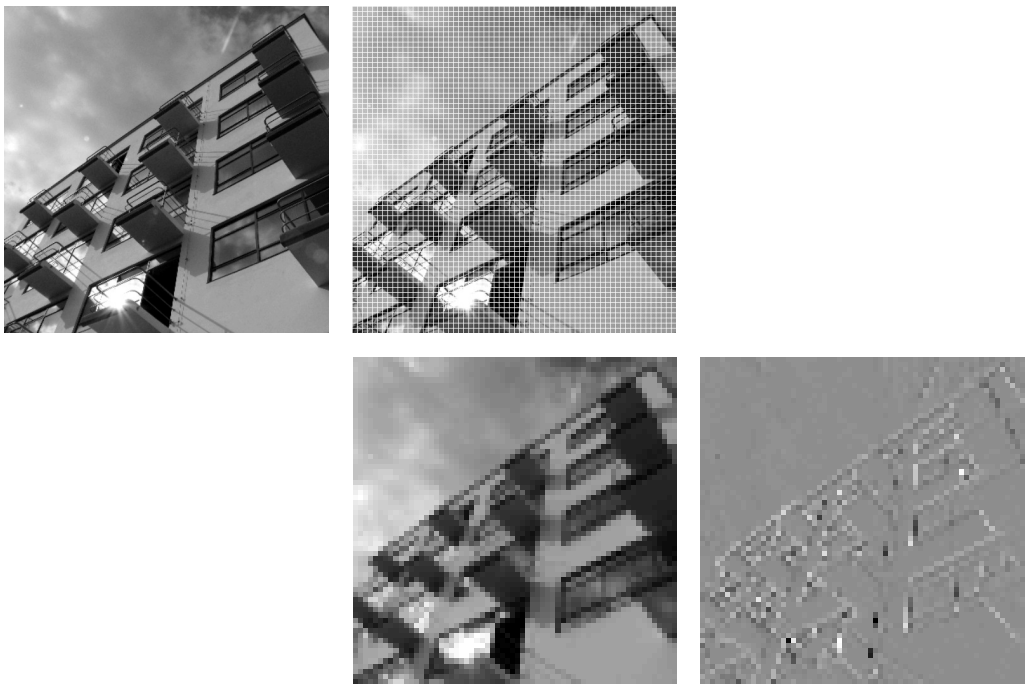
university of groningen

faculty of mathematics and natural sciences

A wavelet-based grid-switching method for energy cascades in Burgers' equation



Master's thesis in Applied Mathematics

April 2014

Student: Erik Mulder

First supervisor: Dr. ir. F. W. Wubs

Second supervisor: Prof. dr. H. Waalkens

Institute of Mathematics and Computing Science

Contents

1 Introduction **3**

2 Preliminaries **3**
2.1 The finite volume method . . . 4
2.1.1 Matrix-vector form . . . 5
2.2 The Haar wavelet transform matrix . . . 5

3 Derivation of the wavelet elimination method **6**
3.1 Partitioning . . . 6
3.1.1 Conservation of momentum on a coarse sub-grid . . . 7
3.1.2 A naive grid-switching approach . . . 8
3.2 Reordering . . . 8
3.2.1 Reordered discretization . . . 9
3.3 Elimination . . . 10
3.3.1 Obtaining a Jacobian . . . 11
3.3.2 Linearized discretization . . . 11
3.3.3 Approximating the inverse . . . 12
3.4 Returning to the fine grid . . . 13
3.5 Recursive extension of WEM . . . 14

4 Analysis of the wavelet elimination method **16**
4.1 The linearized FVM operator . . . 16
4.2 Analysis of the linearized WEM operator . . . 17
4.3 Time integration . . . 21
4.4 RK4 and the WEM scheme . . . 23
4.4.1 Energy decay . . . 25

5 Test problems **26**
5.1 Preliminary spectral code results . . . 26
5.1.1 Initial Brownian motion and higher order diffusion . . . 28
5.1.2 Filtering . . . 29
5.2 Grid-switching schemes . . . 30

6 Numerical results **31**
6.1 Recursive grid-switching results . . . 32
6.2 Initial Brownian motion . . . 32
6.2.1 Convergence aspects . . . 33

7 Concluding remarks **36**

A The Fourier-Galerkin spectral method **37**

B The Haar wavelet and image compression **39**

C An efficient implementation of the bilinear form $A\langle B\mathbf{x}, C\mathbf{y}\rangle$ **40**

1 Introduction

A wide variety of natural phenomena contain sophisticated structures at various levels of detail. Take, for instance, the motion of a turbulent fluid, in which large whirls interact with a seemingly infinite cascade of smaller structures. An interesting feature of such turbulent phenomena is the interplay between motions at different scales. Large moving structures transfer their motion to smaller ones, which break up into even smaller structures. Moreover, small environmental variations may have a significant effect on the motion's general structure, much like a chaotic dynamical system. In fact, from a dynamical systems point of view, turbulence has been described as the presence of a strange attractor [20].

A complete mathematical description of the motion of fluids is given by the Navier-Stokes equations, for which analytical solutions are only available in a few, classical situations. In every other case an accurate approximation is required. Numerical approximations of the Navier-Stokes equations attempt to capture the flow, with all its complexity, in a computational 'net', or *grid*. It follows that, when approximating turbulent motions, such a grid should be sufficiently fine. Unfortunately, with finer grids, computing times may drastically increase.

The general objective of this research is to alternate between fine and coarse grids, while approximating solutions at discrete points in time during a so called *time-stepping* procedure. Similar ideas exist in the solution of linear systems, where they are known as *multigrid methods* [26]. An advantage this alternation may have, is that the coarse calculations are generally cheaper, i.e. a smaller number of operations is involved. A major difficulty, however, concerns the loss of details when restricting a numerical solution to a coarse subset of the computational grid. The focus of this research will therefore lie with either storing or modeling the lost details, enabling their recovery when returning to a fine grid. Several different approaches will be discussed in the upcoming sections.

Instead of using Navier-Stokes, we will apply our methods to the one-dimensional Burgers' equation [2], a less complete model of fluid motion but sharing a lot of key-properties with the Navier-Stokes equations, most importantly its non-linear convective term. It is this non-linearity that gives a cascade-like energy transfer from large to small scales, requiring a numerical method to capture a wide range of details. A method that simply ignores the smallest variations is destined to fail when a solution of Burgers' equation is sufficiently complex, which makes it a decent testing ground. More specifically, Burgers' equation gives shock-like solutions when supplied with the appropriate initial conditions. Approximating these solutions will test the grid-switching ideas, since the smallest variations become of great importance in regions where steep transitions occur.

A convenient approach to viewing a solution with steep gradients is in terms of its averages and variations. In this study we will make extensive use of the Haar wavelet [13], to express a solution in terms of these averages and details. The great advantage of this, is that we can locally dissect a solution, preserving the flexibility of the underlying numerical method. A wavelet transform is a popular tool in decomposing any kind of information and in Appendix B we demonstrate its application in image compression.

In Section 2 we will start with explaining a few concepts required to understand the derivation of our main grid-switching method, which will be presented in Section 3. In Section 4 we will investigate and proof several numerical properties of the newly acquired method. Section 5 will contain an overview of the test problems and preliminary results using a spectral code, including an attempt to create an actual turbulent velocity profile in the one-dimensional Burgers' equation. Finally, in Section 6, we will present several numerical experiments using various grid-switching methods.

2 Preliminaries

Burgers' equation is a partial differential equation (PDE). A popular technique for numerically solving partial differential equations is the method of lines (MOL) (see [14], Section I.6). In this approach the spatial derivatives are discretized, giving a system of ordinary differential equations (ODEs), a so-called *semi-discretization*. Then this system is integrated in time using an ODE solver.

In space we will use a finite volume method (FVM) and for the time-integration we use the classic four-stage Runge-Kutta method (RK4). First we will pay some attention to the spatial discretization. We will postpone our discussion of the time-integration method until Section 4.3.

2.1 The finite volume method

Let $u = u(x,t) : [0,L] \times (0,T] \to \mathbb{R}$, where $[0,L]$ and $(0,T]$ denote the spatial and temporal domain respectively. Burgers' equation is usually given in the following form:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2}, \tag{2.1}$$

with μ the diffusion coefficient. As initial condition we take $u(x,0) = u_0(x)$. Since we do not aim to model any influences given by the boundaries of the domain, we should take periodic boundaries. That is, let $u(x+L,t) = u(x,t)$. Periodic boundaries are especially convenient for the modeling of turbulence, see [8].

Another representation of (2.1) is in the form of a conservation law[1]:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}F(u), \quad \text{with} \quad F(u) = -\frac{1}{2}u^2 + \mu\frac{\partial u}{\partial x}. \tag{2.2}$$

Here $F(\cdot)$ is a flux function describing the transport of momentum[2] due to convection and diffusion. To obtain a discretization we first define a grid with constant mesh-width h:

$$x_j = (j-1)h, \quad h = \frac{L}{N-1}, \quad j = 1,2,\ldots,N. \tag{2.3}$$

With N we will denote the total number of grid-points. A solution value at a grid-point is denoted by $u_j := u(x_j,t)$. Around each grid-point we can imagine a so called *control volume*, having vertices centered halfway between the neighboring grid-points. Taking the integral over this control volume gives a weak formulation of (2.2), where we notice that we actually have a divergence in the right hand side:

$$\int_V \frac{\partial u}{\partial t}dV = \int_V \frac{\partial}{\partial x}F(u)dV = \int_V (\nabla \cdot F(u))dV.$$

Here V denotes the volume. Using the divergence theorem we obtain

$$\int_V \frac{\partial u}{\partial t}dV = \int_S (F(u)\cdot \mathbf{n})dS,$$

where S is the surface of the control volume and $\mathbf{n}$ the outward normal vector. Resolving the integrals gives us a *finite volume discretization* (see for instance [14], section III.4 or [16], Section 4.7). Since we only consider one spatial dimension, the numerical solution on the j-th grid-point will satisfy

$$h\frac{du_j}{dt} = F(u_{j+1/2}) - F(u_{j-1/2}). \tag{2.4}$$

In other words, the change of momentum is given by the flux-difference over the control volume. Now it is necessary to approximate the flux function at values halfway between the grid-points. To this end we let $u_{j+1/2} \approx (u_j + u_{j+1})/2$. Furthermore, the spatial derivative will be approximated by a central difference, that is

$$\left.\frac{\partial u}{\partial x}\right|_{j+1/2} \approx \frac{u_{j+1} - u_j}{h}.$$

[1] In the literature one will usually encounter the form $u_t + F(u)_x = 0$.

[2] We can view (2.1) as an equation for momentum, where the density of a fluid parcel moving with the fluid is taken constant (incompressibility).

The resulting discretized flux function for the $(j+1/2)$-th grid-point is given by

$$F(u_{j+1/2}) = -\frac{1}{2}\left(\frac{u_j + u_{j+1}}{2}\right)^2 + \mu\frac{u_{j+1} - u_j}{h}. \tag{2.5}$$

Let $v_j := du_j/dt$. Then, substituting (2.5) in (2.4) gives the following semi-discretization:

$$hv_j = -\frac{1}{2}\left[\left(\frac{u_j + u_{j+1}}{2}\right)^2 - \left(\frac{u_j + u_{j-1}}{2}\right)^2\right] + \frac{\mu}{h}\left[u_{j+1} - 2u_j + u_{j-1}\right], \quad j = 1, 2, \ldots, N. \tag{2.6}$$

Thus, we are left with a system of N ordinary differential equations. To close the dependencies we need to include the periodic boundary conditions. For the first equation, corresponding to grid point $j = 1$, we let $j - 1 = 0 = N$. Similarly, for the final equation where $j = N$, we let $j + 1 = N + 1 = 1$.

2.1.1 Matrix-vector form

It is convenient to write the system of ODEs (2.6) in matrix-vector form. Let M and A be matrices defined by

$$M = \frac{1}{2}\begin{bmatrix} 1 & 1 & & & \\ & 1 & 1 & & \\ & & \ddots & \ddots & \\ & & & 1 & 1 \\ 1 & & & & 1 \end{bmatrix}, \qquad A = \frac{\mu}{h}\begin{bmatrix} -2 & 1 & & & 1 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 1 & & & 1 & -2 \end{bmatrix}. \tag{2.7}$$

The relation $w_j = (u_j + u_{j+1})/2$ for $j = 1, 2, \ldots, N$ with periodic boundaries is equivalent to the matrix-vector representation $\mathbf{w} = M\mathbf{u}$. It follows that the relation $w_j = (u_j + u_{j-1})/2$ is equivalent to $\mathbf{w} = M^T\mathbf{u}$. Moreover, the relation $w_j = \mu/h\left[u_{j+1} - 2u_j + u_{j-1}\right]$ can be written as $\mathbf{w} = A\mathbf{u}$. Now, denoting the pointwise product as[3] $\mathbf{uv} = \langle \mathbf{u}, \mathbf{v}\rangle$, the finite volume discretization (2.6) in matrix-vector form can be written as

$$h\mathbf{v} = -\frac{1}{2}\left[\langle M\mathbf{u}, M\mathbf{u}\rangle - \left\langle M^T\mathbf{u}, M^T\mathbf{u}\right\rangle\right] + A\mathbf{u}. \tag{2.8}$$

2.2 The Haar wavelet transform matrix

The solution vector $\mathbf{u}$ can be expressed as a linear combination of its average and increasingly detailed variations. We will perform such a decomposition within partitions of $\mathbf{u}$. The basis functions in which a partition will be expressed are given by the Haar wavelet family [13], which combined and orthogonalized form the orthogonal Haar transform matrix.

For instance, consider the following 2×2 Haar transform:

$$H_{2\times 2} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{2.9}$$

The first and second row will capture an average and a difference respectively. Hence, using this matrix we can decompose partitions of $\mathbf{u}$ into *averages* and *details*. Higher dimensional orthogonal Haar transform matrices can be constructed recursively from $H_{2\times 2}$ using:

$$H_{2p\times 2p} = \frac{1}{\sqrt{2}}\begin{bmatrix} H_{p\times p} \otimes [1, 1] \\ I_{p\times p} \otimes [1, -1] \end{bmatrix}, \quad p = 2, 4, 8, \ldots \tag{2.10}$$

Here $\otimes$ denotes the Kronecker product and $I_{p\times p}$ the $p \times p$ identity matrix. From (2.10) it follows that a partition to which we can apply such a transform, is restricted to having dimension 2^k, for $k \in \mathbb{N}$. In the upcoming sections we heavily rely on the local decompositions given by the transformation (2.10). Its most popular application, however, lies in the field of image compression, see Appendix B.

[3] The product $\langle \mathbf{u}, \mathbf{v}\rangle$ is a bilinear form, i.e. it is linear in both its arguments.

3 Derivation of the wavelet elimination method

The main objective in this section is to obtain a *grid-switching* technique, switching back and forth between fine and coarse grids. A major obstacle will be the loss of information on the smallest scales and preventing this will be of particular concern. At the end of Section 3.3.3 we obtain a method that, in theory, should be able to maintain a significant portion of the information contained on a fine grid, while time-stepping on a coarse grid. To achieve this, first a (short) time-integration on the fine grid is assumed. The information obtained on the fine grid will then serve as initial guess for several linearizations. As an alternative, in Section 3.1.2 we will also give a more naive method, still capable of switching from a fine to a coarse grid and maintaining information from the fine grid.

First we need to rephrase our semi-discretization. We have seen that the FVM discretization of Burgers' equation on an equidistant[4] grid of size N gives rise to the following system of ODEs:

$$h\mathbf{v} = -\frac{1}{2}\left[\langle M\mathbf{u}, M\mathbf{u}\rangle - \left\langle M^T\mathbf{u}, M^T\mathbf{u}\right\rangle\right] + A\mathbf{u}, \tag{3.1}$$

where $M, A \in \mathbb{R}^{N\times N}$ and $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$. The bilinear form $\langle\cdot,\cdot\rangle$ is symmetric[5], which enables us to write the discretization as

$$h\mathbf{v} = -\frac{1}{2}\left\langle (M+M^T)\mathbf{u}, (M-M^T)\mathbf{u}\right\rangle + A\mathbf{u} \tag{3.2}$$

$$= -\frac{1}{2}\mathcal{N}(\mathbf{u}) + A\mathbf{u}, \tag{3.3}$$

where we let $\mathcal{N}(\mathbf{u})$ denote the nonlinear part of the system. In the next section we explain how to partition the solution vector $\mathbf{u}$, using the orthogonal Haar transformation.

3.1 Partitioning

Let $Q \in \mathbb{R}^{N\times N}$, be a block diagonal matrix with N/p orthogonal Haar transformation blocks $H \in \mathbb{R}^{p\times p}$:

$$Q = \begin{bmatrix} H & & & \\ & H & & \\ & & \ddots & \\ & & & H \end{bmatrix}, \quad Q^{-1} = Q^T. \tag{3.4}$$

This matrix transforms partitions of length p within a vector of size N, where p should be a divisor of N. As we have seen in Section 2.2, p should also be power of 2. From here on we will call p our *partition size*.

Let $\phi = Q\mathbf{u}$ and $\psi = Q\mathbf{v}$, then ϕ and ψ are transformed solution vectors containing averages and details of N/p partitions, in particular:

$$\phi = Q\mathbf{u} = \begin{bmatrix} H & & & \\ & H & & \\ & & \ddots & \\ & & & H \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_{N/p} \end{bmatrix} = \begin{bmatrix} H\mathbf{u}_1 \\ H\mathbf{u}_2 \\ \vdots \\ H\mathbf{u}_{N/p} \end{bmatrix}, \tag{3.5}$$

where $\mathbf{u}_k = [u_{(k-1)p+1}, \ldots, u_{kp}]^T$ denotes the k-th partition of the solution vector $\mathbf{u}$ and $k = 1, \ldots, N/p$. The FVM discretization (3.3) is transformed as follows:

$$h\psi = -\frac{1}{2}Q\mathcal{N}\left(Q^T\phi\right) + QAQ^T\phi. \tag{3.6}$$

[4] A grid with a constant mesh-width h.

[5] $\langle\mathbf{u},\mathbf{v}\rangle = \langle\mathbf{v},\mathbf{u}\rangle$.

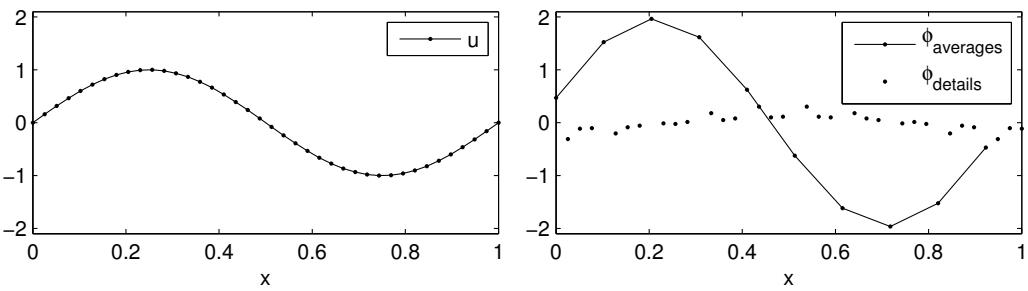
Example 3.1. Suppose $\mathbf{u} \in \mathbb{R}^N$ is a discretized sine wave on the domain $x \in [0,1]$ with grid size $N = 40$. The elements of $\mathbf{u}$ are given by:

$$u_i = \sin(2\pi x_i), \quad x_i = \frac{i-1}{N-1}, \quad i = 1,2,\ldots,N.$$

The original solution $\mathbf{u}$ is plotted in Figure 3.1(a). Let $p = 4$ be the partition size. Using Equations (2.9) and (2.10), we obtain the 4×4 orthogonal Haar transform matrix:

$$H = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}.$$

The result of the orthogonal transformation, $\phi = Q\mathbf{u}$, is shown in Figure 3.1(b). Here we make the



(a) (b)

Figure 3.1: Before (a) and after (b) a partitioned Haar transformation: $\phi = Q\mathbf{u}$, with partition size $p = 4$.

distinction between elements of ϕ containing averages, and elements containing details. Specifically, if we denote the averages as ϕ_a and the details as ϕ_d we have that

$$\phi_a = \bigcup_k \{\phi_{l+1}\}, \quad \phi_d = \bigcup_k \{\phi_{l+2}, \ldots, \phi_{l+p}\}, \text{ with } l = (k-1)p \text{ and } k = 1,2,\ldots,N/p. \tag{3.7}$$

Note that the averages in ϕ_a differ with a factor $\sqrt{p} = 2$ from the actual averages of the partitions in $\mathbf{u}$. Nevertheless, we will still call the values in ϕ_a averages. ▲

3.1.1 Conservation of momentum on a coarse sub-grid

After the partitioning using the Haar wavelet, the solution is decomposed in a part containing averages and a part containing details (see Figure 3.2). For a coarse sub-grid it is natural to select the grid-points containing averages, seeing that conservation of momentum is easily shown for these grid-points. Let $\phi = Q\mathbf{u}$ and $\psi = Q\mathbf{v}$ with $\mathbf{v} = \frac{d}{dt}\mathbf{u}$. Recall that for a finite volume discretization we have (cf. (2.4))

$$hv_j = F(u_{j+1/2}) - F(u_{j-1/2}), \quad j = 1,2,\ldots,N.$$

Using the notation of Equation (3.7) we let an average velocity be located at ψ_{l+1}, with $l = (k-1)p$ and $k = 1,2,\ldots,N/p$. From the Haar transform we know that ψ_{l+1} is given by $\psi_{l+1} = \frac{1}{\sqrt{p}}\sum_{i=1}^{p} v_{l+i}$. Consequently, we have that

$$\begin{aligned} hp\psi_{l+1} &= \sqrt{p}\sum_{i=1}^{p} hv_{l+i} \\ &= \sqrt{p}\left[F(u_{l+3/2}) - F(u_{l+1/2}) + F(u_{l+5/2}) - F(u_{l+3/2}) + \ldots + F(u_{l+p+1/2}) - F(u_{l+p-1/2})\right] \\ &= \sqrt{p}\left[F(u_{l+p+1/2}) - F(u_{l+1/2})\right]. \end{aligned} \tag{3.8}$$

Since the inner flux-values cancel we are left with the flux difference over a large cell with cell-width hp. If we take all partitions into account and recall that we have periodic boundaries, we find that all flux contributions cancel. This implies the conservation of momentum ϕ and (3.8) is clearly a discretized conservation law.

3.1.2 A naive grid-switching approach

Now that we have a coarse sub-grid, we can choose to create a new finite volume discretization with the averages obtained by the wavelet partitioning, while storing the details for later use. This will be our naive grid-switching approach (NGS). With storing the details we implicitly assume that the smallest scales will be time-invariant.

Let an initial solution on the fine grid be available. With the wavelet partitioning we obtain averages and details contained in $\phi = Q\mathbf{u}$. A coarse FVM discretization on the averages is given by:

$$\begin{aligned} hp\psi_k = &-\frac{1}{2}\left[\left(\frac{\phi_{(k-1)p+1}+\phi_{kp+1}}{2}\right)^2 - \left(\frac{\phi_{(k-1)p+1}+\phi_{(k-2)p+1}}{2}\right)^2\right] \\ &+\frac{\mu}{hp}\left[\phi_{kp+1} - 2\phi_{(k-1)p+1} + \phi_{(k-2)p+1}\right], \quad k = 1,2,\ldots,N/p. \end{aligned} \tag{3.9}$$

The boundaries are taken periodic in the index k, which, for instance, leads to using the average of the N/p-th partition when the grid-point $-p+1$ is required. After a given number of time-steps we return to the fine grid. Here the newly obtained averages will be combined with the stored details. The scheme is summarized in the pseudo-code of Algorithm 3.1.

input: Initial discrete solution: $\mathbf{u}(\mathbf{x}, t_0)$, partition size: p, number of time-steps: s.
output: Solution $\mathbf{u}(\mathbf{x}, t)$ at time $t = t_0 + s\Delta t$.

1: Construct Q, based on the partition size p and dimension N of $\mathbf{u}$.
2: $\phi := Q\mathbf{u}$, obtain partitioned solution.
3: $i_a := \{1, p+1, 2p+1, \ldots, (N/p-1)p+1\}$, create indexing of the averages.
4: $i_d := \{1, 2, \ldots, N\} \setminus \{i_a\}$, indexing of the details that will remain unchanged.
5: $\phi_a := \phi_{\{i_a\}}/\sqrt{p}$, store and scale the averages.
. . . . . *Move to the coarse grid.*
6: **for** $j = 1, 2, \ldots, s$ **do**
7: Perform an explicit time-step (RK4) on ϕ_a with a coarse FVM discretization.
8: **end for**
. . . . . *Return to the fine grid.*
9: $\phi_{\{i_a\}} := \sqrt{p}\phi_a$, update the coarse grid-points with unscaled averages.
10: $\mathbf{u} := Q^T\phi$, return to a complete solution using new averages and old details.

Algorithm 3.1: Naive grid-switching procedure (NGS).

Remark. Note that in line 5 of Algorithm 3.1 we need to scale the solution to obtain true averages for the coarse time-stepping in line 7, but at the transformation in line 10 we require the solution to be in its unscaled form.

3.2 Reordering

Algorithm 3.1 retrieves the complete solution in a somewhat crude way. We would like to return to the fine grid using an additional model for the details, instead of using the exact entries we leave

behind when moving to the coarse grid. The first step in creating such a model is separating the details from the averages, to obtain expressions for their respective velocities. To this end we will perform a reordering of the vector ϕ.

The desired reordering gives a decomposition of the form $\widetilde{\phi} = [\phi_d, \phi_a]^T$. The vectors ϕ and $\widetilde{\phi}$ are related through a row-permutation: $\widetilde{\phi} = P_\pi \phi$, where π denotes a permutation on the indices of ϕ:

$$\pi : \{1, \ldots, N\} \to \{1, \ldots, N\},$$

given by

$$\pi(j) = \begin{cases} \frac{N(p-1)}{p} + \frac{j-1}{p} + 1 & \text{if } (j-1) \bmod p = 0 \\ \left(\frac{j-q}{p}\right)(p-1) + (j-1) \bmod p & \text{if } (j-q) \bmod p = 0 \\ & \text{for some } q \in \{2, 3, \ldots, p\}. \end{cases} \tag{3.10}$$

The permutation matrix is given by $P_\pi = [\mathbf{e}_{\pi(1)}, \mathbf{e}_{\pi(2)}, \ldots, \mathbf{e}_{\pi(N)}]$, where $\mathbf{e}_i \in \mathbb{R}^N$ denotes the i-th unit column vector. Since this is just the identity matrix with its rows reordered, we have that $P_\pi^{-1} = P_\pi^T$. Another representation of this permutation matrix is through a vector $\mathbf{q}$, with elements q_j given by the row index of a 1 occurring in the j-th column of P_π (see Section 3.4 of [11]). This would give $\mathbf{q} = (\pi(1), \pi(2), \ldots, \pi(N))$. In terms of this vector $\mathbf{q}$, the reordering $\widetilde{\phi} = P_\pi \phi$ is given by $\widetilde{\phi}(q_j) = \phi(j)$ for $j = 1, 2, \ldots, N$.

Example 3.2. Consider the situation in Example 3.1 where, after the wavelet transformation, we end up with a vector $\phi \in \mathbb{R}^{N=40}$, with partitions of size $p = 4$. The averages are located at indices $j = (k-1)p + 1$, the details at $j = \{(k-1)p + 2, \ldots, kp\}$, with $k = 1, 2, \ldots, N/p$, see Figure 3.2.

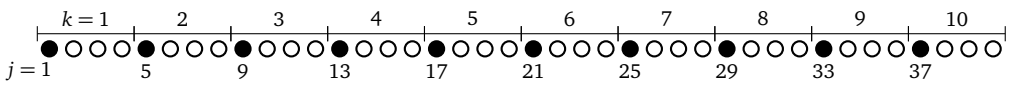


Figure 3.2: Ordering in the transformed vector ϕ of Example 3.1. The averages are marked black and the details white. On top the partitions and partition numbers k are shown, on bottom the actual indices of ϕ.

The permutation given by (3.10) reorders the vector ϕ in the desired form $\widetilde{\phi} = [\phi_d, \phi_a]^T$. For instance, $\phi_{j=9}$ contains an average and satisfies $(j-1) \bmod p = 0$. It is therefore mapped to $j_{\text{new}} = \frac{N(p-1)}{p} + \frac{j-1}{p} + 1 = 40\left(\frac{3}{4}\right) + \frac{8}{4} + 1 = 33$. On the other hand, $\phi_{j=10}$ contains a detail and satisfies $(j-q) \bmod p = 0$ for $q = 2$, hence it gets mapped to $j_{\text{new}} = \left(\frac{j-q}{p}\right)(p-1) + (j-1) \bmod p = \left(\frac{8}{4}\right)(3) + 1 = 7$. The result of a complete reordering is shown in Figure 3.3.

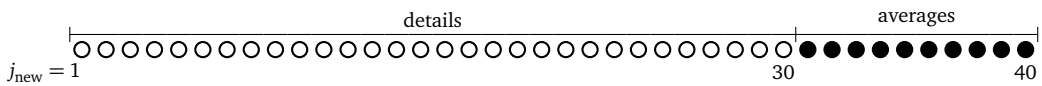


Figure 3.3: Reordering given by $\widetilde{\phi} = [\phi_d, \phi_a]^T = P_\pi \phi$. The averages are marked black and the details white.

▲

3.2.1 Reordered discretization

With a reordering we are able to separate the system of ODEs (3.6) and distinguish the equations governing the average velocities from the equations governing the detailed velocities. Rewriting (3.6) in terms of the permuted vectors $\widetilde{\phi} = P_\pi \phi$ and $\widetilde{\psi} = P_\pi \psi$ gives

$$h\widetilde{\psi} = -\frac{1}{2} P_\pi Q \mathcal{N}\left(Q^T P_\pi^T \widetilde{\phi}\right) + P_\pi Q A Q^T P_\pi^T \widetilde{\phi}. \tag{3.11}$$

Or, defining $R := P_\pi Q$ and $G := RAR^T$:

$$h\widetilde{\psi} = -\frac{1}{2}R\mathcal{N}\left(R^T\widetilde{\phi}\right) + G\widetilde{\phi} \quad \Leftrightarrow$$

$$h\begin{bmatrix}\psi_d\\ \psi_a\end{bmatrix} = -\frac{1}{2}\begin{bmatrix}R_{11} & R_{12}\\ R_{21} & R_{22}\end{bmatrix}\mathcal{N}\left(R^T[\phi_d,\phi_a]^T\right) + \begin{bmatrix}G_{11} & G_{12}\\ G_{21} & G_{22}\end{bmatrix}\begin{bmatrix}\phi_d\\ \phi_a\end{bmatrix}, \tag{3.12}$$

with $R, G \in \mathbb{R}^{N\times N}$. Now we can start separating the scheme. The nonlinear term in (3.12) is given by:

$$\mathcal{N}\left(R^T[\phi_d,\phi_a]^T\right) = \left\langle E[\phi_d,\phi_a]^T, F[\phi_d,\phi_a]^T\right\rangle,$$

with $E = (M + M^T)R^T$ and $F = (M - M^T)R^T$ (cf. (3.2)). We obtain separate nonlinear terms for the details and averages by writing

$$\mathcal{N}\left(R^T[\phi_d,\phi_a]^T\right) = \begin{bmatrix}\mathcal{N}_d\left(\phi_d,\phi_a\right)\\ \mathcal{N}_a\left(\phi_d,\phi_a\right)\end{bmatrix}, \quad \text{with}$$

$$\mathcal{N}_d\left(\phi_d,\phi_a\right) = \left\langle E_{11}\phi_d + E_{12}\phi_a, F_{11}\phi_d + F_{12}\phi_a\right\rangle, \tag{3.13}$$

$$\mathcal{N}_a\left(\phi_d,\phi_a\right) = \left\langle E_{21}\phi_d + E_{22}\phi_a, F_{21}\phi_d + F_{22}\phi_a\right\rangle. \tag{3.14}$$

The complete separation of average and detailed velocities gives two equations:

$$h\psi_d = -\frac{1}{2}R_{11}\mathcal{N}_d(\phi_d,\phi_a) - \frac{1}{2}R_{12}\mathcal{N}_a(\phi_d,\phi_a) + G_{11}\phi_d + G_{12}\phi_a, \tag{3.15}$$

$$h\psi_a = -\frac{1}{2}R_{21}\mathcal{N}_d(\phi_d,\phi_a) - \frac{1}{2}R_{22}\mathcal{N}_a(\phi_d,\phi_a) + G_{21}\phi_d + G_{22}\phi_a. \tag{3.16}$$

Let $n_d = N(p-1)/p$ and $n_a = N/p$, then $N = n_d + n_a$ and the matrix blocks have dimensions $(\cdot)_{11} \in \mathbb{R}^{n_d\times n_d}$, $(\cdot)_{12} \in \mathbb{R}^{n_d\times n_a}$, $(\cdot)_{21} \in \mathbb{R}^{n_a\times n_d}$ and $(\cdot)_{22} \in \mathbb{R}^{n_a\times n_a}$.

3.3 Elimination

As a result of the partitioning and reordering we obtain Equations (3.15) and (3.16), for the time-derivatives of both the details and the averages, ψ_d and ψ_a. The best choice for a coarse sub-grid appears to be the locations of the averages. Now it is necessary to derive a time-stepping relation on this coarse grid, without fully neglecting the details on the old, fine grid. The actual time-stepping procedure will be defined in Section 4.3. For now we only need to know that we need a relation of the form

$$\psi_a = \mathcal{F}(\phi_a).$$

To achieve such a relation, we let $(\phi_d)_t = \psi_d = 0$ and assume ϕ_a is available. Then (3.15) can be solved for ϕ_d and consequently, ϕ_d can be eliminated from (3.16). The result is an approximation of the complete discretization (3.15)-(3.16), based on the assumption that the details ϕ_d do not change over time, i.e. ϕ_d is *stationary*. A similar approach can be found in [15].

The assumption that the details are stationary gives a model that can be used for calculations on the coarse sub-grid. Instead of explicitly calculating their values, the effect of the details are made implicitly available due to their elimination. In the rest of this section the specifics of this elimination will be explained.

The coarse vector ϕ_a will be subject to a time-stepping procedure. This implies that for every new calculation, a previous ϕ_a will be known. This previous value will be denoted as $\widehat{\phi}_a$. Rewriting (3.15)-(3.16) and substituting $\psi_d = 0$ gives

$$\frac{1}{2}R_{11}\mathcal{N}_d(\phi_d,\widehat{\phi}_a) + \frac{1}{2}R_{12}\mathcal{N}_a(\phi_d,\widehat{\phi}_a) - G_{11}\phi_d - G_{12}\widehat{\phi}_a = 0, \tag{3.17}$$

$$h\psi_a + \frac{1}{2}R_{21}\mathcal{N}_d(\phi_d,\widehat{\phi}_a) + \frac{1}{2}R_{22}\mathcal{N}_a(\phi_d,\widehat{\phi}_a) - G_{21}\phi_d - G_{22}\widehat{\phi}_a = 0. \tag{3.18}$$

Equations (3.17)-(3.18) allow for the elimination of stationary details in a locally wavelet transformed finite volume discretization but, for simplicity, we will stick to calling it a wavelet elimination method (WEM).

The next step is to linearize the system (3.17)-(3.18) with respect to ϕ_d, such that we may eliminate the dependency on ϕ_d in (3.18). We need linearizations of the terms $\mathcal{N}_d(\phi_d, \widehat{\phi}_a)$ and $\mathcal{N}_a(\phi_d, \widehat{\phi}_a)$:

$$\mathcal{N}_d(\phi_d, \widehat{\phi}_a) \approx \mathcal{N}_d(\widehat{\phi}_d, \widehat{\phi}_a) + J_d(\widehat{\phi}_d, \widehat{\phi}_a)\Delta\phi_d, \tag{3.19}$$

$$\mathcal{N}_a(\phi_d, \widehat{\phi}_a) \approx \mathcal{N}_a(\widehat{\phi}_d, \widehat{\phi}_a) + J_a(\widehat{\phi}_d, \widehat{\phi}_a)\Delta\phi_d, \tag{3.20}$$

where J_d is the Jacobian of $\mathcal{N}_d$ in the point $(\widehat{\phi}_d, \widehat{\phi}_a)$, $\Delta\phi_d = \phi_d - \widehat{\phi}_d$ is an increment and $\widehat{\phi}_d$ is an initial known value of ϕ_d. In fact, $\widehat{\phi}_d$ will correspond the last known value of the details on the fine grid, before switching to the coarse grid. In the next section we take a short detour into producing the Jacobians necessary for the linearization of the discretization.

3.3.1 Obtaining a Jacobian

In the implementation of WEM the Jacobians J_d and J_a are not required explicitly, only their *actions*, i.e. the result of the matrix vector product $J\mathbf{v}$ for some vector $\mathbf{v}$. This means we pursue a *Jacobian-free* implementation, see for instance [1] and [10]. The idea is to replace the product $J_{(\cdot)}(\widehat{\phi}_d, \widehat{\phi}_a)\mathbf{v}$ with a difference quotient:

$$J_{(\cdot)}(\widehat{\phi}_d, \widehat{\phi}_a)\mathbf{v} \approx \frac{\mathcal{N}_{(\cdot)}(\widehat{\phi}_d + \varepsilon\mathbf{v}, \widehat{\phi}_a) - \mathcal{N}_{(\cdot)}(\widehat{\phi}_d, \widehat{\phi}_a)}{\varepsilon}. \tag{3.21}$$

Using a Taylor expansion of $\mathcal{N}_{(\cdot)}(\widehat{\phi}_d + \varepsilon\mathbf{v}, \widehat{\phi}_a)$ around $\widehat{\phi}_d$ it is easy to see that (3.21) holds:

$$\mathcal{N}_{(\cdot)}(\widehat{\phi}_d + \varepsilon\mathbf{v}, \widehat{\phi}_a) \approx \mathcal{N}_{(\cdot)}(\widehat{\phi}_d, \widehat{\phi}_a) + J_{(\cdot)}(\widehat{\phi}_d, \widehat{\phi}_a)\varepsilon\mathbf{v}.$$

Using (3.21) and (3.13)-(3.14) we find that

$$J_d(\widehat{\phi}_d, \widehat{\phi}_a)\mathbf{v} = \left\langle E_{11}\mathbf{v}, F_{11}\widehat{\phi}_d + F_{12}\widehat{\phi}_a \right\rangle + \left\langle E_{11}\widehat{\phi}_d + E_{12}\widehat{\phi}_a, F_{11}\mathbf{v} \right\rangle, \tag{3.22}$$

$$J_a(\widehat{\phi}_d, \widehat{\phi}_a)\mathbf{v} = \left\langle E_{21}\mathbf{v}, F_{21}\widehat{\phi}_d + F_{22}\widehat{\phi}_a \right\rangle + \left\langle E_{21}\widehat{\phi}_d + E_{22}\widehat{\phi}_a, F_{21}\mathbf{v} \right\rangle, \tag{3.23}$$

in which we recognize the product rule. Since $\mathbf{v}$ is arbitrary (3.22)-(3.23) also determine the Jacobians explicitly. Using the symmetry of the bilinear form we find:

$$J_d(\widehat{\phi}_d, \widehat{\phi}_a) = \operatorname{diag}\left(F_{11}\widehat{\phi}_d + F_{12}\widehat{\phi}_a\right) E_{11} + \operatorname{diag}\left(E_{11}\widehat{\phi}_d + E_{12}\widehat{\phi}_a\right) F_{11}, \tag{3.24}$$

$$J_a(\widehat{\phi}_d, \widehat{\phi}_a) = \operatorname{diag}\left(F_{21}\widehat{\phi}_d + F_{22}\widehat{\phi}_a\right) E_{21} + \operatorname{diag}\left(E_{21}\widehat{\phi}_d + E_{22}\widehat{\phi}_a\right) F_{21}. \tag{3.25}$$

3.3.2 Linearized discretization

In order to eliminate the details we need to obtain a linearized discretization. To avoid clutter, for the remaining part of the derivation we will define $\mathcal{N}_d := \mathcal{N}_d(\widehat{\phi}_d, \widehat{\phi}_a)$, $\mathcal{N}_a := \mathcal{N}_a(\widehat{\phi}_d, \widehat{\phi}_a)$, $J_d := J_d(\widehat{\phi}_d, \widehat{\phi}_a)$ and $J_a := J_a(\widehat{\phi}_d, \widehat{\phi}_a)$. Substituting the linearizations (3.19)-(3.20) in the system (3.17)-(3.18) and rewriting in terms of the increment $\Delta\phi_d$ gives

$$\frac{1}{2}\left(R_{11}J_d + R_{12}J_a - G_{11}\right)\Delta\phi_d = -\frac{1}{2}\left(R_{11}\mathcal{N}_d + R_{12}\mathcal{N}_a\right) + G_{11}\widehat{\phi}_d + G_{12}\widehat{\phi}_a, \tag{3.26}$$

$$h\psi_a + \frac{1}{2}\left(R_{21}J_d + R_{22}J_a - G_{21}\right)\Delta\phi_d = -\frac{1}{2}\left(R_{21}\mathcal{N}_d + R_{22}\mathcal{N}_a\right) + G_{21}\widehat{\phi}_d + G_{22}\widehat{\phi}_a. \tag{3.27}$$

Or, in the form of a linear system:

$$\begin{bmatrix} K_1 - G_{11} & 0 \\ K_2 - G_{21} & hI \end{bmatrix} \begin{bmatrix} \Delta\phi_d \\ \psi_a \end{bmatrix} = \begin{bmatrix} \mathbf{b}_d \\ \mathbf{b}_a \end{bmatrix}, \quad \text{with} \tag{3.28}$$

$$K_1 = R_{11}J_d + R_{12}J_a, \quad K_2 = R_{21}J_d + R_{22}J_a,$$
$$\mathbf{b}_d := -\frac{1}{2}\left(R_{11}\mathcal{N}_d + R_{12}\mathcal{N}_a\right) + G_{11}\widehat{\phi}_d + G_{12}\widehat{\phi}_a,$$
$$\mathbf{b}_a := -\frac{1}{2}\left(R_{21}\mathcal{N}_d + R_{22}\mathcal{N}_a\right) + G_{21}\widehat{\phi}_d + G_{22}\widehat{\phi}_a.$$

We are not going to solve (3.28) in the usual fashion, since we are only interested in a relation of the form $\psi_a = \mathcal{F}(\widehat{\phi}_a)$, which we can directly use in a time-stepping method. As explained in the beginning of this section, to achieve such a relation we need to eliminate the unknown $\Delta\phi_d$. From the first equation in (3.28) we have $\Delta\phi_d = (K_1 - G_{11})^{-1}\mathbf{b}_d$. Then, substituting this expression in the second equation of (3.28) gives

$$\psi_a = \frac{1}{h}\left[\mathbf{b}_a - (K_2 - G_{21})(K_1 - G_{11})^{-1}\mathbf{b}_d\right]. \tag{3.29}$$

Since $\mathbf{b}_d$, $\mathbf{b}_a$, K_1 and K_2 depend on $\widehat{\phi}_d$ and $\widehat{\phi}_a$, we have a relation of the form $\psi_a = \mathcal{F}(\widehat{\phi}_d, \widehat{\phi}_a)$. The vector $\widehat{\phi}_d$, containing initial details, will remain constant during the coarse time-stepping. We have therefore effectively achieved the desired relation: $\psi_a = \mathcal{F}(\widehat{\phi}_a)$.

Note that due to the elimination we are left with an inverse. Calculating this inverse directly is inconvenient. Thus, in the next section we will give a suitable approximation.

3.3.3 Approximating the inverse

Rewriting (3.29) gives

$$\psi_a = \frac{1}{h}\left[\mathbf{b}_a + (K_2 - G_{21})(I - G_{11}^{-1}K_1)^{-1}G_{11}^{-1}\mathbf{b}_d\right].$$

A popular approach to approximating $(I - G_{11}^{-1}K_1)^{-1}$ is taking a truncated Neumann expansion, see for instance [7] and [28]. Note that this expansion converges when the spectral radius $\rho(G_{11}^{-1}K_1) < 1$ ([23], Theorem 3.15), which we will assume from here on. The expansion is given by

$$(I - G_{11}^{-1}K_1)^{-1} = \sum_{i=0}^{\infty}(G_{11}^{-1}K_1)^i.$$

Taking only the first two terms of the series gives the following approximation of (3.29):

$$\psi_a \approx \frac{1}{h}\left[\mathbf{b}_a + (K_2 - G_{21})((I + G_{11}^{-1}K_1)G_{11}^{-1}\mathbf{b}_d)\right]. \tag{3.30}$$

In (3.30) there is still an inverse present and we will deal with it in the same way. Let $G_{11} = D - B$ be a splitting of G_{11}, where D contains the diagonal and $B = -(G_{11} - D)$. Again, using the first two terms in the series expansion we obtain the approximation

$$G_{11}^{-1} = (I - D^{-1}B)^{-1}D^{-1} \approx (I + D^{-1}B)D^{-1},$$

where the inverse of the diagonal matrix D is easily calculated. The final relation for time-stepping on the coarse grid is given by

$$\psi_a \approx \frac{1}{h}\left[\mathbf{b}_a + (K_2 - G_{21})((I + \widetilde{G}_{11}^{-1}K_1)\widetilde{G}_{11}^{-1}\mathbf{b}_d)\right], \quad \text{with} \quad \widetilde{G}_{11}^{-1} = (I + D^{-1}B)D^{-1}. \tag{3.31}$$

With this relation it is possible to perform coarse time-steps, while letting the fine variations contribute implicitly.

3.4 Returning to the fine grid

At last we are able to partition the grid, select the averages and perform elimination-based time-steps on a coarse grid. What remains is having to return to a complete solution on a fine grid. As explained in Section 3.1.2 and Algorithm 3.1 we could just take the details corresponding to the point at which we leave the fine grid. This will, however, lead to a solution that is *out of sync*. That is, the details that are introduced belong to an old solution and the combination of old details ($\widehat{\phi}_d$) and new averages ($\widehat{\phi}_a$) will inevitably affect the smoothness of the complete, fine solution.

To reduce this effect we can use one of the earlier obtained equations. Notice that Equation (3.26) is in the form of a Newton-Raphson root-finding algorithm:

$$J(\mathbf{x}_{\text{old}})\delta\mathbf{x} = -F(\mathbf{x}_{\text{old}}),$$

with $\delta\mathbf{x} = \mathbf{x}_{\text{new}} - \mathbf{x}_{\text{old}}$ and J the Jacobian of F (see for instance [18], Section 9.6). Let ϕ_d act as the unknown $\mathbf{x}$. Then a few Newton iterations with (3.26) will return details that will fit better with the newly obtained averages $\widehat{\phi}_a$. Note that, as initial guess, it seems natural to use the old details $\widehat{\phi}_d$, but in practice it turns out that using a zero vector $\mathbf{x}_0 = 0$ leads to better results. Furthermore, recall that for the elimination of $\Delta\phi_d$ in Section 3.3.2 we use an approximation of the inverse $(K_1 - G_{11})^{-1}$. The same approximation will be put to use in performing the Newton iteration.

A summary of the strategy for moving back and forth between fine and coarse grids is presented in Algorithm 3.2.

input: Initial discrete solution: $\mathbf{u}(\mathbf{x}, t_0)$, partition size: p, number of time-steps: s, tolerance for the Newton-Raphson iteration: `tol`.
output: Solution $\mathbf{u}(\mathbf{x}, t)$ at time $t = t_0 + s\Delta t$.

Construct Q and P_π, based on the partition size p and dimension N of $\mathbf{u}$.
$\phi := P_\pi Q\mathbf{u}$, obtain partitioned and reordered solution.
$\phi_d := \phi_{\{1,2,\ldots,N(p-1)/p\}}$, store the details.
$\phi_a := \phi_{\{N(p-1)/p+1,\ldots,N\}}$, store the averages.
. . . . . *Move to the coarse grid.*
for $j = 1, 2, \ldots, s$ **do**
    Perform an RK4 step on ϕ_a using the WEM discretization (3.31), letting $\widehat{\phi}_d = \phi_d$.
end for
. . . . . *Return to the fine grid.*
$\phi_d := 0$, set initial guess for the Newton-Raphson return iteration.
while $\|\Delta\phi_d\| >$ tol **do**
    Calculate $\Delta\phi_d$ using (3.26), letting $\widehat{\phi}_d = \phi_d$.
    $\phi_d := \Delta\phi_d + \phi_d$, calculate new ϕ_d.
end while
$\phi_{\{1,2,\ldots,N(p-1)/p\}} := \phi_d$, update the fine grid-points.
$\phi_{\{N(p-1)/p+1,\ldots,N\}} := \phi_a$, update the coarse grid-points.
$\mathbf{u} := Q^T P_\pi^T \phi$, return to a complete, fine solution.

Algorithm 3.2: WEM grid-switching procedure.

3.5 Recursive extension of WEM

A question that may arise is whether there exists a bound on the allowed size of the partitions and the coarse sub-grid. As long as it is a multiple of 2 and a divisor of N we can choose the partition size p as large as we like. However, when the number of details is large in comparison to the number of averages, the assumption that these details are stationary might become unrealistic. To overcome this problem, it could be worthwhile repeating the wavelet elimination on several nested levels, using a moderate partition size.

Instead of choosing a large p we can achieve the same level of coarseness, but with several intermediate eliminations. One possible implementation of this idea is combining Algorithms 3.1 and 3.2. First, using the method given in Algorithm 3.2 we transform the solution and perform a few coarse time-steps with ϕ_a. Then, using the approach in Algorithm 3.1 we can scale ϕ_a such that it corresponds to an untransformed coarse solution $\mathbf{u}_c \in \mathbb{R}^{N/p}$. Together with its corresponding coarse sub-grid, $\mathbf{u}_c$ can serve as an input for the WEM algorithm. Thus, we are tempted to design a recursive wavelet elimination method (RWEM), see Algorithm 3.3. Note that the arrangement of the grid-switching will be such that it happens in a symmetric fashion, requiring an even number of time-steps at each recursion level, see also Figure 3.4.

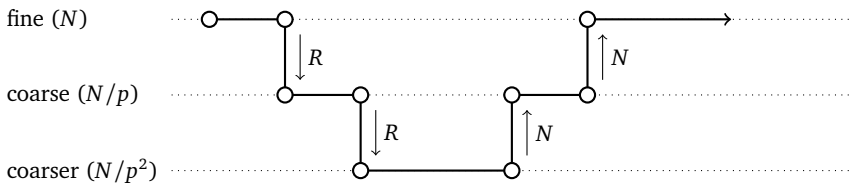


Figure 3.4: Visualization of the RWEM strategy in Algorithm 3.3 with one extra nesting ($r = 1$). R denotes a transformation step with $R = P_\pi Q$. N denotes a Newton-Raphson return iteration, obtaining details one level higher. At each coarse level a WEM-based RK4 time-integration is performed.

Remark. Of course it is possible to create a wide variety of recursive WEM-based grid-switching algorithms. For instance, the partition size p could be adapted at each level. We will, however, stick to the approach formulated in Algorithm 3.3 and visualized in Figure 3.4.

input: Initial discrete solution: $\mathbf{u}(\mathbf{x}, t_0)$, partition size: p,
tolerance for the Newton-Raphson iteration: `tol`,
desired recursion-depth: r,
number of time-steps for each recursion level: $\mathbf{s} \in \mathbb{R}^{r+1}$.
output: Solution $\mathbf{u}(\mathbf{x}, t)$ at time $t = t_0 + rs\Delta t$.

function `rwem`($\mathbf{u}$, p, $\mathbf{s}$, r)
  Construct Q and P_π, based on the partition size p and dimension N of $\mathbf{u}$.
  $\phi := P_\pi Q\mathbf{u}$, obtain partitioned and reordered solution.
  $\phi_d := \phi_{\{1,2,\ldots,N(p-1)/p\}}$, store the details.
  $\phi_a := \phi_{\{N(p-1)/p+1,\ldots,N\}}$, store the averages.
  ..... *Move to the coarse grid, perform $s_{r+1}/2$ time-steps.*
  for $j = 1, 2, \ldots, s_{r+1}/2$ **do**
    Perform an RK4 step on ϕ_a using the WEM discretization (3.31), letting $\widehat{\phi}_d = \phi_d$.
  end for
  ..... *Recursion: move to a coarser grid.*
  if $r > 0$ **then**
    $\mathbf{u}_c := \phi_a/\sqrt{p}$, scale the averages.
    $\mathbf{u}_c :=$ `rwem`($\mathbf{u}_c$, p, $\mathbf{s}$, $r-1$), call `rwem` recursively with decremented r.
    $\phi_a := \sqrt{p}\mathbf{u}_c$, unscale to obtain wavelet averages.
  end if
  ..... *Return from recursion, perform remaining $s_{r+1}/2$ time-steps.*
  for $j = s_{r+1}/2 + 1, \ldots, s_{r+1}$ **do**
    Perform an RK4 step on ϕ_a using the WEM discretization (3.31), letting $\widehat{\phi}_d = \phi_d$.
  end for
  ..... *Return to the fine grid.*
  $\phi_d := 0$, set initial guess for the Newton-Raphson return iteration.
  while $\|\Delta\phi_d\| >$ tol **do**
    Calculate $\Delta\phi_d$ using (3.26), letting $\widehat{\phi}_d = \phi_d$.
    $\phi_d := \Delta\phi_d + \phi_d$, calculate new ϕ_d.
  end while
  $\phi_{\{1,2,\ldots,N(p-1)/p\}} := \phi_d$, update the fine grid-points.
  $\phi_{\{N(p-1)/p+1,\ldots,N\}} := \phi_a$, update the coarse grid-points.
  $\mathbf{u} := Q^T P_\pi^T \phi$, return to a complete, fine solution.
end function

Algorithm 3.3: Recursive WEM grid-switching function.

4 Analysis of the wavelet elimination method

In this section we set out to investigate whether some favorable properties of the complete FVM scheme also propagate to the WEM discretization. Previously, in Section 3.1.1, it is shown that using averages on the coarse grid automatically gives conservation of momentum. Several other properties are of similar importance. We start with obtaining a condition for the *positivity* of the linearized FVM operator. Then, a part of the WEM operator will turn out to inherit this positivy. Furthermore, it will be shown that the complete WEM discretization is monotone whenever the FVM operator is positive. Finally, in the remaining part of this section, we will determine a conservative bound on the step-size for RK4 time-integrations on the coarse grid.

4.1 The linearized FVM operator

Recall the finite volume discretization of Burgers' equation (2.6). Slightly rewritten it is given by

$$\begin{aligned} hv_i &= -\frac{1}{8}\left[(u_i+u_{i+1})^2-(u_{i-1}+u_i)^2\right]+\frac{\mu}{h}\left(u_{i+1}-2u_i+u_{i-1}\right), \\ &= -\frac{1}{8}(u_{i+1}+2u_i+u_{i-1})(u_{i+1}-u_{i-1})+\frac{\mu}{h}\left(u_{i+1}-2u_i+u_{i-1}\right), \quad i=1,2,\ldots,N. \end{aligned} \tag{4.1}$$

For a simple linearization of this discretization we distinguish between previously known values $\widetilde{u}_i$ and unknown values u_i:

$$hv_i \approx -\frac{(\widetilde{u}_{i+1}+2\widetilde{u}_i+\widetilde{u}_{i-1})}{4}\frac{(u_{i+1}-u_{i-1})}{2}+\frac{\mu}{h}\left(u_{i+1}-2u_i+u_{i-1}\right), \tag{4.2}$$

where the term $(\widetilde{u}_{i+1}+2\widetilde{u}_i+\widetilde{u}_{i-1})/4$ is an average of known values around the i-th index. Equation (4.2) may be viewed as a Picard linearization of (4.1), which is a popular method of linearizing the convection term in the Navier-Stokes equation [6].

A non-dimensional parameter, frequently arising in the analysis of discretizations is the *mesh-* or *cell-Péclet number* (see [16], Section 2.15 or [14], Section 3.4). Let the i-th mesh-Péclet number be defined by:

$$\mathrm{Pe}_i = \frac{h}{4\mu}(\widetilde{u}_{i+1}+2\widetilde{u}_i+\widetilde{u}_{i-1}).$$

We will now consider the stationary case with $v_i = 0$ and multiply the linearization (4.2) with $-\frac{h}{\mu}$, giving coefficients in terms of the i-th mesh-Péclet number:

$$-\Big(\frac{1}{2}\mathrm{Pe}_i+1\Big)u_{i-1}+2u_i+\Big(\frac{1}{2}\mathrm{Pe}_i-1\Big)u_{i+1}=0, \tag{4.3}$$

which has the form

$$\alpha_i u_{i-1}+\beta_i u_i+\gamma_i u_{i+1}=0. \tag{4.4}$$

Using (4.4) we can put the linearization in matrix-vector form: $\mathcal{L}\mathbf{u}=0$, where $\mathcal{L}$ is a tridiagonal matrix given by

$$\mathcal{L}=\begin{bmatrix} \beta_1 & \gamma_1 & 0 & \cdots & \alpha_1 \\ \alpha_2 & \beta_2 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \gamma_{N-1} \\ \gamma_N & \cdots & & \alpha_N & \beta_N \end{bmatrix}. \tag{4.5}$$

If the operator $\mathcal{L}$ is of *positive type*, it obeys a discrete maximum (or minimum) principle (see [24] or [27], Section 4.4), which prevents the development of non-physical oscillations due to numerical instabilities (wiggles). For positivity we have the following two requirements (see [27], Definition 4.4.1):

Definition 4.1. The operator $\mathcal{L}$ as defined in (4.5) is of positive type when

(i) $\alpha_i + \beta_i + \gamma_i = 0$, for $i = 2, 3, \ldots, N-1$, and

(ii) $\alpha_i < 0$, $\gamma_i < 0$, for $i = 2, 3, \ldots, N-1$.

Its should be clear that, for $\mathcal{L}$, condition (i) in Definition 4.1 is always satisfied. To obtain a positive operator we also need $\alpha_i < 0$ and $\gamma_i < 0$. If Pe_i is positive, $\gamma_i < 0$ is satisfied and we need $\frac{1}{2}\mathrm{Pe}_i - 1 < 0 \Rightarrow \mathrm{Pe}_i < 2$. If on the other hand Pe_i is negative, $\gamma_i < 0$ holds and we need $-\frac{1}{2}\mathrm{Pe}_i - 1 < 0 \Rightarrow -2 < \mathrm{Pe}_i$. Thus, for a positive operator we need:

$$|\mathrm{Pe}_i| < 2, \tag{4.6}$$

which we will call the *Péclet criterion*.

An operator that is of positive type is also *monotone*[6] (see the Corollary of Theorem A.1.2 in [24]):

Definition 4.2. The operator $\mathcal{L}$ is monotone when $\mathcal{L}\mathbf{x} > 0 \Rightarrow \mathbf{x} > 0$, where the inequality $\mathbf{x} > 0$ implies that all elements of $\mathbf{x}$ are positive.

As with positivity, a monotone operator prevents numerical wiggles (see [27]). However, note that a monotone scheme is not necessarily of positive type. In the next section we will see that the WEM scheme preserves the monotonicity of the FVM discretization.

4.2 Analysis of the linearized WEM operator

Consider again the Picard linearization in a stationary case: $\mathcal{L}\mathbf{u} = 0$, where $\mathcal{L}$ is the tridiagonal matrix with coefficients α_i, β_i, γ_i in the i-th row. In the WEM approach we first transform and then permute the problem, giving

$$P_\pi Q \mathcal{L} Q^T P_\pi^T \begin{bmatrix} \phi_d \\ \phi_a \end{bmatrix} = \begin{bmatrix} \mathcal{G}_{11} & \mathcal{G}_{12} \\ \mathcal{G}_{21} & \mathcal{G}_{22} \end{bmatrix} \begin{bmatrix} \phi_d \\ \phi_a \end{bmatrix} = 0. \tag{4.7}$$

Here we let $\mathcal{G}$ denote the transformed matrix. Eliminating ϕ_d yields an operator on ϕ_a given by the Schur complement of $\mathcal{G}_{11}$ in $\mathcal{G}$, where we assume $\mathcal{G}_{11}$ is invertible:

$$S\phi_a = \left(\mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}^{-1}\mathcal{G}_{12}\right)\phi_a = 0. \tag{4.8}$$

The Schur complement is similar to the WEM discretization matrix, using a (first-order) Picard linearization instead of the (second-order) Newton linearization in (3.31). We will now present a few results regarding this approximation, where we let $\mathcal{G} = P_\pi Q \mathcal{L} Q^T P_\pi^T$, with P_π and Q defined as Section 3. First we show that *positive definiteness* is preserved on the coarse grid.

Definition 4.3. A matrix $\mathcal{L}$ is positive definite when $\mathbf{x}^T\mathcal{L}\mathbf{x} > 0$ for arbitrary $\mathbf{x} \neq 0$.

Lemma 4.1. *If $\mathcal{L}$ is a positive definite matrix, then the Schur complement of $\mathcal{G}_{11}$ in $\mathcal{G}$ is positive definite as well.*

Proof. Let $\mathcal{L}$ be positive definite, that is $\mathbf{x}^T\mathcal{L}\mathbf{x} > 0$ for any $\mathbf{x} \neq 0$. Let $\mathcal{G} = R\mathcal{L}R^T$, with $R = P_\pi Q$ and $R^{-1} = R^T$. Note that this is a similarity transformation. Without loss of generality we can use $\mathbf{x} = R^T\widehat{\mathbf{x}}$, since the transformation matrix $R = P_\pi Q$ has full column rank, implying $R^T\widehat{\mathbf{x}} = 0 \Leftrightarrow \widehat{\mathbf{x}} = 0$. Or, in other words $\mathbf{x} \neq 0 \Rightarrow \widehat{\mathbf{x}} \neq 0$. Then

$$\mathbf{x}^T\mathcal{L}\mathbf{x} > 0 \iff \widehat{\mathbf{x}}^T R\mathcal{L}R^T\widehat{\mathbf{x}} > 0 \iff \widehat{\mathbf{x}}^T\mathcal{G}\widehat{\mathbf{x}} > 0, \quad \text{for arbitrary } \widehat{\mathbf{x}} \neq 0.$$

Hence $\mathcal{G}$ is positive definite.

[6] A property due to Collatz [5]. Here we assume that the operator is non-singular.

To show that the Schur complement of $\mathcal{G}_{11}$ in $\mathcal{G}$ is positive definite we use the block LU decomposition of $\mathcal{G}$:

$$\begin{aligned}
\mathcal{G} = \begin{bmatrix} \mathcal{G}_{11} & \mathcal{G}_{12} \\ \mathcal{G}_{21} & \mathcal{G}_{22} \end{bmatrix} &= \begin{bmatrix} I & O \\ \mathcal{G}_{21}\mathcal{G}_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{G}_{11} & \mathcal{G}_{12} \\ O & S \end{bmatrix}, \quad \text{with } S = \mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}^{-1}\mathcal{G}_{12}, \\
&= \begin{bmatrix} I & O \\ \mathcal{G}_{21}\mathcal{G}_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{G}_{11} & \mathcal{G}_{12} \\ O & S \end{bmatrix} \begin{bmatrix} I & -\mathcal{G}_{21}\mathcal{G}_{11}^{-1} \\ O & I \end{bmatrix} \begin{bmatrix} I & \mathcal{G}_{21}\mathcal{G}_{11}^{-1} \\ O & I \end{bmatrix} \\
&= \begin{bmatrix} I & O \\ \mathcal{G}_{21}\mathcal{G}_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{G}_{11} & -\mathcal{G}_{11}\mathcal{G}_{21}\mathcal{G}_{11}^{-1} + \mathcal{G}_{12} \\ O & S \end{bmatrix} \begin{bmatrix} I & \mathcal{G}_{21}\mathcal{G}_{11}^{-1} \\ O & I \end{bmatrix}. \qquad (4.9)
\end{aligned}$$

Here the term $\begin{bmatrix} I & -\mathcal{G}_{21}\mathcal{G}_{11}^{-1} \\ O & I \end{bmatrix} \begin{bmatrix} I & \mathcal{G}_{21}\mathcal{G}_{11}^{-1} \\ O & I \end{bmatrix}$ is just the identity. Now we have a block decomposition of the form $\mathcal{G} = LUL^T$. Let $\widehat{\mathbf{z}} = L^T\mathbf{z}$, with $\mathbf{z} \neq 0 \Rightarrow \widehat{\mathbf{z}} \neq 0$, since L has full column rank. Then, since $\mathcal{G}$ is positive definite,

$$\begin{aligned}
\mathbf{z}^T\mathcal{G}\mathbf{z} > 0 \iff \mathbf{z}^T LUL^T\mathbf{z} > 0 \iff \widehat{\mathbf{z}}^T U\widehat{\mathbf{z}} > 0 \iff \\
\widehat{\mathbf{z}}^T \begin{bmatrix} \mathcal{G}_{11} & -\mathcal{G}_{11}\mathcal{G}_{21}\mathcal{G}_{11}^{-1} + \mathcal{G}_{12} \\ O & S \end{bmatrix} \widehat{\mathbf{z}} > 0. \qquad (4.10)
\end{aligned}$$

Thus, the upper triangular block matrix is positive definite. Now, let $B = [0, I]^T$ and $\widehat{\mathbf{z}} = B\widetilde{\mathbf{z}}$. Since B has full column rank, $\widehat{\mathbf{z}} \neq 0 \Rightarrow \widetilde{\mathbf{z}} \neq 0$. Then (4.10) is equivalent to

$$\widetilde{\mathbf{z}}^T \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{G}_{11} & -\mathcal{G}_{11}\mathcal{G}_{21}\mathcal{G}_{11}^{-1} + \mathcal{G}_{12} \\ O & S \end{bmatrix} \begin{bmatrix} 0 \\ I \end{bmatrix} \widetilde{\mathbf{z}} > 0 \iff \widetilde{\mathbf{z}}^T S\widetilde{\mathbf{z}} > 0, \quad \text{with arbitrary } \widetilde{\mathbf{z}} \neq 0.$$

Therefore, since $\mathcal{L}$ is positive definite, the Schur complement given by $S = \mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}^{-1}\mathcal{G}_{12}$ is positive definite as well.

□

In the proof of Lemma 4.1 we use an argument similar to proving that every principal submatrix of a positive definite matrix is positive definite, see Corollary 4.2.2 in [11]. If the Péclet criterion is satisfied it can be shown that $\mathcal{L}$ is positive definite. For instance, in Section 4.3 we encounter the Gershgorin circles of $\mathcal{L}$, which show that all of its eigenvalues have a positive real part. In turn this implies that $\mathcal{L}$ is positive definite. Having a positive definite discretization matrix on the coarse grid is a convenient property, as we will see in Section 4.4.1.

Our main concern in this section is whether the favorable positivity property is preserved. Unfortunately, *numerical observations* show that this is not the case. We can, however, show that a part of the Schur complement is of positive type.

Lemma 4.2. *Let $\mathcal{L}$ and $\mathcal{G}_{22}$ be as defined in* (4.5) *and* (4.7) *respectively. If $\mathcal{L}$ is a positive operator then $\mathcal{G}_{22}$ is also a positive operator.*

Proof. First we write $\mathcal{L}$ in block tridiagonal form, where we also include the contributions in the upper right and lower left corner due to the periodic boundary conditions:

$$\mathcal{L} = \begin{bmatrix} T_1 & V_1 & 0 & \cdots & W_1 \\ W_2 & T_2 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & V_{N/p-1} \\ V_{N/p} & \cdots & & W_{N/p} & T_{N/p} \end{bmatrix}.$$

The blocks are given by:

$$T_k = \begin{bmatrix} \beta_{l+1} & \gamma_{l+1} & & & \\ \alpha_{l+2} & \beta_{l+2} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{l+p-1} \\ & & & \alpha_{l+p} & \beta_{l+p} \end{bmatrix}, \; W_k = \begin{bmatrix} 0 & \cdots & 0 & \alpha_{l+1} \\ \vdots & \ddots & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}, \; V_k = \begin{bmatrix} 0 \cdots & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & \ddots & \vdots \\ \gamma_{l+p} & 0 & \cdots & 0 \end{bmatrix},$$

where we let $l = (k-1)p$, with $k = 1, 2, \ldots, N/p$. Then the partitioning gives

$$Q\mathcal{L}Q^T = \begin{bmatrix} H & & & & \\ & H & & & \\ & & H & & \\ & & & \ddots & \\ & & & & H \end{bmatrix} \begin{bmatrix} T_1 & V_1 & 0 & \cdots & W_1 \\ W_2 & T_2 & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & V_{N/p-1} \\ V_{N/p} & \cdots & & W_{N/p} & T_{N/p} \end{bmatrix} \begin{bmatrix} H^T & & & & \\ & H^T & & & \\ & & H^T & & \\ & & & \ddots & \\ & & & & H^T \end{bmatrix},$$

$$= \begin{bmatrix} HT_1H^T & HV_1H^T & 0 & \cdots & HW_1H^T \\ HW_2H^T & HT_2H^T & \ddots & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & HV_{N/p-1}H^T \\ HV_{N/p}H^T & \cdots & & HW_{N/p}H^T & HT_{N/p}H^T \end{bmatrix}. \tag{4.11}$$

By construction, the permutation $PQ\mathcal{L}Q^TP^T$ rearranges the elements of (4.11) such that the lower right block, $\mathcal{G}_{22}$, has the same structure as (4.11), but with the blocks replaced by their respective upper left elements.

In other words, the k-th diagonal element of $\mathcal{G}_{22}$ is the upper left element of HT_kH^T. The first row of H is given by $\mathbf{h}_1 = \frac{1}{\sqrt{p}}(1, 1, \ldots, 1) \in \mathbb{R}^{1\times p}$. Then we have that the upper left element of HT_kH^T is given by

$$\mathbf{h_1} T_k \mathbf{h_1}^T = \frac{1}{p}\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \beta_{l+1} & \gamma_{l+1} & & & \\ \alpha_{l+2} & \beta_{l+2} & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \gamma_{l+p-1} \\ & & & \alpha_{l+p} & \beta_{l+p} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Since we assume that $\mathcal{L}$ is a positive operator, by part (i) of Definition 4.1 we have that $\alpha_i + \beta_i + \gamma_i = 0$. This gives

$$\mathbf{h_1} T_k \mathbf{h_1}^T = \frac{1}{p}\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \beta_{l+1} + \gamma_{l+1} \\ 0 \\ \vdots \\ \alpha_{l+p} + \beta_{l+p} \end{bmatrix} = \frac{1}{p}\left(\beta_{l+1} + \gamma_{l+1} + \alpha_{l+p} + \beta_{l+p}\right).$$

The k-th upper diagonal element of $\mathcal{G}_{22}$ is given by

$$\mathbf{h_1} V_k \mathbf{h_1}^T = \frac{1}{p}\begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & & \ddots & \vdots \\ \gamma_{l+p} & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} = \frac{1}{p}\gamma_{l+p},$$

and we obtain a similar coefficient for the k-th lower diagonal element:

$$\mathbf{h_1} W_k \mathbf{h_1}^T = \frac{1}{p}\begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}\begin{bmatrix}0 & \cdots & 0 & \alpha_{l+1}\\ \vdots & \ddots & & 0\\ \vdots & & \ddots & \vdots\\ 0 & \cdots & \cdots & 0\end{bmatrix}\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix} = \frac{1}{p}\alpha_{l+1}.$$

Now we have that $\left(\mathcal{G}_{22}\phi\right)_k = \widetilde{\alpha}_k\phi_{k-1} + \widetilde{\beta}_k\phi_k + \widetilde{\gamma}_k\phi_{k+1}$, with

$$\begin{aligned}
\widetilde{\alpha}_k &= \frac{1}{p}\alpha_{l+1} = -\frac{1}{p}\left(\frac{1}{2}\mathrm{Pe}_{l+1} + 1\right),\\
\widetilde{\beta}_k &= \frac{1}{p}\left(\beta_{l+1} + \gamma_{l+1} + \alpha_{l+p} + \beta_{l+p}\right) = \frac{1}{p}\left[2 + \frac{1}{2}\left(\mathrm{Pe}_{l+1} - \mathrm{Pe}_{l+p}\right)\right],\\
\widetilde{\gamma}_k &= \frac{1}{p}\gamma_{l+p} = \frac{1}{p}\left(\frac{1}{2}\mathrm{Pe}_{l+p} - 1\right).
\end{aligned}$$

We find that the first condition for a positive operator automatically holds:

$$\widetilde{\alpha}_k + \widetilde{\beta}_k + \widetilde{\gamma}_k = \frac{1}{p}\left(-\frac{1}{2}\mathrm{Pe}_{l+1} - 1 + 2 + \frac{1}{2}\mathrm{Pe}_{l+1} - \frac{1}{2}\mathrm{Pe}_{l+p} + \frac{1}{2}Pe_{l+p} - 1\right) = 0.$$

Then we are left with the requirements given by the second condition of Definition 4.1:

$$\widetilde{\alpha}_k < 0 \Rightarrow \frac{1}{p}\alpha_k < 0, \quad \widetilde{\gamma}_k < 0 \Rightarrow \frac{1}{p}\gamma_k < 0,$$

which are the same conditions we need for $\mathcal{L}$, i.e. $|\mathrm{Pe}_k| < 2$. Thus, we may conclude that $\mathcal{G}_{22}$ is a positive operator if $\mathcal{L}$ is a positive operator. □

The previous lemma shows that a part of the Schur complement preserves the positivity of $\mathcal{L}$. In Section 4.4 we will revisit this result and use it to show that for specific cases the spectrum of $\mathcal{G}_{22}$ dominates the spectrum of the complete Schur complement.

It is easy to provide a counter example in which positivity is not preserved for the entire WEM discretization. A slightly weaker notion is that of a monotone operator. If the Péclet criterion is satisfied, $\mathcal{L}$ will be of positive type and consequently also monotone. In the following theorem we will show that, in this case, WEM preserves the monotonicity property.

Theorem 4.1. *Let $\mathcal{L}$ and S be as defined* (4.5) *and* (4.8) *respectively. If $\mathcal{L}$ is a monotone matrix, then the Schur complement S is monotone as well.*

Proof. Assume $\mathcal{L}$ is monotone and consider the following matrix equation:

$$\mathcal{G}\mathbf{x} = \begin{bmatrix}\mathcal{G}_{11} & \mathcal{G}_{12}\\ \mathcal{G}_{21} & \mathcal{G}_{22}\end{bmatrix}\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{bmatrix} = \begin{bmatrix}0\\ \mathbf{b}_2\end{bmatrix} =: \mathbf{b}, \quad \text{with } \mathbf{b}_2 \geq 0. \tag{4.12}$$

Let the vectors have dimensions $\mathbf{b} \in \mathbb{R}^N$, $\mathbf{x}_1 \in \mathbb{R}^{N(1-1/p)}$ and $\mathbf{x}_2, \mathbf{b}_2 \in \mathbb{R}^{N/p}$, where N and p are as described in the text. Equation (4.12) is similar to the non-stationary WEM discretization, where we assume the details are stationary and eliminate the unknown ϕ_d (cf. Section 3.3).

Using $\mathcal{G} = R\mathcal{L}R^T$ with $R = P_\pi Q$ and $R^{-1} = R^T$, we obtain

$$R\mathcal{L}R^T\mathbf{x} = \mathbf{b} \iff \mathcal{L}R^T\mathbf{x} = R^T\mathbf{b} = R^T[0, \mathbf{b}_2]^T. \tag{4.13}$$

$$\begin{aligned}
\text{Define} \quad & \widehat{\mathbf{b}} := R^T[0, \mathbf{b}_2]^T \quad \text{and}\\
& \mathbf{b}_2 := [\beta_1, \beta_2, \ldots, \beta_{N/p}]^T, \quad \text{with each } \beta_i \geq 0.
\end{aligned}$$

The transformation $\widehat{\mathbf{b}} = R^T[0, \mathbf{b}_2]^T = Q^T P_\pi^T [0, \mathbf{b}_2]^T$ is an inverse permutation followed by an inverse wavelet partitioning.

First the inverse permutation will move the j-th element of $\mathbf{b}_2$ to the $\big[(j-1)p+1\big]$-th element of $\widehat{\mathbf{b}}$ (cf. Equation (3.10)). Then the inverse wavelet partitioning is applied, in which each inverse Haar transform will act on a partition containing an average at the first position and zeros elsewhere. This implies that the j-th partition will be given by $\mathbf{h}_1^T \beta_j$, where $\mathbf{h}_1$ is the first row of the Haar wavelet (see the proof of Lemma 4.2). It is not difficult to check that the complete transformation will return the following:

$$\widehat{\mathbf{b}} = \frac{1}{\sqrt{p}} [\beta_1, \beta_1, \ldots, \beta_1, \beta_2, \beta_2, \ldots, \beta_2, \ldots, \ldots, \beta_{N/p}, \beta_{N/p}, \ldots, \beta_{N/p}]^T.$$

Consequently, we observe that $\mathbf{b}_2 \geq 0 \Rightarrow \widehat{\mathbf{b}} \geq 0$.

Furthermore, Equation (4.13) gives $\mathcal{L}\widehat{\mathbf{x}} = \widehat{\mathbf{b}}$, where $\widehat{\mathbf{x}} = R^T \mathbf{x}$. Since $\mathcal{L}$ is monotone we have that

$$\mathbf{b}_2 \geq 0 \Rightarrow \widehat{\mathbf{b}} \geq 0 \Rightarrow \widehat{\mathbf{x}} \geq 0. \tag{4.14}$$

Now we use that

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = R\widehat{\mathbf{x}} = P_\pi Q \widehat{\mathbf{x}}.$$

Since $\widehat{\mathbf{x}} \geq 0$, the wavelet partitioning $Q\widehat{\mathbf{x}}$ will return positive averages. Next the reordering $P_\pi Q\widehat{\mathbf{x}}$ will move these positive averages to the lower block $\mathbf{x}_2$. Thus, with the transformation $R = P_\pi Q$ we have that $\widehat{\mathbf{x}} \geq 0 \Rightarrow \mathbf{x}_2 \geq 0$ and combined with (4.14) we obtain

$$\mathbf{b}_2 \geq 0 \Rightarrow \mathbf{x}_2 \geq 0.$$

Now we return to the system (4.12). The first equation gives $\mathbf{x}_1 = -\mathcal{G}_{11}^{-1}\mathcal{G}_{12}\mathbf{x}_2$. Substituting this in the second equation and thereby eliminating $\mathbf{x}_1$ gives

$$\left(\mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}\mathcal{G}_{12}\right)\mathbf{x}_2 = \mathbf{b}_2. \tag{4.15}$$

Since $\mathbf{b}_2 \geq 0 \Rightarrow \mathbf{x}_2 \geq 0$, we conclude that the Schur complement $\mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}\mathcal{G}_{12}$ must be monotone. □

Theorem 4.1 shows that, with the appropriate transformations, we can eliminate a significant part of the solution vector, while maintaining a monotone discretization matrix. This result implies that, whenever the Péclet criterion is satisfied, we can choose any large partition size without the risk of numerical wiggles. As expected, in later sections we will see that the partition size is restricted by other factors.

Now that we have investigated the spatial stability properties of the discretization it is time to move to the non-stationary case and discuss some stability aspects of the time integration.

4.3 Time integration

The linearized finite volume discretization in the non-stationary case is given by

$$\mathbf{u}_t = -\frac{\mu}{h^2}\mathcal{L}\mathbf{u}, \tag{4.16}$$

where $\mathcal{L}$ is the tridiagonal matrix described in Section 4.1. Equation (4.16) is a system of ODEs, which will be integrated in time using the classic four-stage Runge-Kutta method (RK4). For an explanation of this method and its stability criteria see, for instance, Sections 9-10 of [12], Sections II.1-2 of [14] or Section 5.8 of [27].

The numerical solution of (4.16) will be computed at integer multiples of a fixed timestep Δt, i.e. at $t_n = n\Delta t$, with $n \in \mathbb{N}$. A solution at $t_n = n\Delta t$ is denoted by $\mathbf{u}^n$. For a system of ODEs of the form $\mathbf{u}_t = F(\mathbf{u})$, a general Runge-Kutta scheme is given by

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \sum_{i=1}^{s} b_i F(\widetilde{\mathbf{u}}_{ni}), \quad \widetilde{\mathbf{u}}_{ni} = \mathbf{u}^n + \Delta t \sum_{j=1}^{s} a_{ij} F(\widetilde{\mathbf{u}}_{nj}),$$

where the coefficients b_i and a_{ij} are to be prescribed by the method. The classic RK4 method requires:

$$s = 4, \quad \mathbf{b} = \left[\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}\right]^T \quad \text{and} \quad A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Let $\tau = (\Delta t \mu)/h^2$, then the RK4 scheme applied to (4.16) leads to the following recursive relation:

$$\mathbf{u}^{n+1} = \left(I - \tau\mathcal{L} + \frac{\tau^2}{2}\mathcal{L}^2 - \frac{\tau^3}{6}\mathcal{L}^3 + \frac{\tau^4}{24}\mathcal{L}^4 \right) \mathbf{u}^n = \mathcal{R}(-\tau\mathcal{L})\mathbf{u}^n. \tag{4.17}$$

Note that the stability function $\mathcal{R}(z) = 1 + z + z^2/2 + z^3/6 + z^4/24$ is an approximation of the exponential that gives the analytical solution of the system:

$$\mathbf{u}(t + \Delta t) = \exp\left(-\frac{\Delta t \mu}{h^2}\mathcal{L} \right) \mathbf{u}(t).$$

Assuming $\mathcal{L}$ is diagonalizable we can write $\mathcal{L} = PDP^{-1}$, where P contains the eigenvectors of $\mathcal{L}$ and D the corresponding eigenvalues:

$$\mathcal{L} = P \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix} P^{-1}. \tag{4.18}$$

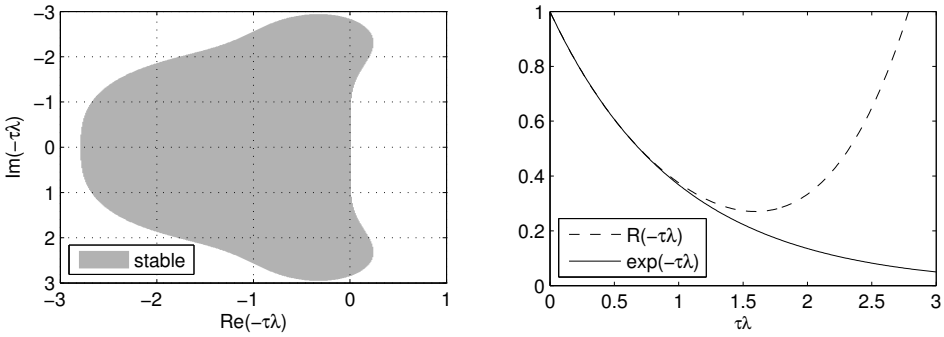
Substituting (4.18) in (4.17) and letting $\mathbf{u}^n = P\mathbf{w}^n$ we find

$$\mathbf{u}^{n+1} = P\mathcal{R}(-\tau D)P^{-1}\mathbf{u}^n \iff P^{-1}\mathbf{u}^{n+1} = \mathcal{R}(-\tau D)P^{-1}\mathbf{u}^n \iff \mathbf{w}^{n+1} = \mathcal{R}(-\tau D)\mathbf{w}^n. \tag{4.19}$$

In fact, (4.19) is a decoupling of the system and the j-th equation is given by

$$w_j^{n+1} = \mathcal{R}(-\tau\lambda_j)w_j^n = \mathcal{R}(-\tau\lambda_j)^n w_j^0. \tag{4.20}$$

We obtain absolute stability if we require $|\mathcal{R}(-\tau\lambda_j)| < 1$ for each eigenvalue λ_j, see Figure 4.1(a). However, apart from absolute stability we also want the approximation $\mathcal{R}(z)$ to mimic the damping of the exponential. For real eigenvalues Figure 4.1(b) shows how the approximation separates from the exponential function. We define a value for which the damping appears to be optimal: $\tau\lambda = \delta_{\text{opt}} \approx 1.6$.



(a) (b)

Figure 4.1: Absolute stability region for the RK4 method (a), where $|\mathcal{R}(-\tau\lambda)| < 1$, $\lambda \in \sigma(\mathcal{L})$. Damping of the exponential and the approximating stability polynomial (b).

As we see from Figure 4.1(b) it is not desirable to take as bound for $\tau\lambda$ the actual bound of the absolute stability region. Choosing the optimal damping value as bound seems to be a safer option, since we aim to imitate the analytical damping and up to $\tau\lambda = \delta_{\text{opt}}$ the approximation $\mathcal{R}$ is more or less acceptable.

For the calculation of the stability limit we need to localize the spectrum of $\mathcal{L}$. The i-th row of $\mathcal{L}$ contains coefficients α_i, β_i and γ_i, as given by Equations (4.3)-(4.4). Let the spectrum of $\mathcal{L}$ be denoted by $\sigma(\mathcal{L})$. Using Gershgorin's disk theorem ([19], Section 7.2) we find:

$$\sigma(\mathcal{L}) \subset \bigcup_i \{z \in \mathbb{C} : |z - \beta_i| \leq |\alpha_i| + |\gamma_i|\}. \tag{4.21}$$

Due to the positivity of $\mathcal{L}$, $\beta_i = |\alpha_i| + |\gamma_i|$ and since $\beta_i = 2$ for $i = 1, 2, \ldots, N$ we find that (4.21) describes a single large disk with center 2 and radius 2. Hence all eigenvalues have a positive real part and $\rho(\mathcal{L}) \leq 4$, where $\rho(\cdot)$ denotes the spectral radius. Note that the same bound follows from the fact that for every consistent norm $||\cdot||$ it holds that $\rho(\mathcal{L}) \leq ||\mathcal{L}||$. Using the infinity norm and the elements of $\mathcal{L}$ we immediately see that the $||\mathcal{L}||_\infty = 4$, since $||\mathcal{L}||_\infty := \max_i \sum_j |\mathcal{L}_{ij}|$.

Thus, for stability and sufficient damping properties it seems fair to require

$$\tau\rho(\mathcal{L}) \leq \tau||\mathcal{L}||_\infty = 4\frac{\Delta t\mu}{h^2} \leq \delta_{\text{opt}}. \tag{4.22}$$

With this requirement we satisfy $\tau\lambda_j \leq \delta_{\text{opt}}$ for all eigenvalues $\lambda_j \in \sigma(\mathcal{L})$. Note that the factor τ acts as a contraction on the Gershgorin disks in (4.21) such that the values $-\tau\lambda_j$ lie inside a circle with center $-\delta_{\text{opt}}/2$ and radius $\delta_{\text{opt}}/2$. If δ_{opt} is chosen correctly this disk will lie within the stability area shown in Figure 4.1(a).

4.4 RK4 and the WEM scheme

To obtain a stability condition for the RK4 method applied to the WEM discretization, we need a bound on the spectrum of the Schur complement: $S = \mathcal{G}_{22} - \mathcal{G}_{21}\mathcal{G}_{11}^{-1}\mathcal{G}_{12}$. In the case of a symmetric positive definite (s.p.d.) $\mathcal{G}$ it is easy to show that the spectrum of S is bounded by the spectral radius of $\mathcal{G}_{22}$.

Lemma 4.3. *Let S, $\mathcal{G}$ and $\mathcal{G}_{22}$ be given by* (4.8) *and* (4.7). *Suppose $\mathcal{G}$ is symmetric positive definite. Then we have that $\rho(S) \leq \rho(\mathcal{G}_{22})$, where $\rho(\cdot)$ denotes the spectral radius.*

Proof. Since $\mathcal{G}$ is symmetric we see from the LUL^T decomposition in (4.9) that S must be symmetric as well. This enables the use of the Rayleigh quotient $(\mathbf{x}^*S\mathbf{x})/(\mathbf{x}^*\mathbf{x})$, where $\mathbf{x}$ is an arbitrary vector and $\mathbf{x}^*$ its conjugate transpose. Moreover, using Lemma 4.1 we find that S must be symmetric positive

definite, which implies that all its eigenvalues are positive and real. Then, for the Rayleigh quotient it holds that

$$\rho(S) = \lambda_{\max} = \max_{\mathbf{x}} \frac{\mathbf{x}^* S \mathbf{x}}{\mathbf{x}^* \mathbf{x}},$$

where $\lambda_{\max}$ is the largest eigenvalue in $\sigma(S)$. Using that $\mathcal{G}_{21} = \mathcal{G}_{12}^*$ we obtain

$$\frac{\mathbf{x}^* S \mathbf{x}}{\mathbf{x}^* \mathbf{x}} = \frac{\mathbf{x}^*(\mathcal{G}_{22} - \mathcal{G}_{12}^* \mathcal{G}_{11}^{-1} \mathcal{G}_{12})\mathbf{x}}{\mathbf{x}^* \mathbf{x}} > 0 \iff \frac{\mathbf{x}^* \mathcal{G}_{12}^* \mathcal{G}_{11}^{-1} \mathcal{G}_{12} \mathbf{x}}{\mathbf{x}^* \mathbf{x}} < \frac{\mathbf{x}^* \mathcal{G}_{22} \mathbf{x}}{\mathbf{x}^* \mathbf{x}}.$$

Every principal submatrix of a positive definite matrix is positive definite ([11], Corollary 4.2.2), therefore $\mathcal{G}_{11}$ is positive definite, which implies that $\mathcal{G}_{11}^{-1}$ is positive definite: $\mathbf{y}^* \mathcal{G}_{11}^{-1} \mathbf{y} > 0 \ \forall \ \mathbf{y} \neq 0$. Now, letting $\mathbf{y} = \mathcal{G}_{12}\mathbf{x}$ shows that $\mathcal{G}_{12}^* \mathcal{G}_{11}^{-1} \mathcal{G}_{12}$ is at least positive semidefinite:

$$\frac{\mathbf{x}^* \mathcal{G}_{12}^* \mathcal{G}_{11}^{-1} \mathcal{G}_{12} \mathbf{x}}{\mathbf{x}^* \mathbf{x}} \geq 0 \ \forall \ \mathbf{x} \neq 0.$$

Then it follows that

$$\frac{\mathbf{x}^*(\mathcal{G}_{22} - \mathcal{G}_{12}^* \mathcal{G}_{11}^{-1} \mathcal{G}_{12})\mathbf{x}}{\mathbf{x}^* \mathbf{x}} \leq \frac{\mathbf{x}^* \mathcal{G}_{22} \mathbf{x}}{\mathbf{x}^* \mathbf{x}} \iff \max_{\mathbf{x}} \frac{\mathbf{x}^* S \mathbf{x}}{\mathbf{x}^* \mathbf{x}} \leq \max_{\mathbf{x}} \frac{\mathbf{x}^* \mathcal{G}_{22} \mathbf{x}}{\mathbf{x}^* \mathbf{x}} \iff \rho(S) \leq \rho(\mathcal{G}_{22}).$$

□

Lemma 4.3 can be interpreted as having the *field of values*[7] of $\mathcal{G}_{22}$ dominate over that of S. Due to the proof of Lemma 4.2 we know the entries of $\mathcal{G}_{22}$, from which we can easily calculate a bound on its spectrum, which in turn will give a bound on the spectrum of the Schur complement. In addition, for certain non-symmetric $\mathcal{G}$ we expect this dominating behavior to occur as well. In the following lemma we will show an implication of having $\rho(S) \leq \rho(\mathcal{G}_{22})$.

Lemma 4.4. *Let S, $\mathcal{G}$ and $\mathcal{G}_{22}$ be given by* (4.8) *and* (4.7) *and let $\mathcal{L}$ be of positive type. If $\rho(S) \leq \rho(\mathcal{G}_{22})$ then $\rho(S) < 2||\mathcal{L}||_\infty/p$, where p is the partition size.*

Proof. Recall that in the proof of Lemma 4.2 we obtain the entries in the k-th row of $\mathcal{G}_{22}$:

$$\widetilde{\alpha}_k = -\frac{1}{p}\left(\frac{1}{2}\mathrm{Pe}_{l+1} + 1\right), \quad \widetilde{\beta}_k = \frac{1}{p}\left[2 + \frac{1}{2}\left(\mathrm{Pe}_{l+1} - \mathrm{Pe}_{l+p}\right)\right], \quad \text{and} \quad \widetilde{\gamma}_k = \frac{1}{p}\left(\frac{1}{2}\mathrm{Pe}_{l+p} - 1\right),$$

with $l = (k-1)p$. The infinity norm of $\mathcal{G}_{22}$ is given by

$$||\mathcal{G}_{22}||_\infty = \max_k \left(|\widetilde{\alpha}_k| + |\widetilde{\beta}_k| + |\widetilde{\gamma}_k|\right).$$

Since $\mathcal{L}$ is of positive type the Péclet condition is satisfied: $|\mathrm{Pe}_i| < 2$, for $i = 1, 2, \ldots, N$. Resolving the absolute values and using $||\mathcal{L}||_\infty = 4$ gives

$$||\mathcal{G}_{22}||_\infty = \max_k \frac{1}{p}\left(4 + \mathrm{Pe}_{l+1} - \mathrm{Pe}_{l+p}\right) = \frac{||\mathcal{L}||_\infty}{p} + \max_k \frac{1}{p}\left(\mathrm{Pe}_{l+1} - \mathrm{Pe}_{l+p}\right) < \frac{8}{p}, \tag{4.23}$$

since $-4 < \mathrm{Pe}_{l+1} - \mathrm{Pe}_{l+p} < 4$. From $\rho(S) \leq \rho(\mathcal{G}_{22}) \leq ||\mathcal{G}_{22}||_\infty$ it follows that

$$\rho(S) < \frac{8}{p} = \frac{2||\mathcal{L}||_\infty}{p}.$$

□

[7] The field of values is the region with values attained by the Rayleigh quotient while $\mathbf{x}$ runs over all of $\mathbb{C}^N$, see e.g. [21], Section 1.9.

The stability and damping requirement of the RK4 time-integration scheme applied to the WEM discretization is similar to (4.22). The parameter $\tau = (\Delta t \mu)/h^2$ remains unchanged and again we require $\tau \lambda_j \leq \delta_{\text{opt}}$, but now for the eigenvalues $\lambda_j \in \sigma(S)$. Suppose we have that $\rho(S) \leq \rho(\mathcal{G}_{22})$. Then, using Lemma 4.4 we obtain the following criterion:

$$\tau \rho(S) < \frac{2\tau ||\mathcal{L}||_\infty}{p} = \frac{8}{p} \frac{\Delta t \mu}{h^2} \leq \delta_{\text{opt}}. \tag{4.24}$$

Whenever $\rho(S) \leq \rho(\mathcal{G}_{22})$ we are therefore allowed to select a timestep that is a factor $p/2$ greater than the timestep for the original FVM scheme.

Remark. In fact, from (4.23) we see that the infinity norm depends on the maximum over all partitions of $\frac{1}{p}\left(\text{Pe}_{l+1} - \text{Pe}_{l+p}\right)$, which represents the difference of the velocity profile over a single partition. This demonstrates that (4.24) is a worst case scenario and we can weaken the requirement with a parameter $\eta \in [0, 1]$, depending on the structure of the velocity profile.

$$\frac{4}{p} \tau (2 - \eta) \leq \delta_{\text{opt}}.$$

For a constant solution we can use $\eta = 1$, whereas for a solution with large shocks we may need to select $\eta \ll 1$.

4.4.1 Energy decay

To conclude the analysis of WEM we will briefly investigate its energy characteristics. After the wavelet transformation and elimination of the details we obtain the following time-dependent relation (cf. (4.16)):

$$\frac{h^2}{\mu} \psi_a = -\left(\mathcal{G}_{22} - \mathcal{G}_{21} \mathcal{G}_{11}^{-1} \mathcal{G}_{12}\right) \phi_a, \tag{4.25}$$

where $\psi_a := \frac{d}{dt} \phi_a$. In Section 3.1.1 we show that, due to the wavelet transform, conservation of momentum holds on the coarse grid. Using the Picard approximation of the WEM operator we will now investigate the conservation of energy. A similar discussion for a standard FVM discretization can be found in [24].

Let $\mathcal{H}$ be a diagonal matrix given by $\mathcal{H} = \frac{h^2}{\mu} I$, with $I \in \mathbb{R}^{N/p \times N/p}$ the identity matrix and let $S = \mathcal{G}_{22} - \mathcal{G}_{21} \mathcal{G}_{11} \mathcal{G}_{12}$. Note that $\mathcal{H}$ is symmetric positive definite and S is positive definite whenever the linearized FVM operator $\mathcal{L}$ is positive definite. Equation (4.25) gives

$$\mathcal{H} \psi_a + S \phi_a = 0. \tag{4.26}$$

Let the *energy* of ϕ_a be given by $||\phi_a||_{\mathcal{H}}^2 := \phi_a^* \mathcal{H} \phi_a$, which is a valid norm since $\mathcal{H}$ is positive definite. Using (4.26) we find that the change of energy is given by

$$\begin{aligned} \frac{d}{dt} ||\phi_a||_{\mathcal{H}}^2 &= \psi_a^* \mathcal{H} \phi_a + \phi_a^* \mathcal{H} \psi_a \\ &= -\phi_a^* S^* \phi_a - \phi_a^* S \phi_a = -\phi_a^* \left(S + S^*\right) \phi_a. \end{aligned}$$

Since S is positive definite but not skew-Hermitian, the energy must be decreasing. Whether the rate of decay corresponds to the diffusion in the PDE remains unanswered. We are satisfied with noting that the energy in our scheme is *non-increasing*.

5 Test problems

In this and the next section we will present numerical solutions of Burgers' equation, acquired using several grid-switching techniques. We will restrict ourselves to two distinct initial velocity profiles for the decaying Burgers equation. Let $x \in [0, 2\pi]$ and consider the following cases:

(A) A peak at the center of the interval with amplitude a and slopes determined by c_l and c_r for the left and right part respectively, see Figure 5.1(a):

$$u(x,0) = \begin{cases} a\exp(-c_l(x-\pi)^2) & \text{for } x \le \pi, \\ a\exp(-c_r(x-\pi)^2) & \text{for } x > \pi. \end{cases}$$

(B) An initial profile given by a random walk, see Figure 5.1(b). Since we require periodic boundaries we will define the random walk as a truncated Fourier expansion, a construction also used in [17]:

$$u(x,0) = \omega \operatorname{Re}\left[\sum_{k=-N/2}^{N/2-1} c_k e^{ikx}\right], \quad \text{with } c_k \sim N(0, k^{-2}).$$

The Fourier components c_k are complex, normally distributed random variables with zero mean and standard deviation $\sigma = \frac{1}{|k|}$. The parameter ω is a scaling.

A time-integration of Burgers' equation will, for sufficiently small diffusion μ, lead to shock-like solutions. Profile (A) will tend to a single decaying shock, traveling to the right. Profile (B) will show a multitude of shocks on different scales. Prescribing a random walk as initial velocity for Burgers' equation in the limit $\mu \to 0$ has been used as a model for turbulence, see for instance [17], [22], [25] or section 5 of [9].

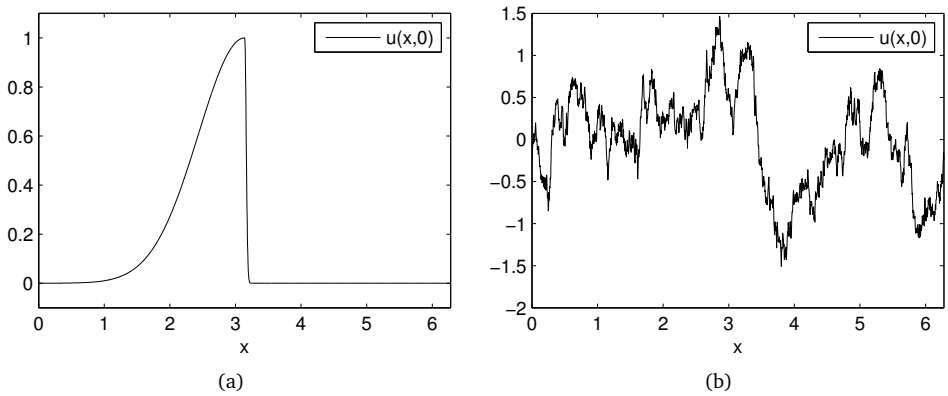


Figure 5.1: Velocity profile (A) describing a single shock with parameters $a = 1$, $c_l = 1$ and $c_r = 1 \times 10^3$ (left), velocity profile (B) given by a random walk and scaled with $\omega = 1 \times 10^3$ (right).

5.1 Preliminary spectral code results

To show the time-evolution of the velocity profiles we will present numerical results using a Fourier-Galerkin (FG) spectral discretization, integrated in time with RK4. A detailed explanation of this method is given in appendix A and in [3]. Spectral methods have their basis functions defined on the entire domain. In a sense, such a method is aware of the global structure of the solution, achieving adequate approximations even at low resolutions.

Performing calculations on a coarse sub-grid using the FG spectral method is equivalent to temporarily filtering the highest frequencies. The implementation of such a filter is straightforward and a few results will be given in Section 5.1.2. However, geometrically a spectral method is less flexible than FVM and we are therefore not abandoning the WEM-approach in the finite volume framework.

Typical time-evolutions of the velocity profiles (A) and (B) are given in Figures 5.2 and 5.3. In both cases the diffusion will be $\mu = 2 \times 10^{-3}$, the number of Fourier modes in the FG spectral implementation will be $N = 2^{11}$ and the time-step in RK4 is taken $\Delta t = 1 \times 10^{-3}$. As expected, profile (A) gives a shock moving to the right with an exponentially decaying amplitude. Profile (B) shows the development of a few shocks on different levels. To obtain more shocks we should drastically decrease the amount of diffusion.

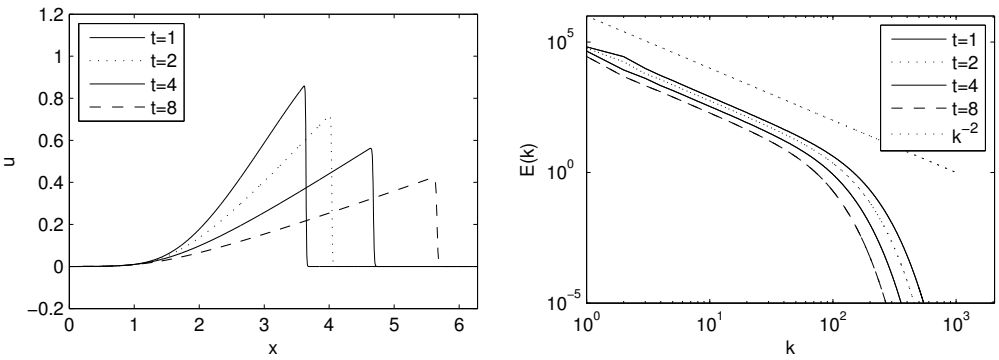


Figure 5.2: Solution (left) and Fourier spectrum (right) at $t = 1, 2, 4, 8$. Initial profile (A) with $a = 1$, $c_l = 1$ and $c_r = 1 \times 10^3$, diffusion $\mu = 2 \times 10^{-3}$, time-step $\Delta t = 1 \times 10^{-3}$, number of Fourier modes $N = 2^{11}$.

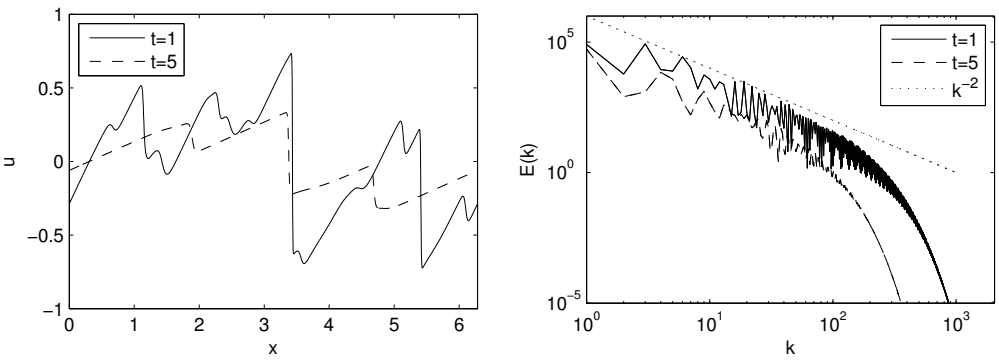


Figure 5.3: Solution (left) and Fourier spectrum (right) at $t = 1, 5$. Initial profile (B) with $\omega = 1 \times 10^3$, diffusion $\mu = 2 \times 10^{-3}$, time-step $\Delta t = 1 \times 10^{-3}$, number of Fourier modes $N = 2^{11}$.

One of the advantages of a spectral method is that the energy spectrum of the numerical solution is readily available. In Figures 5.2 and 5.3 we observe that the energy spectra obey the power law k^{-2}, until dissipation takes over at the high frequencies. This behavior indicates an *energy cascade*. Due to the non-linear convective transport term, energy will gradually cascade from the large scales to the smaller scales. The frequency range in which the convective motion, combined with the inertia of the medium gives this gradual energy transfer, is called the *inertial range*.

In the next section we obtain a cascade, approximating a slightly different power law, corresponding to a *turbulent* flow. To achieve this we need the non-linearity to dominate over the linear diffusion. Fortunately, a spectral method allows for a simple implementation of higher order spatial derivatives. Choosing a higher order diffusion term gives less dissipation at lower frequencies and more at higher frequencies, allowing the turbulent behavior to fully develop.

5.1.1 Initial Brownian motion and higher order diffusion

In [4], a randomly forced Burgers' equation is solved in the inviscid limit $\mu = 9 \times 10^{-40}$. To achieve little dissipation in the inertial range and a sharp cutoff in the dissipation range a higher order diffusion term is taken:

$$\mu(-1)^{q+1}\frac{\partial^{2q}}{\partial x^{2q}}u, \quad \text{with} \quad q = 6.$$

The authors show that the obtained energy spectrum is in accordance with Kolmogorov's 5/3-law.

In this section we will try to obtain a similar result by letting an unforced Burger's equation start with a profile given by Brownian motion. Here we will take similar parameters: $\mu = 1 \times 10^{-40}$ and $q = 6$. The number of Fourier modes is taken $N = 2^{13}$, the scaling $\omega = 1$, the time-step $\Delta t = 5 \times 10^{-4}$ and the time-integration is ended at $t = 10$. See Figure 5.4.

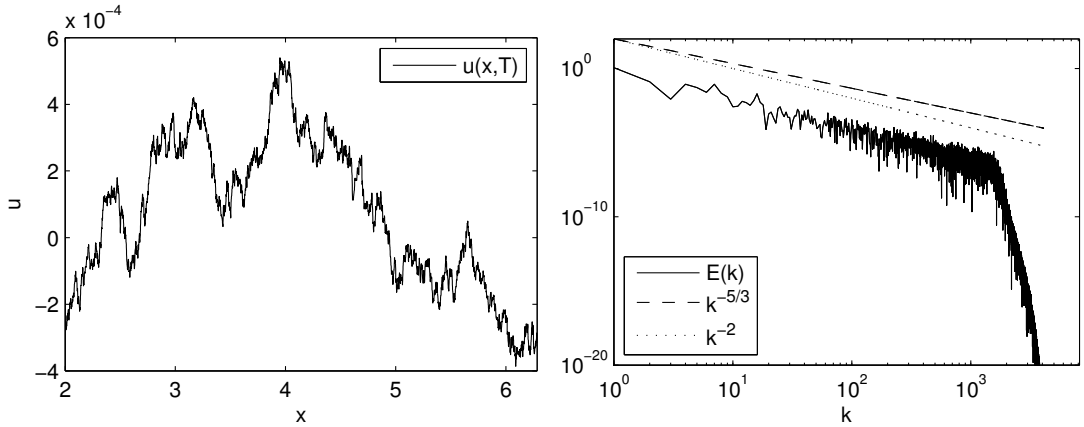


Figure 5.4: Solution (left) and Fourier spectrum (right) of Burgers' equation at time $t = 10$, with initial profile (B) and parameters as given in the text.

The inertial range of the spectrum in Figure 5.4 tends to the power law $k^{-5/3}$, which verifies the possibility of achieving a turbulent energy spectrum in Burgers' equation. Unfortunately, with the FVM-based results in the upcoming sections we will not be able to achieve similar turbulent spectra. Still, we will be able to distinguish an inertial and a dissipative range.

With grid-switching in mind it will be interesting to try reducing the resolution well within the inertial range. Since in this range we expect the small scales to still interact with the larger scales of motion. Simply filtering out these small scales will result in significant errors, as we will see in the next section.

5.1.2 Filtering

In anticipation of the WEM-based grid-switching we will present a few results of an NGS-like approach to the FG spectral discretization. Consider again initial profile (B) with $\omega = 1$. We let $\mu = 3 \times 10^{-5}$, $N = 2^{11}$ and $\Delta t = 1 \times 10^{-3}$. The results are shown in Figure 5.5.

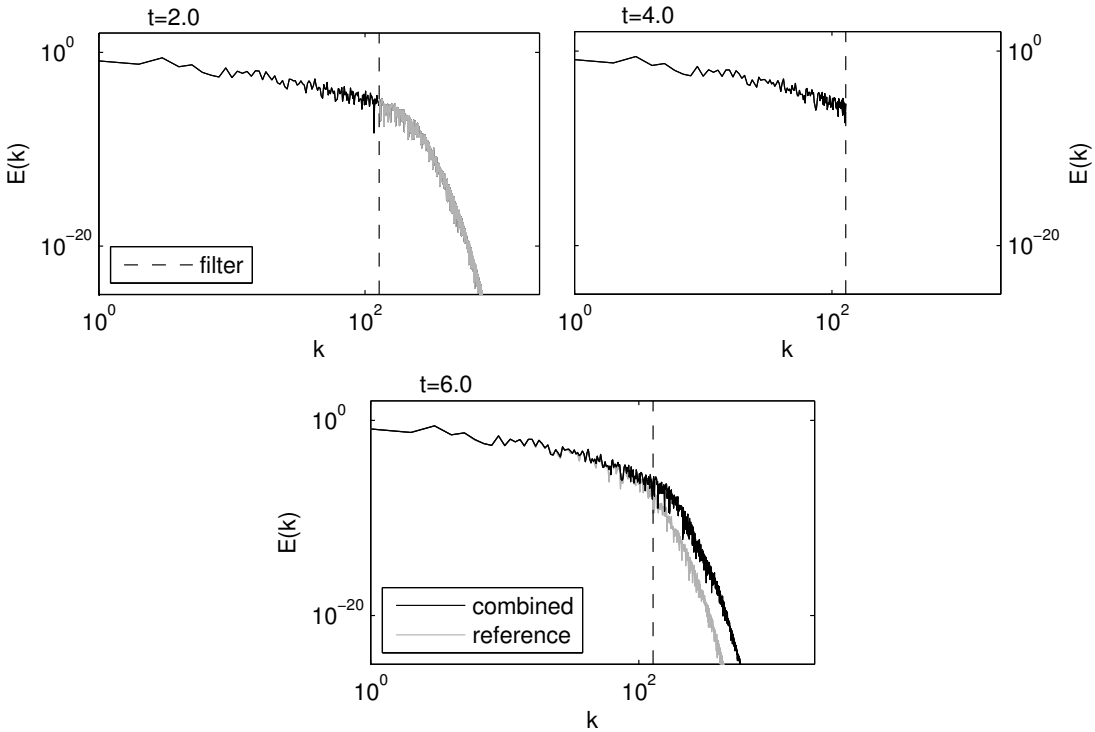


Figure 5.5: Energy spectra of the time-integration results, where the high frequencies are temporarily filtered out. In the lower plot the reference is given by a solution without the filter.

The procedure for Figure 5.5 is as follows:

- A normal time-integration is performed up to $t = 2$ (upper left plot).
- The spectrum is reduced with a factor $p = 8$.
- With this coarse solution a new time-integration is started up to $t = 4$ (upper right plot).
- The stored fine spectrum is combined with the new coarse spectrum and integrated in time up to $t = 6$ (lower plot).

In the lower plot of Figure 5.5 a large discrepancy between the combined and a reference solution is observed. After combining the two separate spectra we see that the result will lag behind. Thus, although the implementation of this spectral coarsening is straightforward its results are very inaccurate. For the FVM-based NGS method this does not look promising, since it employs a similar strategy.

Remark. The most naive form of grid-switching in a spectral method does not behave very well, but there might exist some simple improvements. For instance, as in RWEM we could use small steps and switch recursively, perhaps even eliminating the highest frequencies and using the same model to find an approximation when returning to the fine grid.

5.2 Grid-switching schemes

Before starting the numerical experiments with the FVM-based methods, we need to lay out the grid-switching schemes we plan on using. We are interested in three different types of schemes. The first will be one that alternates between s_f steps on the fine grid (N grid-points) and s_c steps on the coarse grid (N/p grid-points). This strategy will be called type (I), see Figure 5.6.

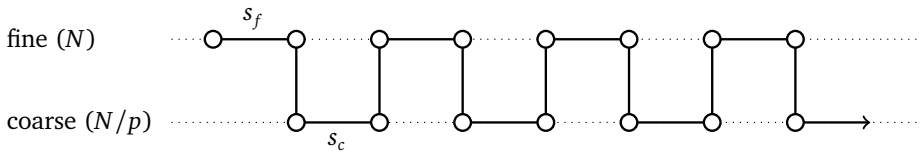


Figure 5.6: Grid-switching scheme of type (I).

The second scheme, type (II), will be similar to type (I) but with an added recursion. On the fine grid we will perform s_f time-steps and for each nested depth we will prescribe $\{s_{c,r}, s_{c,r-1}, \ldots, s_{c,0}\}$, with r the recursion parameter defined in Algorithm 3.3 and $r = 0$ corresponding to the deepest level. Scheme (II) will repeat the cycle until a desired final time is reached, see Figure 5.7.

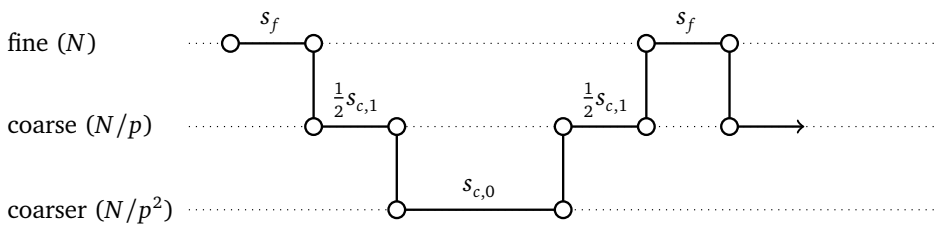


Figure 5.7: Grid-switching scheme of type (II), with $r = 1$.

The final scheme will perform a single step on the fine grid and immediately turn to the coarse grid. At fixed intervals s_c it will revisit the fine grid and perform a single time-step. Then, after n_s of these intervals we return to the complete grid and perform a fine time-integration until some prescribed stop-criteria is reached. We will let this scheme be type (III), see Figure 5.8.

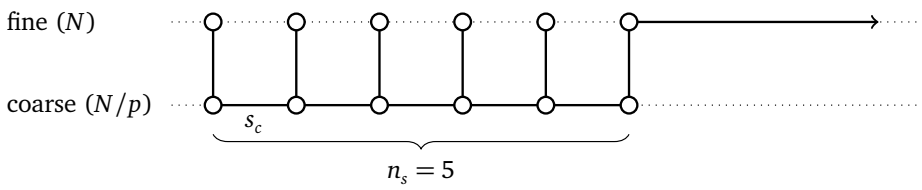


Figure 5.8: Grid-switching scheme of type (III), letting $n_s = 5$.

6 Numerical results

In this section numerical experiments will be presented comparing NGS (Algorithm 3.1), WEM (Algorithm 3.2) and RWEM (Algorithm 3.3), using a fine FVM solution as reference. We start with grid-switching scheme (I) and initial profile (A). The model will contain some diffusion: $\mu = 2 \times 10^{-3}$. The step-size Δt and mesh-width $h = 2\pi/(N-1)$ will be chosen such that both the Péclet criterion (4.6) and the stability requirement (4.22) are satisfied. Note that on a coarse level we are allowed to use a larger time-step. To be on the safe side we will choose to increase the coarse time-step size with a factor $p/2$ (cf. (4.24)). Results for several partition sizes p are shown in Figure 6.1.

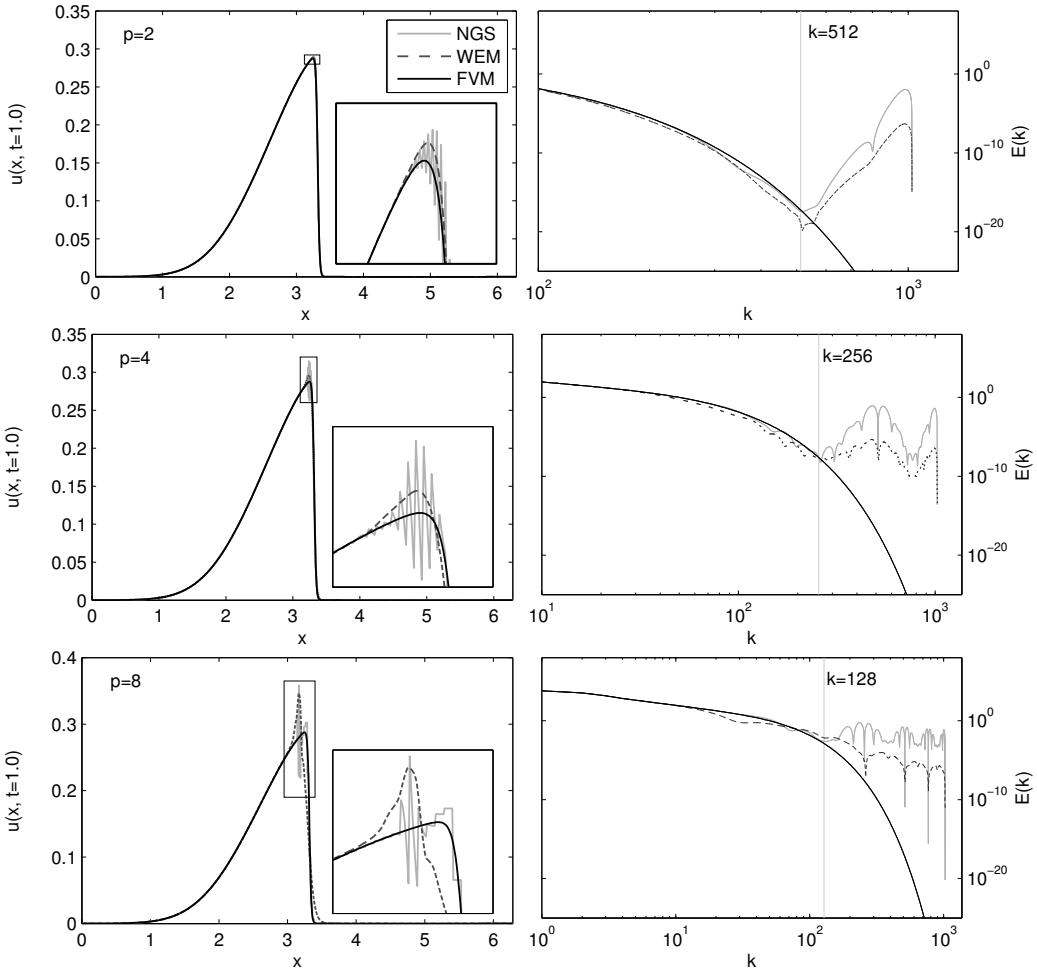


Figure 6.1: Solution (left) and close-up of the energy spectrum (right), using initial velocity (A) and grid-switching of type (I) at time $t = 1$. Grid-switching parameters: $s_f = 10$, $s_c = 200$. Initial velocity profile: $a = .3$, $c_l = 1$, $c_r = 1 \times 10^3$. Diffusion: $\mu = 2 \times 10^{-3}$, number of grid points: $N = 2^{11}$, time-step: $\Delta t = 1.25 \times 10^{-3}$.

Each spectrum in Figure 6.1 contains a vertical line drawn at $k = N/(2p)$, which corresponds to the highest representable frequency on the coarse grid. For higher frequencies the spectrum of NGS and WEM both separate from the fine FVM spectrum, where NGS shows some more separation than WEM. It seems obvious to call $k = N/(2p)$ the *separation line*.

The solutions on the left side of Figure 6.1 show that both WEM and NGS give inaccurate results for large p. Note that the numerical solutions using NGS show the *out of sync* behavior discussed in Section 3.4, which appears to worsen for coarser grids. We will now try to obtain the same level of coarseness ($p = 8$) recursively with RWEM and compare its results to WEM.

6.1 Recursive grid-switching results

In this comparison we have to deal with a partition size for WEM and one for RWEM. Let p_r denote the partition size used in the recursive scheme. The recursive solution will be calculated using 4 time-steps at each intermediate level before arriving at the level corresponding to $N/8$ grid-points, where $s_{c,0} = 192$ time-steps will be taken. With $p_r = 2$ this means setting the recursion parameter to $r = 2$, giving $N/p_r^{r+1} = N/8$. Hence, we go 'down' three levels. In both WEM and RWEM we will let the time-step at the coarsest level be $4\Delta t$, where Δt is the time-step at the fine level. The results are shown in Figure 6.2.

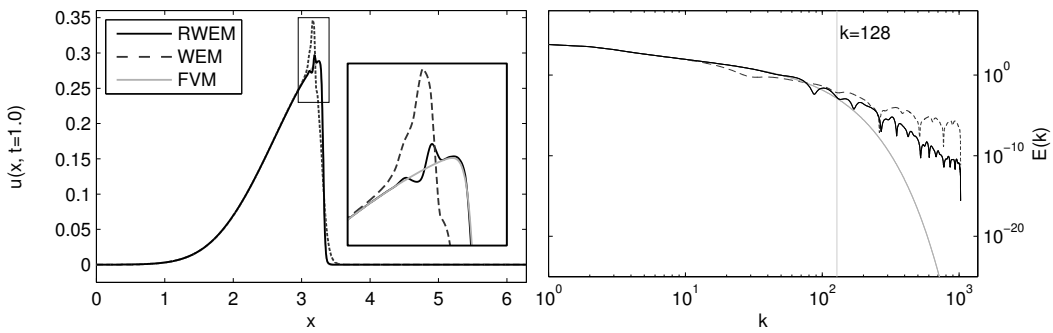


Figure 6.2: Solution (left) and energy spectrum (right), using initial velocity (A) and grid-switching of type (I) for WEM ($p = 8$) and type (II) for RWEM ($p_r = 2$). Type (II) grid-switching parameters: $s_f = 10$, $s_{c,0} = 192$, $s_{c,1} = 4$, $s_{c,2} = 4$. The other parameters remain unchanged.

Both the solution and the spectrum given by RWEM show significant improvements. Recall from Figure 3.4 that we now have extra, nested transformations and return iterations. It appears that having these intermediate partitionings, short WEM-based time-steps and extra Newton-Raphson iterations is beneficial to the accuracy of the solution. The recursive version of the elimination method looks promising and is given some more attention in the next section.

6.2 Initial Brownian motion

Consider initial velocity profile (B) with scaling $\omega = 10$. In the following experiments we will let the diffusion be of a few orders less than in the previous section: $\mu = 3 \times 10^{-5}$. To preserve the Péclet criterion (4.6), we need to double the number of grid-points: $N = 2^{12}$. The time-step remains unchanged, since with $\Delta t = 1.25 \times 10^{-3}$ we are now well within the stability requirement (4.22).

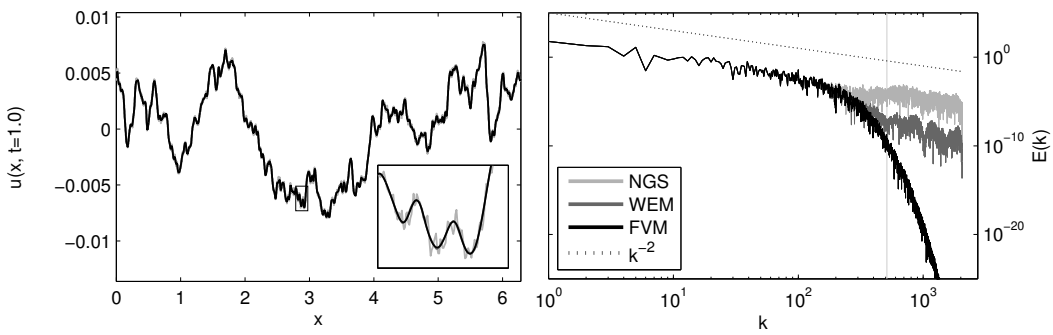


Figure 6.3: Solution (left) and energy spectrum (right), using initial velocity (B) and grid-switching of type (I) at time $t = 1$. Grid-switching parameters: $s_f = 10$, $s_c = 200$. Scaling of initial velocity: $\omega = 10$. Diffusion: $\mu = 3 \times 10^{-5}$, number of grid points: $N = 2^{12}$, time-step: $\Delta t = 1.25 \times 10^{-3}$, partition size: $p = 4$. Separation line at $N/(2p) = 512$.

Figure 6.3 compares the numerical solutions given by NGS and WEM to the fine FVM solution. Next to the separation line the power-law k^{-2} is shown (cf. Section 5). Again we observe that NGS performs worse than WEM. However, both methods now appear to separate from the FVM spectrum before the separation line is reached, indicating erroneous coarse calculations. Although for WEM this effect seems to be quite weak. The solution of NGS on the left side of Figure 6.3 shows how the non-smooth initial condition persists, again showing the inherent *out of sync* behavior.

6.2.1 Convergence aspects

For the remaining part of the numerical experiments we will consider grid-switching of type (III) and observe whether the solutions of NGS, WEM and RWEM converge to the solution of the fine FVM when the grid-switching has finished. To this end we define a *compensation spectrum*. For instance, let $\hat{u}_k^{\mathrm{WEM}}$ and $\hat{u}_k^{\mathrm{FVM}}$ be the k-th Fourier components of the WEM and FVM solutions respectively[8]. The compensation spectrum will be given by[9]

$$C_k = \frac{\hat{u}_k^{\mathrm{WEM}}}{\hat{u}_k^{\mathrm{FVM}}}, \quad \text{for} \quad k = 1, 2, \ldots, N/2 - 1.$$

For convergence we should require $C_k \approx 1$ for all k. However, since $\hat{u}_k^{\mathrm{FVM}}$ lies within the range of machine precision for some wavenumbers, these corresponding C_k will have to be ignored. The convergence of NGS and WEM is shown in Figure 6.4. First a grid-switching of type (III) is performed with $s_c = 200$ and $n_s = 5$. Then, starting at $t \approx 2.5$, a fine time-stepping is performed until we are satisfied with the convergence.

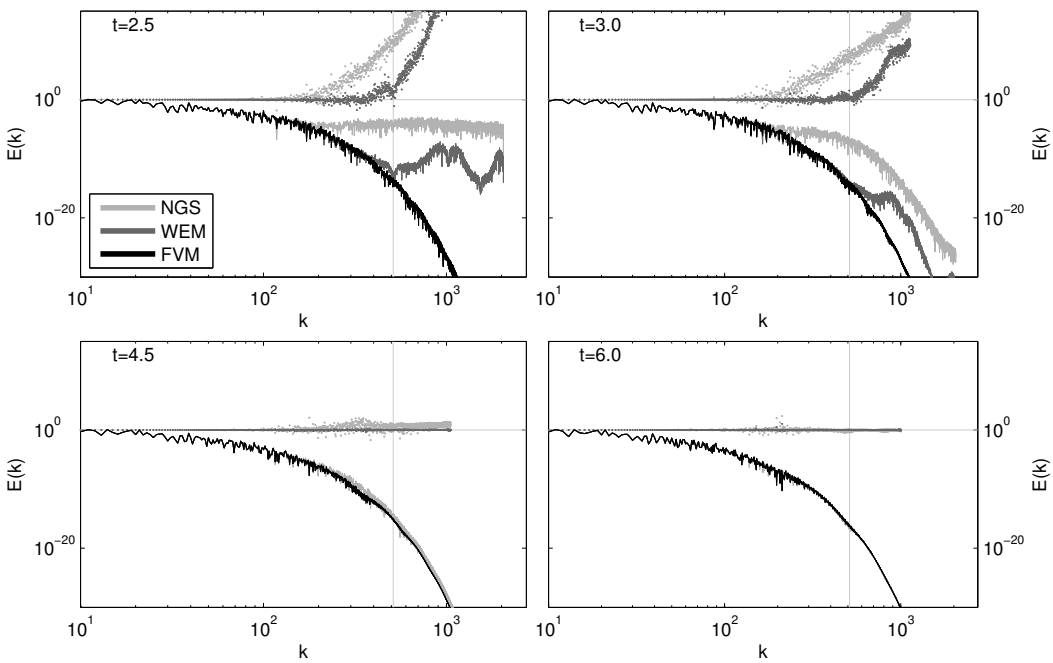


Figure 6.4: Energy spectrum using initial velocity (B) and grid-switching of type (III) at times $t = 2.5, 3.0, 4.5, 6.0$. Grid-switching parameters: $s_c = 200$, $n_s = 5$. The scatter plot shows the compensation spectrum of NGS and WEM. For the remaining parameters, see Figure 6.3.

Both the normal and compensation spectra in Figure 6.4 show how the WEM spectrum separates more or less at the expected location, whereas NGS separates at much lower wavenumbers. Moreover, at $t = 6$ we see that the compensation spectrum of NGS still shows some scattering. At this

[8]For more on Fourier transforms, see Appendix A.

[9]Note that we do not consider negative wavenumbers. Since the solution is real, the absolute value of the Fourier coefficients will be symmetric around $k = 0$.

point WEM appears to be fully converged. For the remaining plots we will therefore disregard NGS and, instead, compare WEM with RWEM.

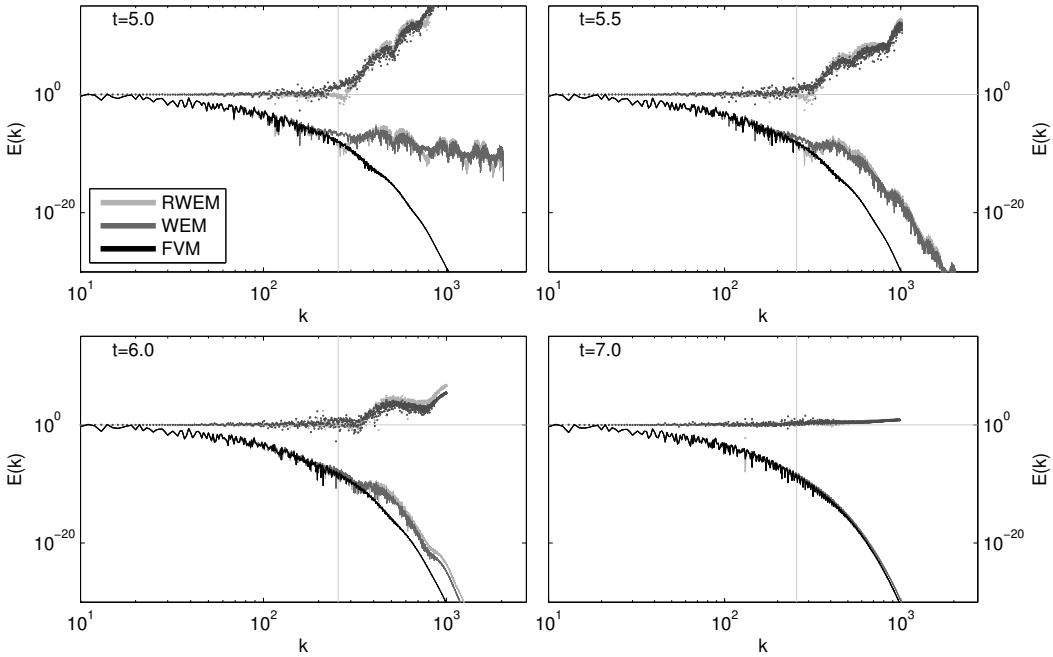


Figure 6.5: Energy spectrum using initial velocity (B) and grid-switching type (II) combined with type (III). Grid-switching parameters (WEM, $p = 8$): $s_f = 10$, $s_c = 200$, $n_s = 5$, (RWEM, $p_r = 2$): $s_f = 10$, $s_{c,2} = 2$, $s_{c,1} = 2$, $s_{c,0} = 196$, $n_s = 5$. Separation line at $N/(2p) = 256$. For the remaining parameters, see Figure 6.3.

Figure 6.5 contains results using a larger partition size: $p = 8$. Here, both WEM and RWEM are capable of giving accurate results and show similar convergence behaviors. Note that the time at which the fine time-stepping starts has moved to $t = 5$, since at coarser grids we are allowed to take larger time-steps. Next we will double the partition size such that the separation line moves to $k = 128$. Now we enter the domain of the larger scales of motion and it will become more difficult to accurately model the details in the WEM-based time-stepping, see Figure 6.6.

The experiment in Figure 6.6 shows that WEM is unable to achieve an accurate solution when $p = 16$. Recall that if WEM uses a 16 times coarser grid it assumes that, in each partition, the 15 details are stationary. This assumption appears to become problematic and it is the Newton-Raphson iteration in WEM that experiences a breakdown[10]. As a result, WEM will converge to a different, faulty solution. This is particularly well visible in the compensation spectrum, which converges to a downward curve.

On the other hand, the recursive adaptation, RWEM, performs relatively well. Although the spectrum seems quite separated at $t \approx 10$, the solution appears to have converged at $t \approx 18$. The recursion is configured such that it first transforms the solution with partition size $p_r = 4$ and performs one WEM-based time-step. After that it moves a factor 4 coarser. In RWEM, the details eliminated at the deepest level are variations of averages at a higher level, instead of variations in the actual solution. With this approach the information on the velocities of the details is distributed more evenly, thereby circumventing the ill behavior of standard WEM.

[10]In the case of a diverging Newton-Raphson iteration the implementation of WEM is designed to abort and revert to the strategy used in NGS. That is, WEM will restore the 'old' details.

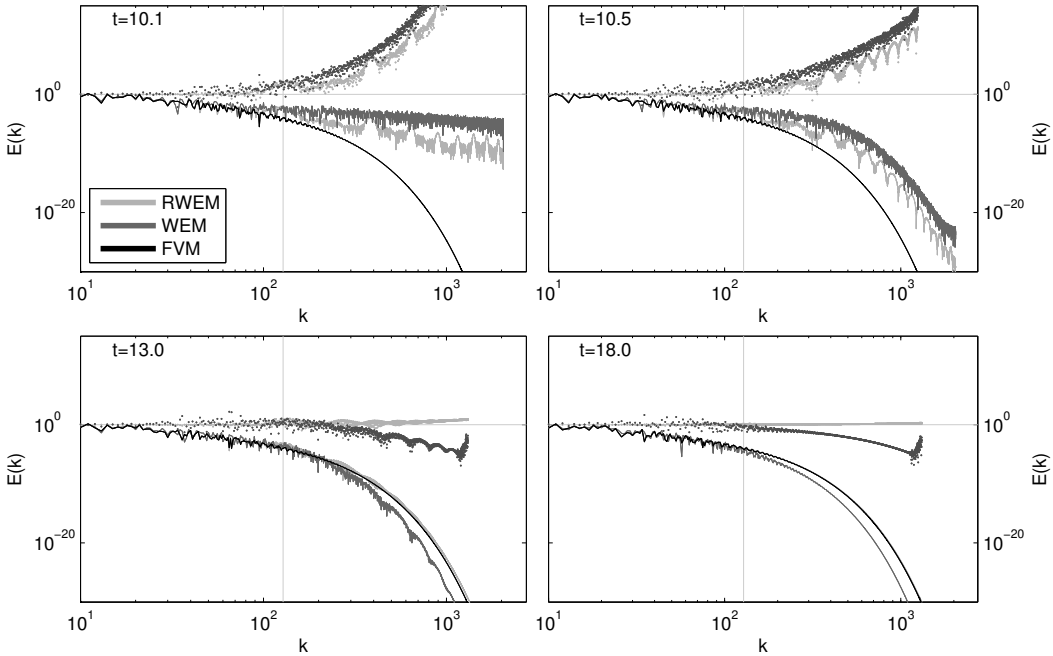


Figure 6.6: Similar configuration as in Figure 6.5, but with partition size $p = 16$. Grid-switching parameters (WEM, $p = 16$): $s_f = 10$, $s_c = 200$, $n_s = 5$, (RWEM, $p_r = 4$): $s_f = 10$, $s_{c,1} = 2$, $s_{c,0} = 198$, $n_s = 5$. Separation line at $N/(2p) = 128$. For the remaining parameters, see Figure 6.3.

μ	NGS	WEM	RWEM ($p_r = 2$)
5×10^{-5}	3.00 s	1.93 s	1.50 s
4×10^{-5}	3.24 s	2.16 s	1.84 s
3×10^{-5}	3.71 s	2.23 s	1.93 s

Table 1: Elapse (model) time between the return to the fine grid and convergence. Partition size $p = 8$.

μ	NGS	WEM	RWEM ($p_r = 2$)
5×10^{-5}	4.26 s	n/a	2.30 s
4×10^{-5}	4.75 s	n/a	2.76 s
3×10^{-5}	4.83 s	n/a	4.05 s

Table 2: Elapsed (model) time between the return to the fine grid and convergence. Partition size $p = 16$.

To conclude this section we present a few convergence timings, see Tables 1 and 2. In these experiments we slightly weaken the requirement for convergence and are satisfied whenever

$$\max_k |C_k - 1| < \texttt{tol},$$

where we set the tolerance $\texttt{tol} = 10$, as an arbitrary measure of *closeness*[11]. Note that the elapsed times do not correspond to CPU timings, but to the elapsed time in the model. Both tables show the trends we have seen earlier in the figures, but now for varying degrees of diffusion. A shorter time implies that the solution on the coarse grid is a better approximation to the fine FVM solution. As expected, WEM does not converge for $p = 16$. To obtain a level of coarseness corresponding to $p = 8$ or $p = 16$, RWEM is configured to use $p_r = 2$. It is evident that, with this configuration, RWEM achieves the best approximation in each experiment.

[11] A smaller tolerance gives longer convergence times, but the results will show the same general trend.

7 Concluding remarks

In this study we use wavelet decomposition techniques to identify and model the smallest details in the numerical solution of the non-linear Burgers' equation. Due to this non-linearity it is impossible to naively omit the small variations, when performing calculations on a relatively coarse grid. Using local transformations with basis functions from the Haar wavelet family, it turns out that a reliable elimination of the smallest variations can be made, preserving many favorable properties of the underlying discretization.

When the transformations are applied to sufficiently local partitions, the wavelet based elimination method performs very well, compared to more naive strategies. In several numerical experiments, however, we find that letting a transformation act on a large partition has an adverse effect on the accuracy of the approximation. To be able to move from a fine to a very coarse grid, we find that it is critical to do this in a gradual, nested manner. Instead of using the elimination method with a large partition size, the use of small partitions combined with the recursive elimination method is much preferred.

It is observed that the Péclet criterion, explained in Section 4, propagates as a necessary condition throughout the subsequent proofs of the numerical properties of WEM. This guarantee of smoothness also appears to be vital for the convergence of the Newton-Raphson return iteration, explained in Section 3.4. Although the return iteration may suffer from a breakdown (see Figure 6.6), this only occurs when partition sizes are chosen unreasonably large. The ill behavior for large partition sizes may be circumvented by improvements in the robustness of the return iteration. Yet, again it seems to be wiser to maintain small partition sizes and move recursively to the desired level of coarseness.

A few remarks are in order. For instance, the Picard linearizations are less accurate, but more robust than the Newton linearizations. It is therefore possible that some of the results in Section 4 are too optimistic. Thus, we should articulate that the theoretical results derived from the Picard linearizations serve merely as indications for the behavior of the Newton linearizations. Perhaps only for moderate partition sizes do they actually correspond.

Furthermore, throughout this research it has been assumed that coarse calculations are less involved than fine ones and therefore faster. However, with the large amount of transformations and linearizations, WEM may generate a considerable overhead, diminishing its attractiveness. In the main body of this report we have chosen not to explore efficient implementations of WEM. However, in Appendix C we will give a few hints which may lead in the right direction.

Future research

This study lacks a theoretical basis for the recursive adaptation of WEM. Theoretical aspects concerning the stability and accuracy of this recursive method are highly desirable. Furthermore, instead of letting this recursion be a combination of both NGS and WEM, it is advised to find a purely WEM-based recursion. That is, instead of restricting and scaling the coarse elimination-based solution, it should be possible to apply a nested partitioning and reordering directly to Equation (3.31). Within the original averages one will acquire a nested set of averages and details, in which again the details may be eliminated. Due to such repeated use of the wavelet partitioning and reordering, it is expected that the favorable properties, shown in Section 4, will be inherited by the recursively obtained coarse solutions.

As explained, it is the wavelet transform and its locality that enables a reliable elimination. It follows that we run into problems when increasing the partition size. Suppose that we would consider not one but two or three spatial dimensions. One can imagine that, with more spatial freedom to increase a partition, negative effects due to non-locality will be postponed. In other words, a two-dimensional partition will be more local than a one-dimensional partition having the same number of elements. It is therefore expected that the wavelet-based elimination methods will perform really well for problems with more spatial dimensions. A possible partitioning of a two-dimensional solution is already given in the context of image compression, see Appendix B.

A variety of other subjects may be of interest for a continuation of this study. Attractive extensions of the WEM algorithm may be obtained using adaptive partitions, either in a spatial sense or for the

recursions. Furthermore, the WEM approach could be applied to hybrid implicit-explicit (IMEX) schemes, where the non-linear term is integrated explicitly. In this explicit part the smallest details may be eliminated, enabling the use of larger time-steps.

One could distinguish between several degrees of variations within the details obtained after a wavelet transformation. As can be seen from the spectra in the numerical experiments, the smallest scales are in the dissipation range, where linear effects dominate. Together with the other linear terms, the integration of the linear details can be done implicitly, creating an IMEX-like wavelet elimination method.

Appendices

A The Fourier-Galerkin spectral method

Consider the one-dimensional forced Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \mu\frac{\partial^2 u}{\partial x^2} + f, \tag{A.1}$$

where μ denotes the diffusion coefficient. In this section we will briefly discuss the spatial discretization of (A.1) using a spectral method (for more, see for instance [3] and [19]). The scheme is based on an approximation of u with a finite series of N orthogonal basis functions:

$$u \approx u^N = \sum_{k=-N/2}^{N/2-1} u_k \phi_k, \quad \text{with} \int_\Omega \phi_k \overline{\phi}_l d\Omega = 0 \ \text{ for } k \neq l.$$

In a spectral method the basis functions ϕ_k are nonzero on the entire domain, in contrast to a finite element method, where they are defined locally.

As an approximation we will use a truncated Fourier expansion. It is therefore convenient to consider the domain $(0, 2\pi)$, since the orthogonal basis functions are then given by $\phi_k(x) = e^{ikx}$, hence

$$u^N(x,t) = \sum_{k=-N/2}^{N/2-1} \hat{u}_k(t) e^{ikx}, \quad f(x) \approx f^N(x) = \sum_{k=-N/2}^{N/2-1} \hat{f}_k e^{ikx}, \tag{A.2}$$

with coefficients $\hat{u}_k(t), \hat{f}_k$ given by the discrete Fourier transform (DFT), for instance:

$$\hat{u}_k(t) = \frac{1}{N}\sum_{j=0}^{N-1} u(x_j,t) e^{-ikx_j}, \quad x_j = \frac{2\pi j}{N}.$$

Next we give the weak formulation of (A.1) and substitute the approximations:

$$\left(\frac{\partial u^N}{\partial t}, v\right) + \left(u^N \frac{\partial u^N}{\partial x}, v\right) - \left(\mu \frac{\partial^2 u^N}{\partial x^2}, v\right) = \left(f^N, v\right) \quad \forall v \in V, \tag{A.3}$$

which requires the PDE (A.1) to be orthogonal to a test space V, with respect to the $L^2(0, 2\pi)$ inner product: $(f, v) = \int_0^{2\pi} f\overline{v} dx$. The problem (A.3) can be approximated by taking a finite amount of test functions v_k. A Galerkin method is obtained if we now use test functions from the same space as used in approximating the solution, that is, letting $v_k = \phi_k$ (see [19], section 12.4). The problem (A.3) becomes

$$\int_0^{2\pi} \frac{\partial u^N}{\partial t} e^{-ikx} dx + \int_0^{2\pi} u^N \frac{\partial u^N}{\partial x} e^{-ikx} dx - \int_0^{2\pi} \mu \frac{\partial^2 u^N}{\partial x^2} e^{-ikx} dx = \int_0^{2\pi} f^N e^{-ikx} dx. \tag{A.4}$$

Substitution of (A.2) in the linear terms of (A.4) and using the the orthogonality of the basis functions gives:

$$\int_0^{2\pi} \frac{\partial}{\partial t}\Big[\sum_{l=-N/2}^{N/2-1} \hat{u}_l(t)e^{ilx}\Big]e^{-ikx}dx = \int_0^{2\pi} \frac{\partial \hat{u}_k(t)}{\partial t}dx = 2\pi\frac{\partial \hat{u}_k(t)}{\partial t},$$

$$\int_0^{2\pi} \mu\frac{\partial^2}{\partial x^2}\Big[\sum_{l=-N/2}^{N/2-1} \hat{u}_l(t)e^{ilx}\Big]e^{-ikx}dx = -2\pi\mu k^2\hat{u}_k(t),$$

$$\int_0^{2\pi} \Big[\sum_{l=-N/2}^{N/2-1} \hat{f}_l e^{ilx}\Big]e^{-ikx} = 2\pi\hat{f}_k,$$

and (A.4) becomes

$$\frac{\partial \hat{u}_k(t)}{\partial t} + \frac{1}{2\pi}\int_0^{2\pi} u^N\frac{\partial u^N}{\partial x}e^{-ikx}dx + \mu k^2\hat{u}_k(t) = \hat{f}_k. \tag{A.5}$$

The second term in (A.5) is the Fourier transform of $u^N\frac{\partial u^N}{\partial x}$. Writing the convective term in conservation form, we have

$$\mathcal{F}\Big[\frac{\partial}{\partial x}\left(\frac{1}{2}(u^N)^2\right)\Big](k) = \frac{1}{2\pi}\int_0^{2\pi}\frac{\partial}{\partial x}\left(\frac{1}{2}(u^N)^2\right)e^{-ikx}dx.$$

Partial integration gives:

$$\mathcal{F}\Big[\frac{\partial}{\partial x}\left(\frac{1}{2}(u^N)^2\right)\Big](k) = \frac{1}{4\pi}(u^N)^2e^{-ikx}\Big|_0^{2\pi} + ik\frac{1}{2\pi}\int_0^{2\pi}\frac{1}{2}(u^N)^2e^{-ikx}dx.$$

The first term in the right hand side vanishes, due to the periodicity of the complex exponential. Hence we obtain the relation for the Fourier transform of a derivative:

$$\mathcal{F}\Big[\frac{\partial}{\partial x}\left(\frac{1}{2}(u^N)^2\right)\Big](k) = \frac{ik}{2}\mathcal{F}\Big[(u^N)^2\Big](k).$$

From (A.2) we have

$$\begin{aligned}\mathcal{F}\left[(u^N)^2\right](k) &= \frac{1}{2\pi}\int_0^{2\pi}\left[\sum_{l=-N/2}^{N/2-1}\hat{u}_l(t)e^{ilx}\right]^2 e^{-ikx}dx,\\ &= \frac{1}{2\pi}\int_0^{2\pi}\left[\sum_{l=-N/2}^{N/2-1}\hat{u}_l(t)e^{ilx}\right]\left[\sum_{m=-N/2}^{N/2-1}\hat{u}_m(t)e^{imx}\right]e^{-ikx}dx.\end{aligned}$$

Expanding the product and calculating the integral we find that only the terms $\hat{u}_l(t)\hat{u}_m(t)$ for which $l+m=k$ remain, due to the orthogonality of the basis functions. Then these terms are given by $\hat{u}_l(t)\hat{u}_m(t) = \hat{u}_l(t)\hat{u}_{k-l}(t)$ and we obtain a convolution sum:

$$\mathcal{F}\left[(u^N)^2\right](k) = \sum_{l=-N/2}^{N/2-1}\hat{u}_l(t)\hat{u}_{k-l}(t). \tag{A.6}$$

Now, the discretization (A.5) is reduced to

$$\frac{\partial \hat{u}_k(t)}{\partial t} + \frac{ik}{2}\sum_{l=-N/2}^{N/2-1}\hat{u}_l(t)\hat{u}_{k-l}(t) + \mu k^2\hat{u}_k(t) = \hat{f}_k. \tag{A.7}$$

Equation (A.7) is a semi-discretization that can be solved using a suitable ODE solver.

Remark. Calculating the convolution sum (A.6) for every k is computationally expensive. We should therefore use the convolution theorem, which enables us to compute two FFT's and a pointwise product instead. This is called a *pseudo-spectral* method, since we are now performing a part of the calculations in the physical domain, see [3] for more on this approach.

B The Haar wavelet and image compression

In the report we try to reduce information in the numerical solution of a PDE, without losing its coherent structures. To this end, we use eliminations within a wavelet-based partitioning of the solution. Similar wavelet transforms are widely used in the field of image compression. In this section we describe a method that drastically reduces the storage requirements of an image, using the familiar Haar wavelet partitioning methods.

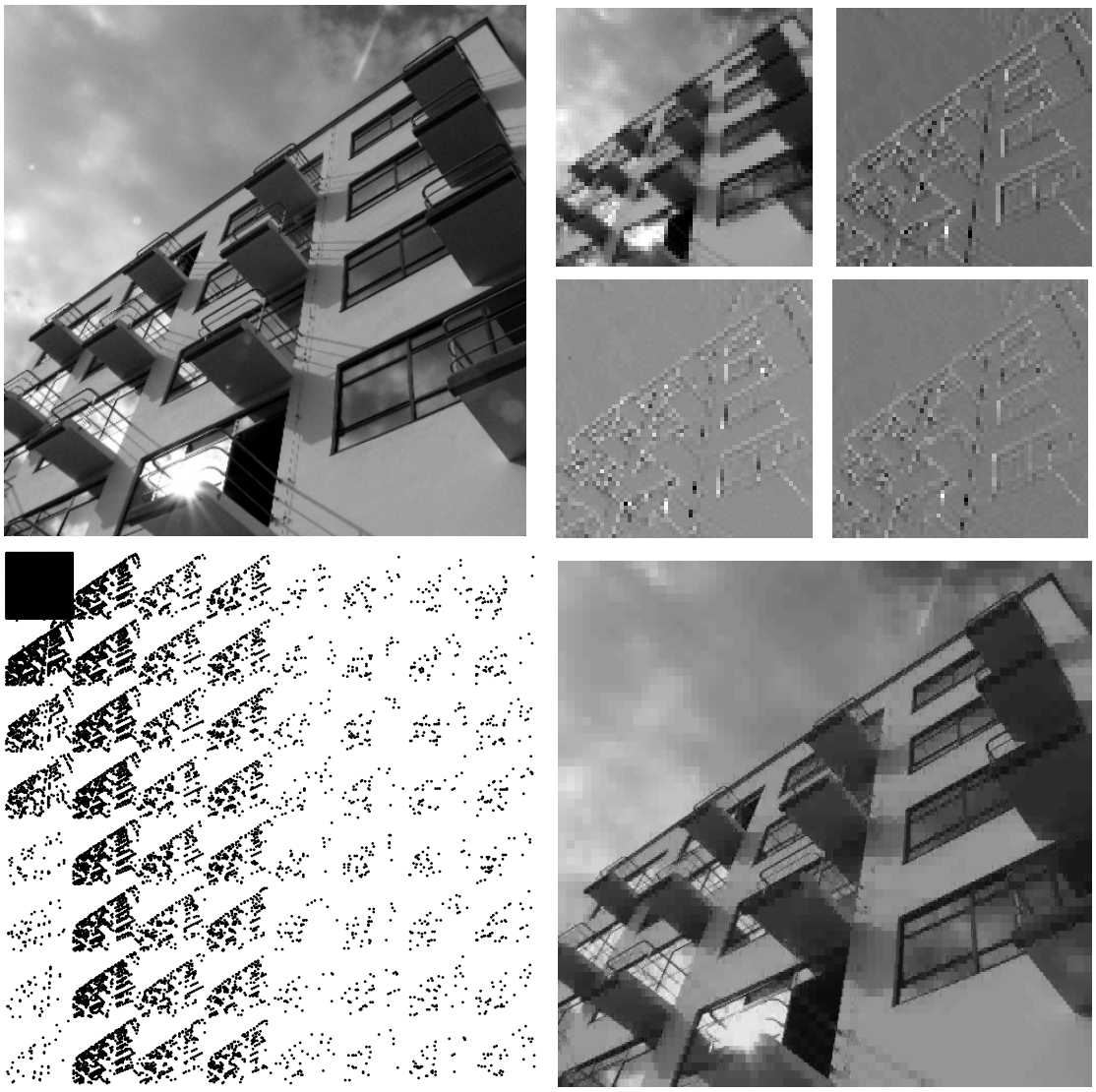


Figure B.1: Wavelet-based image compression using a gray-scale image of the famous balconies of the Bauhaus in Dessau, Germany. The gray-scale-values are in the range $[1, 256]$, the compression threshold is set at 50. The number of non-zeros for the storage of the original (top-left) image is 262144 versus 14372 in the compressed image (bottom-right), a compression ratio of ± 18.24. The average and first three variations within a block partitioning of size $p \times p = 8 \times 8$ are given in the top-right image. The compressed image can be stored as visualized in the bottom-left image.

The gray-scale image shown in the top-left of Figure B.1 can be represented as a matrix, containing elements in the range $[1, 256]$. We perform a block partitioning within this matrix yielding blocks of size $p \times p$. Each of these blocks can be reshaped into a vector of size p^2. To this vector a wavelet transform can be applied, giving one *average* and $p^2 - 1$ *details*. The average and the first three details

of each block are combined in a single picture and shown in the top-right image of Figure B.1.

In Figure B.1 the size of the block is chosen with $p = 8$. Then, the entire picture can be stored in 64 blocks, with one block containing the averages and 63 blocks containing the various details. Now, the number of non-zeros in this matrix determines the storage requirements of the picture. A straightforward choice of reducing the number of non-zeros is omitting the smallest details, i.e. defining a threshold value. Any value in the details that is smaller than this threshold in an absolute sense, will be set to zero. The result is a sparse matrix, for which the storage requirements are considerably reduced. In the bottom-left of Figure B.1 the result with a threshold of 50 is shown. Transforming the reduced details and averages back into a complete picture gives the image in the bottom-right of B.1.

In the compression of the above image, we see that it is possible to simply omit a large part of the details, yet maintain a coherent picture. This is effectively what happens when we use these techniques to speed up the numerical solution of a PDE, although the elimination method actually preserves the details to a certain degree. The close resemblance between the compressed and the original image in Figure B.1 gives an indication of the possible success of the wavelet elimination method.

Remark. The numerical solution of a PDE with two spatial dimensions can be represented as a matrix. However, it is more convenient to apply a differential operator to a solution *vector*. Thus, often the solution matrix is reshaped into a large vector. This reshaping is usually performed column-wise, but when using a wavelet partitioning it is advised to perform a few intermediate steps to maintain the solution's 'locality'.

The image compression technique hints that one should first perform a block partitioning. Then, each block can be reshaped into a vector and be subjected to a wavelet transform. Finally, the result of all wavelet transforms should be put in a single large vector, to which the (transformed) differential operators can be applied.

C An efficient implementation of the bilinear form $A\left\langle B\mathbf{x}, C\mathbf{y}\right\rangle$

A recurring term in the wavelet elimination equations is the bilinear form $A\left\langle B\mathbf{x}, C\mathbf{y}\right\rangle$, where the matrices A, B, C are sparse and stationary in time. The obvious approach to calculate this bilinear form is first resolving the inner matrix-vector products (MV's), then calculate the bilinear form and finally perform another MV with the result. With a few improvements we can significantly reduce the overhead this calculation may generate.

Consider the columns of the matrices $B \in \mathbb{R}^{m\times k}, C \in \mathbb{R}^{m\times l}$:

$$B = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k], \quad C = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l].$$

We can expand the MV's inside the bilinear form in terms of linear combinations of the columns:

$$\begin{aligned}\left\langle B\mathbf{x}, C\mathbf{y}\right\rangle &= \left\langle \sum_{i=1}^{k} \mathbf{b}_i x_i, \sum_{j=1}^{l} \mathbf{c}_j y_j \right\rangle \\ &= \left\langle \mathbf{b}_1 x_1, \mathbf{c}_1 y_1\right\rangle + \left\langle \mathbf{b}_1 x_1, \mathbf{c}_2 y_2\right\rangle + \ldots + \left\langle \mathbf{b}_n x_n, \mathbf{c}_n y_n\right\rangle \\ &= \sum_{i=0}^{k}\sum_{j=0}^{l} x_i y_j \left\langle \mathbf{b}_i, \mathbf{c}_j\right\rangle . \end{aligned} \tag{C.1}$$

Equation (C.1) requires the pointwise combination of every column vector in B and C. If we let a vector $\mathbf{z}$ contain the combinations $x_i y_j$ and a matrix M the corresponding combinations of the columns $\left\langle \mathbf{b}_i, \mathbf{c}_j\right\rangle$, then we can rewrite (C.1) in the form of a matrix-vector product:

$$\sum_{i=0}^{k}\sum_{j=0}^{l} x_i y_j \left\langle \mathbf{b}_i, \mathbf{c}_j\right\rangle = M\mathbf{z}.$$

Due to the sparsity of both B and C, the majority of the products $\left\langle \mathbf{b}_i, \mathbf{c}_j \right\rangle$ will be zero. As a result, the matrix M will be highly sparse, containing many columns that are completely zero. In order to construct a matrix M without these superfluous columns, we need to locate the *overlap* between any $\mathbf{b}_i$ and $\mathbf{c}_j$.

The proposed strategy is given Algorithm C.1:

input: $B \in \mathbb{R}^{m\times k}$, $C \in \mathbb{R}^{m\times l}$, $\mathbf{x} \in \mathbb{R}^k$, $\mathbf{y} \in \mathbb{R}^l$,
output: $\langle B\mathbf{x}, C\mathbf{y}\rangle$.

..... *Initialization phase*

1: Allocate a matrix M and an indexing set $\mathcal{I}$.
2: **for** $i = 1, 2, \ldots, k$ **do**
3: Let $\mathbf{s}$ contain the indices of the nonzero entries in the i-th column of B.
4: Let the indices in $\mathbf{s}$ determine a subset of the rows in $C \rightarrow C_{\mathbf{s}}$.
5: Within $C_{\mathbf{s}}$, locate the indices, $\mathbf{t}$, of the columns containing nonzero elements.
6: **for all** $j \in \mathbf{t}$ **do**
7: Store the vector $\left\langle \mathbf{b}_i, \mathbf{c}_j \right\rangle$ in M.
8: Store the combination (i, j) in $\mathcal{I}$.
9: **end for**
10: **end for**

..... *Calculation phase*

11: Use $\mathcal{I}$ to create a large vector $\mathbf{z}$, containing the combinations $x_i y_j$, corresponding to nonzero $\left\langle \mathbf{b}_i, \mathbf{c}_j \right\rangle$.
12: Calculate $\langle B\mathbf{x}, C\mathbf{y}\rangle = M\mathbf{z}$.

Algorithm C.1: Efficient calculation of the bilinear form $\langle B\mathbf{x}, C\mathbf{y}\rangle$.

To complete the calculation, a sparse MV with A and the vector $M\mathbf{z} \in \mathbb{R}^m$ needs to be performed. With the procedure in Algorithm C.1, we can move the better part of the calculation to the initialization phase, which can be performed before starting the time-stepping. Within the time-stepper, only the instructions given in lines 11 and 12 are required to obtain the result. This effectively reduces the bilinear form to a single sparse matrix vector product, with a slightly larger matrix.

Remark. The WEM equations contain several of these bilinear forms. To use the above approach, it is necessary to initialize the nonzero combinations of the matrices involved in each bilinear form. Then, within the time-stepper, only a small number of operations remains.

Remark. Moving the matrix A inside the expansion and the calculation of overlapping combinations leads to an extra reduction in computational effort:

$$A\langle B\mathbf{x}, C\mathbf{y}\rangle = \sum_{i=0}^{k}\sum_{j=0}^{l} x_i y_j A \left\langle \mathbf{b}_i, \mathbf{c}_j \right\rangle = M\mathbf{z}.$$

References

[1] P. N. Brown and A. C. Hindmarsh. Matrix-free methods for stiff systems of ODEs. *SIAM J. Numer. Anal.*, 23(3):610–638, 1986.

[2] J. M. Burgers. *The Non-Linear Diffusion Equation: Asymptotic Solutions and Statistical Problems*. Lecture series. Springer, 1974.

[3] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods*. Scientific Computation. Springer-Verlag, Berlin, 2006. Fundamentals in single domains.

[4] A. Chekhlov and V. Yakhot. Kolmogorov turbulence in a random-force-driven burgers equation. *Physical Review E*, 51(4):R2739, 1995.

[5] L. Collatz. Aufgaben monotoner Art. *Arch. Math.*, 3:366–376, 1952.

[6] B. M. DeBlois. Linearizing convection terms in the Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 143(3-4):289–297, 1997.

[7] P. F. Dubois, A. Greenbaum, and G. H. Rodrigue. Approximating the inverse of a matrix for use in iterative algorithms on vector processors. *Computing*, 22(3):257–268, 1979.

[8] U. Frisch. *The legacy of A. N. Kolmogorov*. Cambridge University Press Cambridge, Cambridge, England, 1995.

[9] U. Frisch and J. Bec. "Burgulence". In *Turbulence: nouveaux aspects/New trends in turbulence (Les Houches, 2000)*, pages 341–383. EDP Sci., Les Ulis, 2001.

[10] C. W. Gear and Y. Saad. Iterative solution of linear equations in ODE codes. *SIAM J. Sci. Statist. Comput.*, 4(4):583–601, 1983.

[11] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, third edition, 1996.

[12] D. F. Griffiths and D. J. Higham. *Numerical methods for ordinary differential equations*. Springer Undergraduate Mathematics Series. Springer-Verlag London, Ltd., London, 2010. Initial value problems.

[13] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Math. Ann.*, 69(3):331–371, 1910.

[14] W. Hundsdorfer and J. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2003.

[15] F. J. Koerts. *Elimination based hybrid methods for solving time-dependent PDEs.*, 2011. Bachelor thesis, University of Groningen.

[16] K. W. Morton and D. F. Mayers. *Numerical solution of partial differential equations*. Cambridge University Press, Cambridge, second edition, 2005. An introduction.

[17] R. Nguyen van yen, M. Farge, D. Kolomenskiy, K. Schneider, and N. Kingsbury. Wavelets meet Burgulence: CVS-filtered Burgers equation. *Phys. D*, 237(14-17):2151–2157, 2008.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes*. Cambridge University Press, Cambridge, third edition, 2007. The art of scientific computing.

[19] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, second edition, 2007.

[20] D. Ruelle and F. Takens. On the nature of turbulence. *Comm. Math. Phys.*, 20:167–192, 1971.

[21] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[22] Z. S. She, E. Aurell, and U. Frisch. The inviscid Burgers equation with initial data of Brownian type. *Comm. Math. Phys.*, 148(3):623–641, 1992.

[23] R. S. Varga. *Matrix iterative analysis*, volume 27 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, expanded edition, 2000.

[24] A. E. P. Veldman. *Computational Fluid Dynamics*, 2013. Lecture Notes, University of Groningen.

[25] M. Vergassola, B. Dubrulle, U. Frisch, and A. Noullez. Burgers' equation, devil's staircases and the mass distribution for large-scale structures. *Astronomy and Astrophysics*, 289:325–356, 1994.

[26] P. Wesseling. *An introduction to multigrid methods*. Pure and Applied Mathematics (New York). John Wiley & Sons, Ltd., Chichester, 1992.

[27] P. Wesseling. *Principles of computational fluid dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2001.

[28] F. W. Wubs and E. D. de Goede. An explicit-implicit method for a class of time-dependent partial differential equations. *Appl. Numer. Math.*, 9(2):157–181, 1992.