (3.10), $\Delta = \ell/(2\sqrt{3})$ according to (3.11) and $\Delta = \ell$ using the size of the support of the filter to define the width.

The Gaussian filter may be defined by

$$G_G(z) = \Big(\frac{\alpha}{\pi\ell^2}\Big)^{1/2} \exp\Big(-(\frac{\alpha z^2}{\ell^2})\Big) \tag{3.14}$$

which has Fourier-transform given by

$$H_G(k\ell) = \exp\Big(-\frac{(k\ell)^2}{4\alpha}\Big) \tag{3.15}$$

The width of this filter, according to the three definitions given above is: $\Delta = \ell\sqrt{2\pi/\alpha}$ according to (3.10), $\Delta = \ell/\sqrt{2\alpha}$ according to (3.11). This filter has infinite support and correspondingly the third definition of filter width can not be applied. For the special choice $\alpha = 2\pi$ we observe that $\Delta = \ell$ according to (3.10), i.e., identical to the result for the top-hat filter. In case $\alpha = 6$, which is often used in literature [5, 6] we observe that $\Delta_G = \Delta_{th}$ according to (3.11).

filter	filter kernel $G(\mathbf{x} - \boldsymbol{\xi})$	Fourier transform $H(\mathbf{k})$
top-hat	$\begin{cases} 1/\Delta^3 & \text{if } \lvert x_i - \xi_i\rvert < \Delta_i/2 \\ 0 & \text{otherwise} \end{cases}$	$\prod_{i=1}^3 \frac{\sin(k_i\Delta_i/2)}{k_i\Delta_i/2}$
Gaussian	$\prod_{i=1}^3 (\frac{\alpha}{\pi\Delta_i^2})^{\frac{1}{2}} e^{-\alpha\frac{(x_i-\xi_i)^2}{\Delta_i^2}}$	$\prod_{i=1}^3 e^{-(k_i\Delta_i)^2/(4\alpha)}$
spectral cut-off	$\prod_{i=1}^3 \frac{\sin(\pi(x_i-\xi_i)/\Delta_i)}{\pi(x_i-\xi_i)}$	$\begin{cases} 1 & \text{if } \lvert k_i\Delta_i\rvert < \pi \\ 0 & \text{otherwise} \end{cases}$

Table 3.1: Filter functions in physical and spectral space of some product filters in three spatial dimensions.

Finally, as a third example of a commonly adopted filter, we consider the spectral cut-off filter, which is given by

$$G_{co}(z) = \frac{1}{\ell}\frac{\sin(\pi z/\ell)}{(\pi z/\ell)} \tag{3.16}$$

with Fourier-transform given by

$$H_{co}(k\ell) = \begin{cases} 1 & \text{if } |k\ell| < \pi \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

The width of this filter is $\Delta = \ell$ according to (3.10). The filter width of this filter turns out to be infinite in case one of the other two definitions is adopted. In the sequel, we will

mainly concentrate on compact support filters and identify the parameter ℓ in the filters directly with its width Δ.

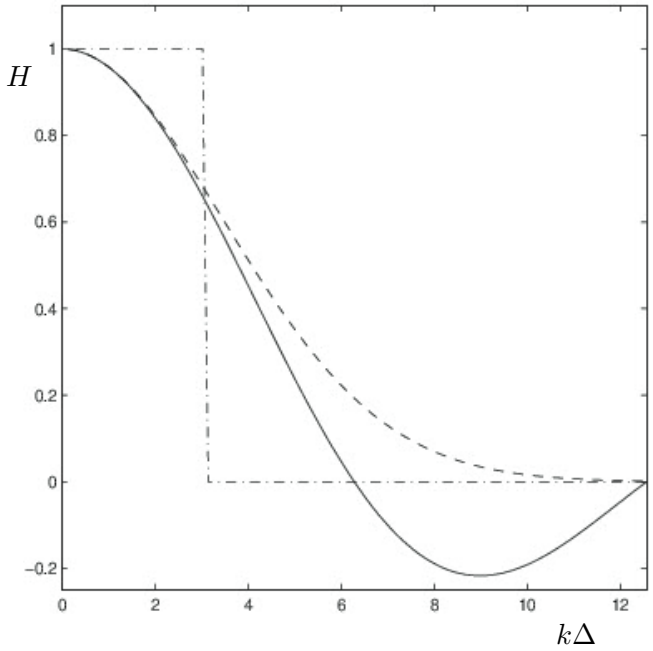


Figure 3.3: Fourier transforms of the filter kernels of the top-hat (solid), Gaussian (dashed) and spectral cut-off (dash-dotted) filters, showing H as a function of $k\Delta$.

3.3.2 Filtering linear and nonlinear PDEs

The application of a convolution filter to a linear or a nonlinear PDE will be discussed next. We notice that convolution filters commutate with partial derivatives with respect to time and space, i.e.,

$$L(\partial_t u) = \partial_t(L(u)) \quad ; \quad L(\partial_x u) = \partial_x(L(u)) \tag{3.18}$$

Moreover, the filters considered here are linear, implying

$$L(\alpha u + \beta v) = \alpha L(u) + \beta L(v) \tag{3.19}$$

for arbitrary constants a and b and arbitrary functions u and v. With these properties established, we may readily investigate the filtering of a linear PDE. As an illustration we

consider an advection-diffusion-reaction equation in which

$$\partial_t u + a\partial_x u - b\partial_{xx} u + cu = 0 \tag{3.20}$$

where a, b, c are positive constants, reflecting the strength of the advection (a), the diffusion (b) and the reaction (c). Application of a convolution filter, using the linearity of the filter, implies:

$$L(\partial_t u) + aL(\partial_x u) - bL(\partial_{xx} u) + cL(u) = 0 \tag{3.21}$$

Moreover, since L commutates with partial derivatives ∂_t and ∂_x, we infer

$$\partial_t \overline{u} + a\partial_x \overline{u} - b\partial_{xx} \overline{u} + c\overline{u} = 0 \tag{3.22}$$

in which we introduced the notation $\overline{u} = L(u)$.

This brief derivation shows that application of the convolution filter L leads to the same type of equation for $\overline{u}$. In fact, if L would be invertible, the definition $\overline{u} = L(u)$ can be readily identified as a change of variables, much like Fourier or Laplace transformation. Conversely, model reduction can be achieved if the filter L is not strictly invertible. In that context, the Gaussian filter would be an invertible filter corresponding to a change of variables, while the top-hat and spectral cut-off filters do reduce the number of scales that contribute to the filtered solution $\overline{u}$.

The situation changes entirely if a nonlinear equation is being filtered by a convolution filter. To illustrate this, consider

$$\partial_t u + \partial_x F(u) = 0 \tag{3.23}$$

where $F(u)$ can be any smooth function. Application of a convolution filter L yields

$$\partial_t \overline{u} + \partial_x \overline{F(u))} = 0 \tag{3.24}$$

We may formally rewrite this as

$$\partial_t \overline{u} + \partial_x F(\overline{u}) + \partial_x([L, F](u)) = 0 \tag{3.25}$$

in which the commutator

$$[L, F](u) = L(F(u)) - F(L(u)) \tag{3.26}$$

was introduced. In cases where $F(u) = au$, i.e., F is a linear function of u, the commutator vanishes and the filtered equation which governs $\overline{u}$ is of identical type as the equation

governing u. Conversely, if $[L,F](u) \neq 0$ an altogether different equation emerges in which a so-called 'closure problem' emerges, i.e., in order to yield a closed PDE formulation that can be considered to predict $\overline{u}$ the commutator needs to be approximated by a so-called 'closure model' . Specifically, we need to propose a model m such that

$$[L,F](u) \approx m(\overline{u}) \tag{3.27}$$

implying a reduced model

$$\partial_t v + \partial_x F(v) + \partial_x m(v) = 0 \tag{3.28}$$

for the solution v which is thought to yield a relevant approximation of $\overline{u}$. Strictly speaking the closure model should be such that

$$\partial_x [L,F](u) \approx \partial_x m(\overline{u}) \tag{3.29}$$

which can be achieved by proposing a model m according to (3.27), i.e., at so-called flux-vector level, but actually requiring the model m to be such that the approximation is suitable at flux level (3.29) would yield more flexibility proposing the reduced model.

A well-known example of a nonlinear evolution equation is the Burgers equation

$$\partial_t u + \partial_x (\frac{1}{2} u^2) - \nu \partial_{xx} u = f \tag{3.30}$$

in which a characteristic quadratic nonlinearity arises, denoting the convection term, next to a dissipative term proportional to $\partial_{xx} u$, and a forcing f. In dimensionless form, this equation $\nu \equiv 1/Re$ represents the viscosity, where we also introduced the so-called Reynolds number $1/Re$.

The high complexity of turbulent flow arises from the action of the nonlinear convective fluxes. This can be illustrated by turning to the unforced viscous Burgers equation (3.30). If we consider an initial state of the form $u = \sin(kx)$, with wave-number k, then we find from insertion into (3.30) a total flux which is given explicitly by

$$\partial_t u \Big|_{t=0} = -\frac{k}{2} \sin(2kx) - \frac{k^2}{Re} \sin(kx) \tag{3.31}$$

We notice that there are two essentially different contributions to the flux. First, arising from the nonlinear convective flux, we see a contribution with wave number $2k$. This will induce a component in the solution with wave number $2k$ as well as the solution evolves. Second, the viscous flux gives a contribution with a wave number equal to that of the initial

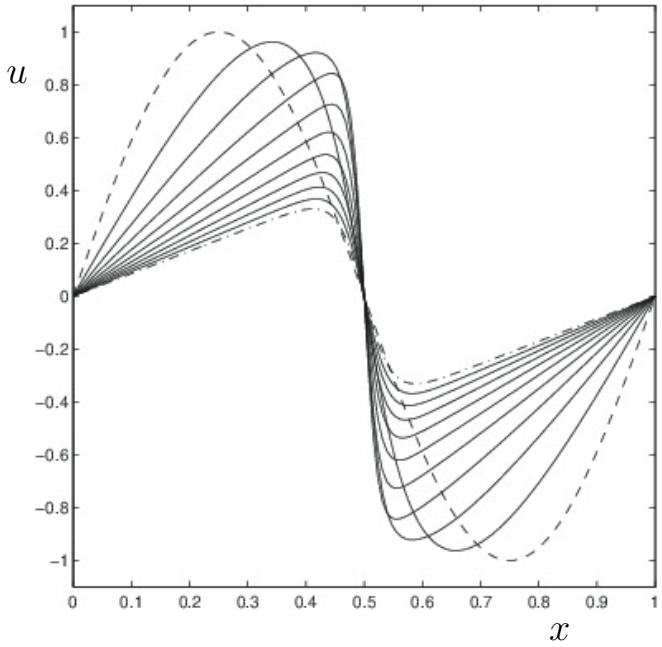


Figure 3.4: Evolution of the solution to the viscous Burgers equation at $Re = 100$. Shown are the initial condition $u(x,0) = \sin(2\pi x)$ (dashed), the intermediate solutions at $t = 0.1,\ 0.2,\ 0.3, \cdots$ (solid) and the solution at $t = 1$ (dash-dotted). Periodic conditions in x are applied.

state. This contribution implies that the amplitude of the initial $\sin(kx)$ will be reduced as the solution develops but it will not alter its spectral content.

In figure 3.4 the evolution of the initial state $u(x,0) = \sin(2\pi x)$ is shown, subject to periodic boundary conditions. As time progresses, further wave numbers will emerge in the solution due to the continuing (self-)interactions between solution components of different wave numbers. The solution develops a sharp gradient, which corresponds to high values of the wave number, next to slowly varying parts in which the solution varies approximately linearly. The latter regions are associated with small values of the wave number. Thus, from an initial state with only one wave number, the evolving Burgers solution is characterized by a number of wave numbers that differ considerably in size.

Application of a convolution filter to this equation yields for the reduced model

$$\partial_t \overline{u} + \partial_x(\frac{1}{2}\overline{u}^2) - \nu\partial_{xx}\overline{u} = \overline{f} - \partial_x(\frac{1}{2}(\overline{u^2} - \overline{u}^2)) \equiv \overline{f} - \partial_x\tau(u,\overline{u}) \approx f - \partial_x(m(\overline{u})) \qquad (3.32)$$

Here, the notation $\tau(u,\overline{u})$ denotes the so-called sub-filter stress. We observe that $\overline{u}$ is

operated on with the same nonlinear operator $\partial_t + u\partial_x - \nu\partial_{xx}$ as in the unfiltered Burgers equation (3.30). However, instead of a right-hand side containing the filtered forcing term f only, we now observe an additional closure term to the flux in the form of $-\partial_x \tau(u, \overline{u})$. By itself, this does not constitute a useful reduced model since both u and $\overline{u}$ are required to evaluate the flux for $\overline{u}$, while only $\overline{u}$ is available. That is why a model $m(\overline{u})$ needs to be supplemented.

There are many heuristic suggestions in literature for the sub-filter closure model m. The motivation for such models is often intuitive and lacks a solid theoretical basis. Nevertheless, the importance of solving turbulent flow problems forces pragmatic models to be adopted in order to make practical steps possible. This heuristic modeling step of course also introduces important uncertainties in the quality of the predictions, opening up new fields of mathematics and the requirement of careful validation of modeling steps.

The next major step is the discretization of the coarsened model. This will confront us with the fact that coarse discretization yields modifications of the equations of their own, which need particular attention.

3.4 Accuracy and stability of numerical methods

In the process of formulating a suitable finite-dimensional representation of the system of partial differential equations, a number of approximations regarding the properties of the desired solution are made. These approximations differ from method to method and may strongly influence the final discrete solution, in particular for features of the solution which are not well resolved by the computation. Two important basic factors that influence the appropriateness of numerical methods are accuracy and stability, which we will discuss after a general overview of the discretization process.

3.4.1 Physical space discretization

Many problems in science and technology are formulated in terms of a continuous conservation law given by

$$\partial_t U + \partial_j F_j(U) = 0 \quad ; \quad t > 0,\ \mathbf{x} \in V \tag{3.33}$$

Here, U denotes the 'state-vector', F_j the total flux-vector in the x_j direction, t represents time, and V the flow-domain with boundary ∂V. Partial derivatives with respect to time t and spatial coordinate x_j are denoted by ∂_t and ∂_j respectively. Summation over repeated

indices is implied here, i.e., $\partial_j F_j = \partial_1 F_1 + \partial_2 F_2 + \partial_3 F_3$ in three dimensions, expressing the divergence operator acting on $\mathbf{F} = [F_1, F_2, F_3]$. The conservation property of this equation is clear upon integrating (3.33) over an arbitrary volume $\widetilde{V}$. According to the divergence theorem of Gauss, the divergence of $\mathbf{F}$ integrated over $\widetilde{V}$ can be expressed as a surface integral of $F_j n_j$ over the boundary $\partial\widetilde{V}$, in which n_j denotes the j-th component of the outward pointing normal vector $\mathbf{n}$ on $\partial\widetilde{V}$. Correspondingly, the integral over $\widetilde{V}$ of U changes with time only through contributions that enter along the surface $\partial\widetilde{V}$ and does not contain any contribution from the inner of the volume $\widetilde{V}$. Stated differently, if nothing enters or leaves through the boundary, then the integral of U over $\widetilde{V}$ is constant, i.e., U is 'conserved'. A simple, one-dimensional linear example is the advection-diffusion-reaction equation

$$\partial_t U + a\partial_x U - \nu\partial_{xx} U = \partial_t U + \partial_x(aU - \nu\partial_x U) = -cU + d \tag{3.34}$$

which includes advection ($a\partial_x U$), diffusion ($\nu\partial_{xx} U$), as well as a source term (cU) and an autonomous external forcing (d) in terms of constants a, ν, c and d.

The type of formulation (3.33) determines the evolution of an initial condition $U(\mathbf{x}, 0) = U_0(\mathbf{x})$. The complete formulation requires the specification of boundary conditions, denoted by $B(U) = b$ for $\mathbf{x} \in \partial V$. The operator B may contain various types of boundary conditions and b represents externally imposed properties of the unknown solution U. As an illustration, in one spatial dimension typical boundary conditions include Dirichlet, e.g., $u = b$, Neumann, e.g., $\partial_x u = b$ or mixed conditions, e.g., a Robin boundary condition $u + \alpha\partial_x u = b$.

The complete initial boundary value problem can be discretized in different ways. In physical-space discretization one typically starts by introducing a computational grid covering the flow-domain V and represents the desired solution by its values in discrete grid points, often collocated, i.e., with values assigned at the grid points themselves, or staggered, i.e., with values assigned at the centers of the grid cells. Apart from the word 'grid' one may also often read the word 'mesh' for the network of discrete locations used to approximate the solution U.

Finite volume discretization

We consider discretization on a structured computational grid of 'primary' points $\{\mathbf{x}_{ijk}\}$ on the domain V. The introduction of the grid $\{\mathbf{x}_{ijk}\}$ defines so-called grid-cells as the volumes marked by the neighboring grid-points on its corners. Likewise, staggered grid-cells

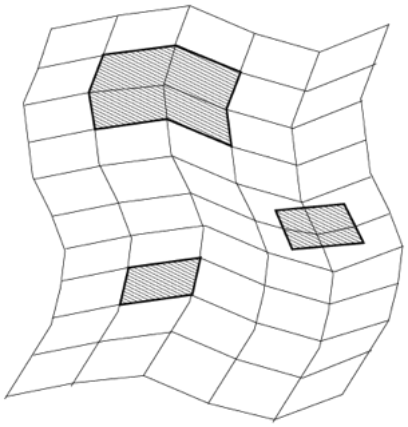


Figure 3.5: An example of a structured grid in two dimensions with several different control volumes indicated by the hashed regions. Both vertex and cell-based control volumes are shown, consisting of a single or multiple grid cells.

can be introduced in terms of the 'staggered' grid defined by grid points placed 'halfway' in grid cells formed by the primary grid points, e.g., $x_{i+\frac{1}{2}} = (x_i + x_{i+1})/2$ in one dimension. Combination of one or more grid-cells defines so-called 'control volumes' V_{ijk}. Some examples are shown in figure 3.5.

There are a number of different but closely related ways to obtain a discrete representation of (3.33) in physical space. An intuitively appealing way which directly reflects the conservation property is obtained in so-called 'finite volume' methods. The basis for this discretization arises by integrating (3.33) over an arbitrary control volume V_{ijk}, implying

$$\int_{V_{ijk}} \partial_t U \; dV + \int_{V_{ijk}} \partial_m F_m \; dV \equiv |V_{ijk}| d_t U_{ijk}(t) + \int_{\partial V_{ijk}} F_m \; n_m \; dS = 0 \tag{3.35}$$

where use was made of the Gauss divergence theorem. In this formulation, ∂V_{ijk} denotes the boundary and $|V_{ijk}|$ the volume of V_{ijk}. In (3.35) we introduced the unknowns $\{U_{ijk}\}$ by

$$U_{ijk}(t) = \frac{1}{|V_{ijk}|} \int_{V_{ijk}} U(\mathbf{x}, t) \; d\mathbf{x} \tag{3.36}$$

which are time-dependent volume averages of the unknown function U. After dividing

(3.35) by $|V_{ijk}|$ we obtain the discrete dynamical system

$$d_t U_{ijk}(t) + f_{ijk} = 0 \tag{3.37}$$

where we introduced the discrete flux

$$f_{ijk} = \frac{1}{|V_{ijk}|} \int_{\partial V_{ijk}} F_m \; n_m \; dS \tag{3.38}$$

The system of ordinary differential equations (3.37) has been obtained through the so-called 'method of lines' , or 'semi discretization' in which the time-variable is considered continuous, while space has been divided up into control volumes. To complete this discrete formulation, we add the initial condition

$$U_{ijk}(0) = \frac{1}{|V_{ijk|}} \int_{V_{ijk}} U_0(\mathbf{x}) \; d\mathbf{x} \tag{3.39}$$

where U_0 specifies the solution at time $t = 0$. Notice that no approximations have been required to derive (3.37) from (3.33). Of course, the numerical solution will only yield an approximation of the average of U over the various control volumes, implying that we have a discrete, finitely resolved representation of the solution. Moreover, the discrete flux f_{ijk} needs to be approximated to arrive at a dynamical system that can be used for computations. Often, an algebraic expression in terms of grid points and unknowns appearing in the discrete formulation is adopted. The precise approximations used in the specification of f_{ijk} will determine the properties of the resulting dynamical system (3.37).

3.4.2 Formal order of accuracy

Consider a uniform grid with grid-spacing h. A spatial discretization method is said to have formal order of accuracy p if the truncation error is of order h^p if $h \to 0$. The error is defined as a measure of the difference between the analytical and numerical solutions. The order of accuracy of a discretization method can be determined by performing a Taylor expansion of the desired solution, which is assumed to be sufficiently differentiable. As an example, consider the linear equation

$$\partial_t u + \partial_x u = 0 \tag{3.40}$$

in one spatial and temporal variable. The term $\partial_x u$ can, e.g., be discretized around the grid-point $x_i = ih$ as:

$$\partial_x u(x_i, t) = \frac{1}{h}(u_i - u_{i-1}) \tag{3.41}$$

where we identify $u_i = u(x_i, t)$. The Taylor expansion of u_{i-1} around x_i is

$$u_{i-1} = u_i - h\partial_x u_i + \frac{1}{2}h^2\partial_{xx}u_i + \quad \ldots \tag{3.42}$$

where $\ldots$ denote terms of higher order in h. This expansion implies that

$$\frac{1}{h}(u_i - u_{i-1}) = \partial_x u_i - \frac{1}{2}h\partial_{xx}u_i + \quad \ldots \tag{3.43}$$

from which we infer that the error is of order h^1 and proportional to the second derivative of u. This spatial discretization method is said to be first order accurate. In particular, the discretization is consistent, i.e., the approximate derivative on the left hand side of (3.43) approaches the desired $\partial_x u$ at the location x_i in case we follow a sequence of refinements of the grid in which $h \to 0$. Following the same reasoning, it is not difficult to show that the central discretization

$$\partial_x u_i = \frac{u_{i+1} - u_{i-1}}{2h} - \frac{1}{3}h^2\partial_{xxx}u_i + \quad \ldots \tag{3.44}$$

is second order accurate with a discretization error proportional to the third derivative of u.

In more dimensions, the evaluation of the formal order of accuracy of a method on a general, non-uniform grid, can proceed along the same lines but is technically more tedious. In such cases, also terms related to the smoothness of the grid lines come into play, as well as measures for the local grid-stretching. As a general rule, grids should be as near to a uniform grid as possible, in order to preserve as much of the formal accuracy-potential as possible.

Accuracy at coarse resolution

If the formal order of accuracy $p > 0$, then a sufficiently fine resolution will correspond to a small error for smooth solutions. In practice, one would favor methods which have a (relatively) large value of p and display convergence behavior consistent with their formal order of accuracy. Preferably, such asymptotic convergence should express itself also on quite coarse grids in which the grid-spacing h could even be comparable to length scales ℓ characteristic of the solution itself. Empirically, however, higher-order methods display their asymptotic error-dependence only if h is significantly smaller than ℓ. Thus, on coarse grids, it is not obvious whether a formally higher order method should be preferred over a formally lower order method. Maintaining mathematical structure such as the conservation form or respecting skew-symmetry [27] appears very important at coarse resolutions.

3.4.3 Modified equation: dissipation and dispersion

The 'physical' implications of the truncation error of a spatial discretization method are relevant as these affect the dynamics of the small scales in a numerical solution. These characteristics may be interpreted, e.g., as dissipative or dispersive. The long-time evolution of a numerical solution may be altered considerably due to accumulated truncation-error effects. As an example, the spatial discretization method in (3.41) has a truncation error proportional to the second derivative of the solution cf. (3.43). So instead of solving $\partial_t u + \partial_x u = 0$ one effectively considers, to leading order in the truncation error, the so-called modified equation

$$\partial_t u + \partial_x u = \frac{1}{2} h \partial_{xx} u \tag{3.45}$$

if (3.41) is adopted. Although the term which represents the error may be controlled by reducing h, the physical nature of this term remains dissipative, leading to numerical smoothing of the solution, particularly on coarse meshes and after sufficiently long integration times.

To quantify the dissipation, we consider periodic solutions to (3.40) which have the property that $u(x + \ell, t) = u(x, t)$ for some period ℓ, and define the average total kinetic energy of the solution by

$$E(t) = \frac{1}{\ell} \int_0^\ell \frac{1}{2} u^2(x, t)\ dx \tag{3.46}$$

One may readily show that the energy is conserved, i.e., $dE/dt = 0$ for (3.40). However, the use of (3.41) in the numerical treatment implies for the modified equation, after some partial integration

$$\frac{dE}{dt} = \frac{1}{\ell} \int_0^\ell u \partial_t u\ dx = -\frac{h}{2\ell} \int_0^\ell (\partial_x u)^2\ dx \leq 0 \tag{3.47}$$

which shows that the exact conservation property for the kinetic energy E in (3.40) is no longer maintained in the computational dynamical system that arises after the spatial discretization (3.41). The dominant error induced by this discretization is said to contribute to the dissipation. Turning to the central scheme (3.44) a similar calculation in which we only include the dominant truncation error shows that

$$\frac{dE}{dt} \sim \int_0^\ell u \partial_{xxx} u\ dx \sim \int_0^\ell \partial_x u \partial_{xx} u\ dx = \int_0^\ell \frac{1}{2} \partial_x ((\partial_x u)^2)\ dx = 0 \tag{3.48}$$

Consequently, to leading order in the truncation error, the central scheme does not contribute to the dissipation. In fact, it can be shown that all higher order terms of this scheme do not give rise to dissipative effects in linear equations such as (3.45).

It may be shown that the dominant effect of the truncation error of the central scheme, i.e., of the term $\sim \partial_{xxx} u$, is dispersive. Such dispersion also arises as the next higher order numerical effect associated with the upwind scheme (3.41). This can be illustrated by considering a running wave solution of the form $u = \exp(\mathrm{i}(kx + \omega t))$. If we substitute this in (3.40) we obtain the dispersion relation $\omega = -k$ to be satisfied by the solution, i.e., we recover the proper wave speed. Inserting this wave solution into the dominant modified equation that corresponds with the central discretization, i.e., $\partial_t u + \partial_x u = -h^2 \partial_{xxx} u/3$ we obtain the modified dispersion relation

$$\omega = -k + \frac{1}{3} h^2 k^3 = -k(1 - \frac{1}{3} h^2 k^2) \tag{3.49}$$

The deviations from the original dispersion relation become large in case hk becomes large, which arises if the spatial resolution of the wave is rather coarse.

The type and amount of change in the physical properties of the computational dynamical system compared to those of the original formulation depend strongly on the discretization method that was introduced. Typically, next to dissipative effects, popular discretization methods also introduce dispersion as illustrated in the simple example above. The associated effects may be particularly important in relation to the long-time behavior at fairly low spatial resolution. The modified equation analysis provides a first clue regarding the type of alterations due to the numerical scheme, on fine grids and for linear problems. It remains mainly a matter of trial and error to quantify suitable numerical settings and methods for highly nonlinear problems.

3.4.4 Stability

Apart from accuracy considerations and modification of the basic equation associated with the truncation error of the spatial discretization scheme, numerical stability of the computational dynamical system is an essential property.

In order to capture the temporal evolution of a solution, the discretized system of equations that results after the method of lines needs to be integrated in time. The time-integration method should be consistent and accurate, analogous to the discussion above for spatial discretization. In addition, the time integration should be 'stable'. This is typically analysed by considering the stability of time integration of a linear equation of the form $u' = \lambda u$ for some complex number λ. Here λ is thought to approximate a characteristic local transport velocity of the actual nonlinear system of differential equations, obtained from the quasi-linear formulation.

For illustration purposes, in one spatial dimension, one has two equivalent descriptions for the partial differential equations:

$$\partial_t u + \partial_x(f(u)) = \partial_t u + f'(u)\partial_x u = 0 \tag{3.50}$$

with nonlinearity f and derivative $f' = df/du$ defining the quasi-linear formulation. The (local) value of $f'(u)$ for a specific solution u defines the relevant (local) rate λ. This can readily be extended to systems of equations and to more spatial dimensions, involving the Jacobi matrix of the nonlinearity f. A rigorous analysis is possible for linear systems with constant λ. We consider a simple example next to illustrate the so-called Neumann analysis.

Neumann analysis

The stability of a simulation method depends on both the spatial discretization and the time-integration. For many numerical methods, the stability depends on the time-step; the method is stable only if Δt is in a certain 'stability interval'. Outside this interval, the numerical solution is unstable and may rapidly diverge to unrealistic values. The bounds of this 'stability interval' may depend on, e.g., the mesh size and parameters in the equation.

In order to determine the stability interval, a Neumann analysis on an approximate, linearized equation is a useful approach [28]. Such an analysis consists of two parts: first a Fourier analysis of the spatial discretization method is made which leads to a system of equations governing the Fourier modes. Next, the stability region of the time-integration method is determined. As an example, we consider spatially periodic solutions to the convection-diffusion equation

$$\partial_t u + a\partial_x u = \mu\partial_{xx} u \tag{3.51}$$

with positive constants a and μ. We assume periodicity $u(x+1,t) = u(x,t)$ and use a uniform grid $x_j = jh$ with $h = 1/N$. Using second order accurate central differences we find

$$\frac{du_j}{dt} + \frac{a}{2h}(u_{j+1} - u_{j-1}) = \frac{\mu}{h^2}(u_{j+1} - 2u_j + u_{j-1}) \tag{3.52}$$

for $j = 1, \ldots, N$. The periodicity condition implies $u_N = u_0$ and $u_{N+1} = u_1$. This system of equations has a steady solution denoted by $\overline{u}$. The difference $v_j^n = u_j^n - \overline{u}_j$ can be shown to obey the same equation as u in (3.52). Here, we introduced the notation u_j^n for the solution at location jh and time-level n corresponding to time $n\Delta t$. Periodicity in space

implies that the difference can be expanded in a Fourier series (assuming N to be even for convenience) as:

$$v_j(t) = \sum_{k=-N/2}^{N/2} b_k(t) \exp(2\pi \mathrm{i} jk/N) \tag{3.53}$$

where $b_k(t)$ denotes the k-th Fourier coefficient of the difference. The evolution of the set $\{b_k\}$ can be found from substitution of the expansion of v in the discrete system of equations (3.52). This implies

$$\frac{db_k}{dt} + \frac{\mathrm{i}a}{h} \sin(2\pi k/N) b_k = \frac{2\mu}{h^2} \Big(\cos(2\pi k/N) - 1 \Big) b_k \tag{3.54}$$

for $k = -N/2, \ldots, N/2$, which can be written concisely as $db_k/dt = z_k b_k$ where z_k is referred to as the Fourier-symbol. The set of complex numbers $\{z_k\}$ for $k = -N/2, \ldots, N/2$ is called the Fourier footprint of the central spatial discretization method for the one-dimensional, scalar convection-diffusion equation. In general, the Fourier footprint is contained in an ellipse which lies to the left of the imaginary axis and touches it at the origin. The imaginary part of z_k corresponds to the convective term and the negative real part corresponds to the dissipative term.

To complete the determination of the stability time interval for a computational method we need to specify the time-integration method. We turn to a global sketch of such methods next and complete the Neumann analysis afterwards.

Runge-Kutta methods

In simulations, Runge-Kutta methods are very often employed, both for the calculation of a steady-state solution [29] and for time accurate simulations. The advantages of explicit Runge-Kutta methods are the low cost per time step and the large stability region. Consider the discretized form of a system of evolution equations

$$d_t \mathbf{u}_i + \mathbf{f}_i(\mathbf{u}) = \mathbf{0} \tag{3.55}$$

where $\mathbf{u}_i$ is the solution at a grid point labeled i and $\mathbf{f}_i$ represents the numerical flux in the same grid point. A general form of a Runge-Kutta method is [28]

$$\begin{aligned} \mathbf{u}_i^{(0)} &= \mathbf{u}_i^n \\ \mathbf{u}_i^{(k)} &= \mathbf{u}_i^{(0)} - \Delta t \sum_{j=0}^{m} \beta_{kj} \mathbf{f}_i(\mathbf{u}^{(j)}) \quad \text{for} \quad k = 1, \ldots, m \\ \mathbf{u}_i^{n+1} &= \sum_{k=1}^{m} \gamma_k \mathbf{u}_i^{(k)} \end{aligned} \tag{3.56}$$

where the label n refers to the time-level and (k) identifies the k-th Runge-Kutta stage. Moreover, β_{kj} and γ_k are real numbers for $k = 1, \ldots, m$ and $j = 0, \ldots, m$ and m is the number of Runge-Kutta stages. This general formulation of Runge-Kutta methods yields an explicit method provided $\beta_{kj} = 0$ for $j \geq k$. The explicit Runge-Kutta schemes are so-called one-step schemes which involve knowledge of only the previous solution.

For time-dependent calculations one may determine the coefficients in a Runge-Kutta scheme such that only a few intermediate solutions need to be stored. Instead of storing m intermediate solutions as required in the general formulation (3.56) one may obtain schemes with fewer intermediate solutions at the price of a lower formal order of accuracy than can maximally be obtained. This can be achieved, e.g., by taking $\gamma_k = 0$ for $k < m$ and $\beta_{kj} = 0$ for $j \neq k - 1$. With this choice, only the old solution $\mathbf{u}^n$ and the actual intermediate solution $\mathbf{u}^{(k)}$ have to be stored. These are called compact storage schemes which can be written as

$$\begin{aligned} \mathbf{u}_i^{(0)} &= \mathbf{u}_i^n \\ \mathbf{u}_i^{(k)} &= \mathbf{u}_i^{(0)} - \Delta t \beta_k \mathbf{f}_i(\mathbf{u}^{(k-1)}) \quad \text{for} \quad k = 1, \ldots, m \\ \mathbf{u}_i^{n+1} &= \mathbf{u}_i^{(m)} \end{aligned} \tag{3.57}$$

As an example, the two-stage compact storage Runge-Kutta scheme arises by choosing $\beta_1 = 1/2$ and $\beta_2 = 1$. An important explicit scheme for turbulent flow is the four-stage scheme with $\beta_1 = 1/4$, $\beta_2 = 1/3$, $\beta_3 = 1/2$ and $\beta_4 = 1$ [31]. Using Taylor expansions one may show that this four-stage scheme is fourth-order accurate for linear equations and second order for nonlinear equations.

The formal order of accuracy of Runge-Kutta methods can, in principle, be increased arbitrarily by increasing the number of stages m. It can be proved that the order of accuracy can not be higher than the number of stages and that only up to $m = 4$ the order

can be equal to the number of stages. The Runge-Kutta methods of maximum order are known as the classical Runge-Kutta methods. For $m = 4$ the only non-zero coefficients are $\beta_{10} = \beta_{43} = 1,\ \beta_{21} = \beta_{32} = 1/2$, $\gamma_1 = \gamma_4 = 1/6$, $\gamma_2 = \gamma_3 = 1/3$. The Euler forward method is the only first-order accurate one-stage Runge-Kutta method. A general determination of the order of accuracy of Runge-Kutta schemes and of the coefficients and number of stages required to achieve a given order can best be done using symbolic manipulation software [30].

Stability time-step

In the previous chapter, we introduced the simple Euler forward scheme and showed that the stability of this scheme implies an upper limit on the allowed time step. This is characteristic of all explicit time-integration schemes and is a large disadvantage of these methods. Generally, it is impossible to determine the stability region analytically. We will illustrate the commonly adopted Neumann analysis for the second and fourth-order compact storage Runge-Kutta schemes.

The traditional approach to determining the stability of Runge-Kutta methods proceeds in a few steps. First, one needs to select a suitable model equation. Since convective transport is a significant process in turbulent flows we consider

$$\partial_t u + a\partial_x u = 0 \tag{3.58}$$

where we introduced the transport velocity a. Semi-discretization of this equation with a simple spatial discretization implies

$$\frac{du_i}{dt} + \frac{a}{\Delta x}(u_i - u_{i-1}) = 0 \tag{3.59}$$

where Δx denotes the grid spacing. Traditionally, this equation suggests considering the stability of

$$\frac{du}{dt} = \lambda u \quad ; \quad \lambda = \frac{|a|}{\Delta x} \tag{3.60}$$

This model equation can also be suggested by substituting $u = \exp(ikx)$ into the more general equation $du/dt = f(u)$, after having linearized f which corresponds to $a = f'(u)$ as a local transport velocity. In (3.54) we obtained a similar basic equation in which the parameter λ was identified with the complex Fourier symbol.

The next step is to apply the selected time-stepping scheme. For the two-stage Runge-Kutta scheme, we find

$$\begin{aligned} u^{n+1} &= u_2 = u_0 + \lambda\Delta t u_1 = u_0 + \lambda\Delta t(u_0 + \frac{1}{2}\lambda\Delta t u_0) \\ &= u^n\left(1 + \lambda\Delta t + \frac{1}{2}(\lambda\Delta t)^2\right) \end{aligned} \tag{3.61}$$

This relates the solution at time-level $n+1$ to the solution at time-level n in the form $u^{n+1} = g(z)u^n$ where g denotes the amplification factor and $z = \lambda\Delta t$. For the two-stage scheme $g_2(z) = 1 + z + z^2/2$. Following the same steps for the four-stage scheme (3.57) yields

$$g_4(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24} \tag{3.62}$$

The system is stable when the amplification factor $g(z)$ satisfies $|g(z)| \leq 1$. Further analytical characterization of the stability region is not possible and therefore the next step is to plot contours of $|g(z)|$ in the complex z-plane. The region where the amplification factor is smaller than 1 is the stability region. The variable z in the model equation $du/dt = zu$ is in general taken to be a complex number where the real and imaginary parts represent viscous and convective flux-contributions in the discretized flow equations respectively. In figure 3.6 the stability region of (3.57) is shown for $RK2$ and $RK4$.

The restriction on the time-step such that the solution to the model equation is integrated in a stable manner can be specified in some more detail. Referring to figure 3.6 we should choose z such that $|g(z)| \leq 1$. Since the stability region is a complex set in z-plane, one typically limits z to lie in a square (or rectangular) sub-domain of the stability region. In particular, since the convective contributions are significant, this implies $\lambda\Delta t \leq \sigma$ where σ is the intersection of the contour $g(z) = 1$ with the imaginary axis $y \geq 0$. This yields

$$\Delta t \leq \frac{\sigma}{\lambda} = \frac{\sigma\Delta x}{|a|} \quad ; \quad \sigma_2 = 0 \quad ; \quad \sigma_4 = 2.8 \tag{3.63}$$

We observe that $RK2$ does not enclose an interval on the imaginary axis while $RK4$ contains a portion which is similar in size to the intersection with the negative real axis. Therefore, from this analysis, $RK2$ is found not to be suitable for convectively dominated flow computations while $RK4$ appears a suitable method. It can be shown that all methods RKn with $n \geq 3$ contain an interval on the imaginary axis enclosed by the $g = 1$ contour. These methods are hence acceptable for convectively dominated turbulent flows; in practice $RK4$ combines computational effort with sizeable stability time-step and is preferred in a number of applications.

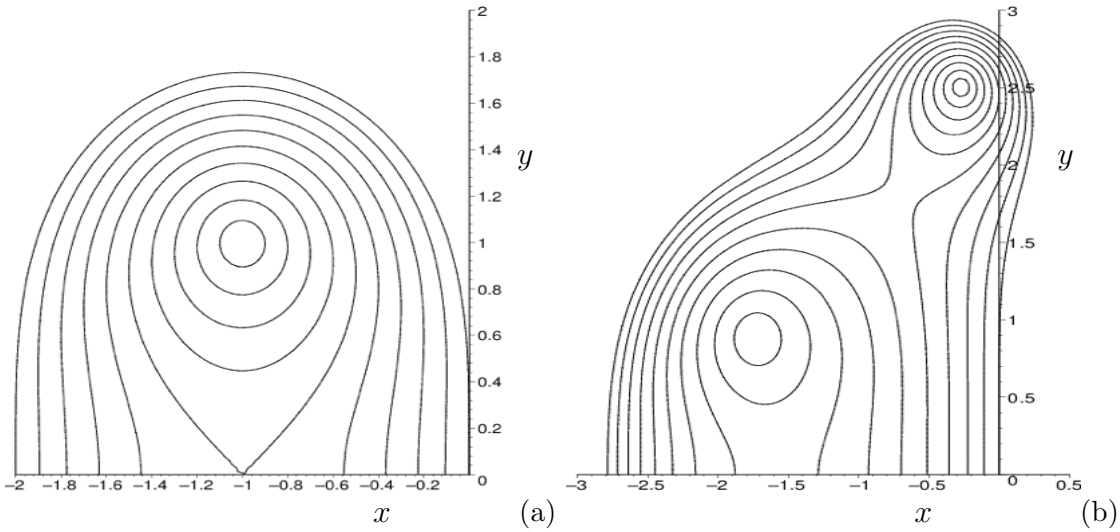


Figure 3.6: Stability region of the two-stage (a) and four-stage (b) compact storage Runge-Kutta scheme. Contours of the growth rate between 0 and 1 with interval 0.1 are plotted. Only the upper parts of the stability regions are shown in view of the symmetry about the x-axis.

3.5 Numerical spatial derivatives

In this subsection, we will consider general finite difference methods. We will introduce a framework to construct such methods of arbitrary order of accuracy and consider the so-called effective or modified wavenumber analysis to illustrate the effects that arise when these methods are applied. We will also consider explicit filtering and describe numerical quadrature in the same framework.

3.5.1 Basic discretization of derivatives

Basic discretization of first order derivatives

A basic finite difference discretization arises by combining values $\{u_i\}$ of the function u on a grid $\{x_i\}$. The approximation of the derivative of u in the point x_j will be written as $\delta_x u_j$. Simple methods require only two nearby values of u, e.g.,

$$\delta_x^{(1)} u_j = \frac{u_{j+1} - u_{j-1}}{x_{j+1} - x_{j-1}} \quad ; \quad \delta_x^{(2)} u_j = \frac{u_{j+1} - u_j}{x_{j+1} - x_j} \quad ; \quad \delta_x^{(3)} u_j = \frac{u_{j-1} - u_j}{x_{j-1} - x_j} \tag{3.64}$$

These methods differ in the actual nearby grid points that are involved in the approximation. In $\delta_x^{(1)}$ the two neighboring points are involved; this method is called a central method. The other two discrete operators take information either from the left or from the right of x_j. The computational effort is the same in these three methods. Their formal order of accuracy is different as may be inferred from expanding u in a Taylor series:

$$\begin{aligned} u_{j\pm1} &= u_j + (x_{j\pm1} - x_j)(u_x)_j + \frac{1}{2}(x_{j\pm1} - x_j)^2(u_{xx})_j \\ &+ \frac{1}{6}(x_{j\pm1} - x_j)^3(u_{xxx})_j + \ \ldots \end{aligned} \tag{3.65}$$

where $(u_x)_j = \partial_x u(x_j)$. Combining the expansions of u_{j+1} and u_{j-1} we arrive, e.g., at

$$\delta_x^{(1)} u_j = \frac{u_{j+1} - u_{j-1}}{x_{j+1} - x_{j-1}} = (u_x)_j + \frac{1}{2}(u_{xx})_j(x_{j+1} - 2x_j + x_{j-1}) + \ \ldots \tag{3.66}$$

Hence, we observe that $\delta_x^{(1)} u_j - (u_x)_j = \mathcal{O}(h_j - h_{j-1})$ where $h_j = x_{j+1} - x_j$. So, in general, for non-uniform grids this method of approximating $(u_x)_j$ is first-order accurate in the grid-spacing. For the special case that the grid is uniform we have $h_j = h_{j-1} = h$ and the first order contribution to the approximation error vanishes. On uniform grids, the central discretization is thus second-order accurate. From this simple illustration, we infer that a smoothly varying grid, i.e., grids for which $h_j \approx h_{j-1}$, will increase the observed level of accuracy, while the formal order of accuracy remains lower than that achievable on uniform grids.

Repeating the same calculation for $\delta_x^{(2)}$ we readily obtain

$$\delta_x^{(2)} u_j = (u_x)_j + \frac{1}{2}(u_{xx})_j(x_{j+1} - x_j) + \ldots \tag{3.67}$$

and hence the error, in this case, is proportional to h_j. For this 'skewed' approximation a gradual transfer to a uniform grid will not increase the formal order of accuracy.

One has to realize that the formal order of accuracy may be obtained only within the limit of sufficiently high spatial resolution. In actual simulations the grid spacing may frequently be too large, and the formal order of accuracy is only a rough indication of the reduction in the simulation errors one may obtain by grid refinement. Where formally a reduction of h by a factor of 2 would lead to an error reduced by a factor $(1/2)^q$ where q is the formal order of accuracy, the actual error reduction on coarse grids usually is much large-scale simulations, especially for higher order methods.

Basic discretization of second order derivatives

A simple finite difference approximation of a second-order derivative can be obtained along the same lines. As an illustration, we consider a central difference approximation of $(u_{xx})_j$. For this purpose we introduce $u_{j+1/2} = u(x_{j+1/2})$ where $x_{j+1/2} = (x_{j+1} + x_j)/2$ is a grid point exactly in the middle between x_j and x_{j+1}. By analogy with the definition of a second derivative, we have

$$(u_{xx})_j \approx \frac{(u_x)_{j+1/2} - (u_x)_{j-1/2}}{x_{j+1/2} - x_{j-1/2}} \tag{3.68}$$

If we approximate $(u_x)_{j+1/2}$ by a second order accurate central method, i.e., $(u_x)_{j+1/2} = (u_{j+1} - u_j)/(x_{j+1} - x_j)$ we obtain

$$\delta_{xx} u_j = a_{j,j+1} u_{j+1} - a_{j,j} u_j + a_{j,j-1} u_{j-1} \tag{3.69}$$

where

$$\begin{aligned} a_{j,j+1} &= \frac{2}{(x_{j+1} - x_j)(x_{j+1} - x_{j-1})} \\ a_{j,j} &= \frac{2}{(x_{j+1} - x_j)(x_j - x_{j-1})} \\ a_{j,j-1} &= \frac{2}{(x_j - x_{j-1})(x_{j+1} - x_{j-1})} \end{aligned} \tag{3.70}$$

On a uniform grid, this approximation reduces to the well-known form

$$\delta_{xx} u_j = \frac{1}{h^2} \Big\{ u_{j+1} - 2u_j + u_{j-1} \Big\} \tag{3.71}$$

which is a second-order approximation of $(u_{xx})_j$. On general, non-uniform grids, the approximation (3.69) can be shown to be only first-order accurate, again emphasizing the importance of smoothly varying grids.

3.5.2 Methods of higher order of accuracy

The above simple illustrations of finite difference discretizations can straightforwardly be generalized to higher order methods. To avoid technical complications we first consider the generalization of central discretizations on uniform grids. Extensions to non-uniform grids and general skewed schemes will be sketched afterward.

The generalization for first-order central discretization can be expressed as:

$$\delta_x^{(n)} u_j = \frac{1}{h} \sum_{l=-n}^{n} b_l u_{j+l} \tag{3.72}$$

where we incorporated $2n+1$ grid points. The coefficients $\{b_l\}$ need to be constructed in such a way that the approximation (3.72) satisfies certain criteria. Here, we consider schemes of maximal order of accuracy within the stencil $[x_{j-n},\ x_{j+n}]$. These follow from the requirements to discretize x^k exactly for $k = 0, .., 2n$. Since $d(x^k)/dx = kx^{k-1}$ this implies in x_j

$$\frac{1}{h} \sum_{l=-n}^{n} b_l ((j+l)h)^k = k(jh)^{k-1} \quad ; \quad k = 0,\ 1, ..., 2n \tag{3.73}$$

This constitutes a system of $2n+1$ linear equations for the $2n+1$ unknowns $\{b_l\}$. After some rewriting, using induction, one may show that this system is equivalent to

$$\sum_{l=-n}^{n} l^k b_l = \delta_{k1} \quad ; \quad k = 0, ..., 2n \tag{3.74}$$

This system of equations can be rewritten as $A\mathbf{b} = \delta$ where $\mathbf{b} = [b_{-n}, ..., b_n]$ and A and δ appropriately defined as in (3.74). It may be shown that A is non-singular and hence the desired maximal order scheme on $2n+1$ grid points exists and can be obtained uniquely by inverting A. The schemes that arise from this approach can be shown to introduce an error of $\mathcal{O}(h^{2n})$.

To generate some examples of high-order central schemes the use of symbolic manipulation software is very sensible. For $n = 1$ we obtain the central operator $\delta_x^{(1)}$ in (3.64). Going to higher order we get for $n = 2$ the well known fourth order accurate method

$$\delta_x u_j = \frac{1}{12h}(-u_{j+2} + 8u_{j+1} - 8u_{j-1} + u_{j-2}) + \mathcal{O}(h^4) \tag{3.75}$$

while for $n = 3$ we obtain

$$\delta_x u_j = \frac{1}{60h}(u_{j+3} - 9u_{j+2} + 45u_{j+1} - 45u_{j-1} + 9u_{j-2} - u_{j-3}) + \mathcal{O}(h^6) \tag{3.76}$$

The extension to the second derivative $(u_{xx})_j$ is straightforward. As before, we start with the requirement that polynomials up to some order are treated exactly. In this case $d^2(x^k)/dx^2 = k(k-1)x^{k-2}$ which implies in x_j

$$\frac{1}{h^2} \sum_{l=-n}^{n} c_l ((j+l)h)^k = k(k-1)(jh)^{k-2} \quad ; \quad k = 0,\ 1, ..., 2n \tag{3.77}$$

This linear system of equations yields $\{c_l\}$. At $n = 1$ we arrive at the second order accurate approximation (3.71). The next higher order scheme at $n = 2$ can be written as

$$\delta_{xx}u_i = \frac{1}{12h^2}\Big\{ - u_{j+2} + 16u_{j+1} - 30u_j + 16u_{j-1} - u_{j-2}\Big\} \tag{3.78}$$

while the sixth order scheme for $n = 3$ has coefficients $c_l = 1/90$, $-3/20$, $3/2$, $-49/18$, $3/2$, $-3/20$, $1/90$ for $l = -3, ..., 3$ respectively.

Another, more insightful derivation of approximations to the second derivative can be obtained by extending (3.68). For this purpose we consider the approximation of a term $\partial_x(\kappa(x)\partial_x u)$ where κ is a known function. In this 'conservation' form we recognize an 'inner' and an 'outer' first derivative and we can approximate this by a suitable combination of two first order derivatives; $\partial_x(\kappa(x)\partial_x u)$ and $\delta_x^{out}(\kappa(x)\delta_x^{in}u)$ where δ_x^{out} and δ_x^{in} can be determined as sketched above.

The main extension that is required is that the inner derivative should provide approximations in the staggered locations $x_{j+k/2}$ and the outer derivative should work with first derivatives in these staggered locations and return an approximation in the grid points x_j. As an example we may introduce

$$\delta_x^{in}u_{j+1/2} = \frac{u_{j+1} - u_j}{x_{j+1} - x_j} \quad ; \quad \delta_x^{out}u_j = \frac{u_{j+1/2} - u_{j-1/2}}{x_{j+1/2} - x_{j-1/2}} \tag{3.79}$$

Both $\delta_x^{in}u_{j+1/2}$ and $\delta_x^{out}u_j$ are second-order accurate approximations in the respective points $x_{j+1/2}$ and x_j. Combining these two discrete operators yields the approximation (3.68) as shown before. On uniform grids, this two-step construction corresponds to the construction following (3.77).

A more accurate construction along these lines can be obtained by approximating $(u_x)_{j+1/2} = \delta_x^{in}u_{j+1/2}$ using the maximal order central four point stencil involving $x_{j\pm1}$ and $x_{j\pm2}$. Likewise, the outer derivative is calculated in x_j using the approximate first order derivatives in $x_{j\pm1/2}$ and $x_{j\pm3/2}$ obtained with $\delta_x^{in}u_{j+k/2}$. The resulting scheme for the second derivative is hence performed in two steps. For the inner derivative, we find [32]

$$\delta_x^{in}u_{j+1/2} = \frac{1}{24h}\Big\{u_{j-1} - 27u_j + 27u_{j+1} - u_{j+2}\Big\} \tag{3.80}$$

In the second step, the outer derivative in the point x_j is calculated by applying the same discretization formula shifted 1/2 grid point down so as to include $u_{j-3/2}, ..., u_{j+3/2}$. Comparing (3.80) with (3.78) we notice that both schemes have the same formal order of accuracy on uniform grids but the first scheme requires a stencil which is 7 points wide

whereas the latter scheme only needs 5. Still, this 'inner-outer' approach to second-order derivatives is much more convenient in flow simulations where the viscosity may depend on the solution. Also, for a number of models this 'inner-outer' formulation is appropriate since it preserves the conservation form of the basic equations.

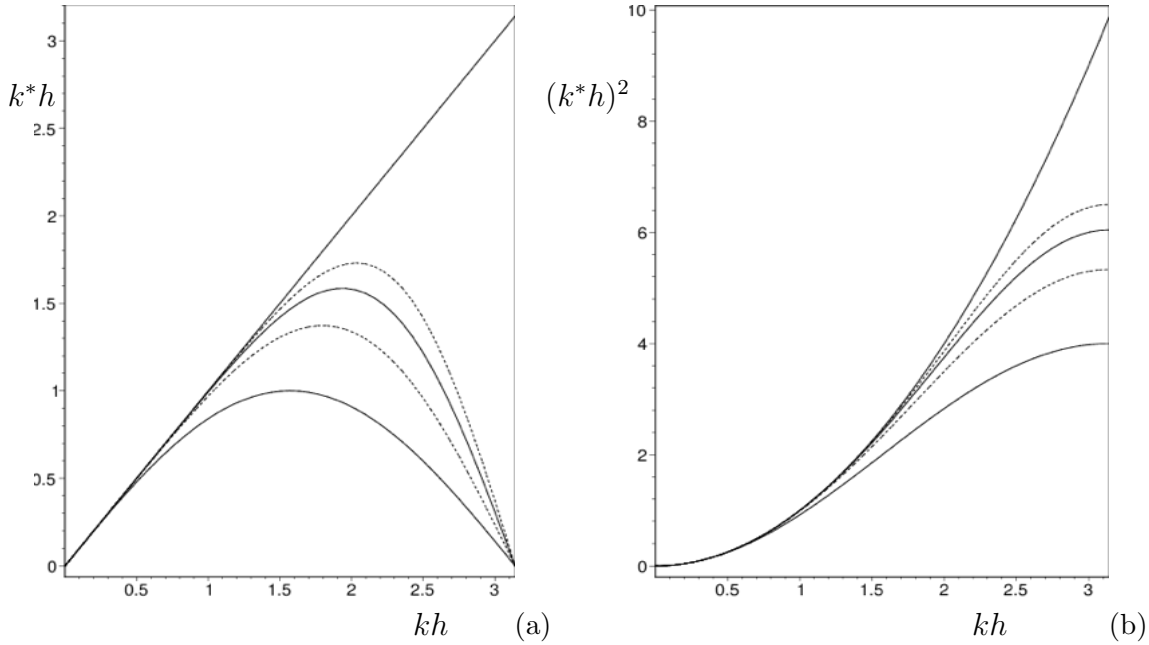


Figure 3.7: In (a) we show the modified wavenumber k^*h plotted against kh for a number of schemes of order h^{2n}. Solid lines correspond to $n = 1,\ 3$ and dash-dotted lines to $n = 2,\ 4$. The straight line corresponds to the exact wavenumber. In (b) the same is shown for second-order derivatives and we plotted $(k^*h)^2$ against kh where the parabola corresponds to the exact wavenumber.

3.5.3 Modified wavenumber analysis

Apart from the formal order of accuracy, which characterizes the treatment of well-resolved flow features, the effect of the discretization on the smaller structures in a flow is of major importance in actual flow simulations. A concise way to illustrate these effects is to consider the action of δ_x on a Fourier mode $u = \cos(kx)$. Analytically we know that $d(\cos(kx))/dx =$

$-k\sin(kx)$. If we turn to the simple central scheme (3.66) on a uniform grid we find

$$\begin{aligned}\delta_x(\cos(kx)) &= \frac{1}{2h}\cos(k(x+h)) - \frac{1}{2h}\cos(k(x-h)) \\ &= -\Big(\frac{\sin(kh)}{h}\Big)\sin(kx) = -k^*\sin(kx) \qquad (3.81)\end{aligned}$$

Hence, where analytically we would expect $-k\sin(kx)$ we infer that this numerical scheme returns a modified wavenumber $k^* = \sin(kh)/h$. In case $kh \ll 1$ we find $k^* = k - k^3h^2/6 + ...$ which shows that this scheme is second order accurate and the modified wavenumber approximates the actual wavenumber properly in case kh is sufficiently small. The latter condition may readily be interpreted in terms of sufficient spatial resolution per wave-length $\lambda = 2\pi/k$ of the mode. If we put $h = \lambda/N$ then $kh = 2\pi/N \ll 1$ if N is large enough.

For central finite differencing (3.72) one finds $\delta_x(\cos(kx))_i = -k^*\sin(kx_i)$ and the effective wave-numbers corresponding to a number of high-order schemes are plotted in figure 3.7. We observe that with increasing n the modified wavenumber approximates the exact value with high accuracy over a wider range of kh values. All these schemes introduce large errors for $kh \to \pi$ where the modified wavenumber of all central schemes is 0. The mode $kh = \pi$ corresponds to a so-called π-mode and on the numerical grid this implies $u_j = (-1)^j$. Clearly, details in the flow that fluctuate between $+1$ and -1 over distances h are poorly resolved and the π-mode 'error' $k^* \to 0$ in the central schemes is actually an appealing feature of these schemes. Components in a numerical solution proportional to the π-mode do not grow since the corresponding flux is zero. In figure 3.7(b) we show the equivalent results for the approximation of the second order derivative. The exact result would be $d^2(\cos(kx))/dx^2 = -k^2\cos(kx)$ while numerically we find $\delta_{xx}(\cos(kx))_i = -(k^*)^2\cos(kx)$. Again we see good approximations if $kh \ll 1$ but all central schemes do not provide sufficient amplitude for the second order derivative. This implies that the corresponding numerical viscous flux is too low.

Modified wave-number and required spatial resolution

The modified wave-number analysis can also be used to estimate up to which wavenumber k_{max} the spatial discretization is reliable. To quantify this we can introduce several measures for the error $|k^* - k|$. A simple choice is to define $k_{max}h$ as the location where k^*h has its maximum. For the second order scheme the maximum is at $\pi/2$ and hence all modes $0 \le k \le k_{max} = \pi/(2h)$ could be regarded as properly represented on a grid with grid spacing h, using this scheme. Turning to the fourth order scheme we observe that the

maximum is achieved at a somewhat higher value of ≈ 1.8. Correspondingly $k_{max} \approx 1.8/h$ and in this schematic, rough estimation the range of modes that is 'properly' represented has increased by about 15% when switching from a second to a fourth order scheme. This does not seem very much but in three spatial dimensions the gain can be useful. Moving to yet higher order schemes in this family we notice that the additional extension of the 'useful' range of modes is rather limited as a function of n. The additional computational effort associated with these higher order schemes does not seem to be warranted by this. Moreover, the ever wider stencils of these high-order schemes make the implementation of boundary conditions gradually more difficult. To achieve very high accuracy in simple flow domains it is therefore not advised to turn to finite difference methods. Good alternatives exist in the form of implicit discretization methods and spectral schemes that will be sketched later.

Index

Bibliography

[1] Antoulas, A.C. *Approximation of Large-Scale Dynamical Systems*, SIAM Advances in Control and Design, Philadelphia, 2005.

[2] Meinsma, G. *Linear System Theory*, Syllabus Linear System, version August 2024.

[3] Meinsma, G., Morsch, H.G. ter, Stoorvogel, A.A., and Zwart H., *Signals and Transforms*, Syllabus Signals and Transforms, version September 2024.

[4] Borthwick, D., *Introduction to Partial Differential Equations*, Springer, ISBN 978-3-319-48934-6, 2016.

[5] Ghosal, S., Lund, T.S., Moin, P., Akselvoll, K.: 1995. A dynamic localization model for large-eddy simulation of turbulent flows. *J. Fluid Mech.*, **286**, 229.

[6] Vreman, A.W., Geurts, B.J., Kuerten, J.G.M.: 1997. Large-eddy simulation of the turbulent mixing layer. *J. Fluid Mech.*, **339**, 357.

[7] Geurts, B.J., Vreman, A.W., Kuerten, J.G.M., van Buuren, R.: 1997. Non-commuting filters and dynamic modeling for LES of turbulent compressible flow in 3d shear layers. *Direct and Large-Eddy simulation II, Grenoble*. Eds: P.R. Voke, L. Kleiser, J.P. Chollet. Kluwer Academic Publishers, 47.

[8] Vasilyev, O.V., Lund, T.S., Moin, P.: 1998. A general class of commutative filters for LES in complex geometries. *J. Comp. Phys.*, **146**, 82.

[9] Vreman, A.W., Geurts, B.J., Kuerten, J.G.M.: 1994. Realizability conditions for the turbulent stress in large eddy simulation. *J. Fluid Mech.*, **278**, 351.

[10] Ortega, J.M.: 1987. *Matrix Theory*. Plenum Press. New York.

[11] Schumann, U.: 1977. Realizability of Reynolds-stress turbulence models. *Phys. of Fluids*, **20**, 721.

[12] Vachat, R. Du: 1977. Realizability inequalities in turbulent flows. *Phys. of Fluids*, **20**, 551.

[13] Rudin, W.: 1973. *Functional analysis*. McGraw-Hill.

[14] Debliquy, O., Knaepen, B., Carati, D.: 2001. A dynamic subgrid-scale model based on the turbulent kinetic energy. *Direct and large-eddy simulation - IV*, Eds: B.J. Geurts, R. Friedrich, O. Métais. Kluwer Academic Publishers, 89.

[15] Piomelli, U., Cabot, W.H., Moin, P., Lee, S.: 1990. Subgrid-scale backscatter in transitional and turbulent flows. *CTR Proceedings of the Summer Program*, 19.

[16] Germano, M.: 1992. Turbulence: the filtering approach. *J. Fluid Mech.*, **238**, 325.

[17] Germano, M., Piomelli U., Moin P., Cabot W.H.: 1991. A dynamic subgrid-scale eddy viscosity model. *Phys.of Fluids*, **3**, 1760.

[18] Lilly, D.K.: 1992. A proposed modification of the Germano subgrid-scale closure method. *Phys. of Fluids A*, **4**, 633.

[19] Meneveau, C., Katz, J.: 2000. Scale-invariance and turbulence models for large-eddy simulation. *Annu. Rev. Fluid Mech.*, **32**, 1.

[20] Geurts, B.J., Holm, D.D.: 2002. Alpha-modeling strategy for LES of turbulent mixing. *Turbulent Flow Computation*. Eds: D. Drikakis, B.J. Geurts. Fluid mechanics and its applications 66, Kluwer Academic Publishers, 237.

[21] Geurts, B.J., Holm, D.D.: 2003. Regularization modeling for large-eddy simulation. *Phys. of Fluids*, **15**, L13.

[22] Leray, J.: 1934. Sur les movements d'un fluide visqueux remplaissant l'espace. *Acta Mathematica*, **63**, 193.

[23] Foias, C., Holm, D.D., Titi, E.S.: 2001. The Navier-Stokes-alpha model of fluid turbulence. *Physica D*, **152**, 505.

[24] Smagorinsky, J.: 1963. General circulation experiments with the primitive equations. *Mon. Weather Rev.*, **91**, 99.

[25] Holm, D.D.: 1999. Fluctuation effects on 3D Lagrangian mean and Eulerian mean fluid motion. *Physica D*, **133**, 215.

[26] Holm, D.D., Marsden, J.E., Ratiu, T.S.: 1998. Euler-Poincaré models of ideal fluids with nonlinear dispersion. *Phys. Rev. Lett.*, **80**, 4173.

[27] Verstappen, R.W.C.P., Veldman, A.E.P.: 2002. Preserving symmetry in convection-diffusion schemes. *Turbulent flow computation*, Eds: D. Drikakis, B.J. Geurts. Kluwer Academic Publishers, 75.

[28] Kuerten, J.G.M.: 1999. Introduction to finite volume methods. Lecture notes JMBC course Computational Fluid Dynamics III, University of Twente.

[29] Buuren, R. van: 1999. Time integration methods for compressible flow. *Ph.D. Thesis*, Twente University Press.

[30] Streng, M., Gils, S. van, Meer, A. van der: 1994. *Maple, wiskunde in berekenbaar perspectief*. Addison-Wesley.

[31] Vreman, A.W.: 1995. Direct and large-eddy simulation of the compressible turbulent mixing layer. *Ph.D. Thesis*, University of Twente.

[32] IJzerman, W.L.: 2000. Signal representation and modeling of spatial structures in fluids. *Ph.D. Thesis*, University of Twente.

[33] Wasistho, B.: 1997. Spatial direct numerical simulation of compressible boundary layer flow. *Ph.D. Thesis*, University of Twente.

[34] Geurts, B.J.: 1999. Balancing errors in LES. *Proceedings: Direct and Large-Eddy simulation III: Cambridge*. Eds: N.D. Sandham, P.R. Voke, L. Kleiser. Kluwer Academic Publishers, 1.

[35] Geurts, B.J.: 2006. Interacting errors in large-eddy simulation: a review of recent developments. J. Turbulence, N55

[36] Van der Bos, F., Geurts, B.J.: 2010. Computational error-analysis of a discontinuous Galerkin discretization applied to large-eddy simulation of homogeneous turbulence, Computer methods in applied mechanics and engineering 199 (13-16), 903-915

[37] Vreman, A.W., Geurts, B.J., Kuerten, J.G.M.: 1994. Discretization error dominance over subgrid-terms in large eddy simulations of compressible shear layers. *Comm. Num. Meth. Eng. Math.*, **10**, 785.

[38] Vreman, A.W., Geurts, B.J., Kuerten, J.G.M.: 1996. Comparison of numerical schemes in large eddy simulation of the temporal mixing layer. *Int.J.Num.Meth.Fl.*, **22**, 297.

[39] Kwak, D., Reynolds, W.C., Ferziger, J.H.: 1975. Three-dimensional time dependent computation of turbulent flow. *Report TF-5*, Stanford.

[40] Love, M.D.: 1980. Subgrid modelling studies with Burgers' equation. *J. Fluid Mech.*, **100**, 87.

[41] Meneveau, C.: 1994. Statistics of turbulence subgrid-scale stresses: necessary conditions and experimental tests. *Phys. of Fluids*, **6**, 815.

[42] Geurts, B.J., Fröhlich, J.: 2002. A framework for predicting accuracy limitations in large-eddy simulation. *Phys. of fluids*, **14**, L41.

[43] Meyers, J., Baelmans, M., Geurts, B.J.: 2002. Accuracy limitations in LES employing eddy-viscosity subgrid-parameterization. *Proceedings European turbulence conference IX*, Eds: I.P. Castro, P.E. Hancock, T.G. Thomas. CIMNE, 858.

[44] Meyers, J., Geurts, B.J., Baelmans, M., 2003, Database analysis of errors in large-eddy simulation, Physics of Fluids 15 (9), 2740-2755

[45] Geurts, B.J., Fröhlich, J.: 2001. Numerical effects contaminating LES; a mixed story. *Modern simulation strategies for turbulent flow*, Ed: B.J. Geurts. R.T. Edwards Inc., Philadelphia, 317.

[46] Lorenz, E.N.: 1963. Deterministic nonperiodic flow. *J. Atmosph. Sc.*, **20**, 130.