Forschungszentrum Karlsruhe

Technik und Umwelt

Wissenschaftliche Berichte

FZKA 6670

Application of Wavelets to Predict Solid Fuel Combustion

B. Mück, B. Peters

Institut für Kern- und Energietechnik

Programm Nachhaltigkeit, Energie- und Umwelttechnik

Forschungszentrum Karlsruhe GmbH, Karlsruhe
2001

Impressum der Print-Ausgabe:

Als Manuskript gedruckt

Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe

Mitglied der Hermann von Helmholtz-Gemeinschaft
Deutscher Forschungszentren (HGF)

ISSN 0947-8620

Zusammenfassung

Die Verwendung von Wavelets zur Berechnung der Feststoffverbrennung

Ein relativ neues Forschungsgebiet ist die Verwendung von Wavelets zur Lösung partieller Differentialgleichungen, die in verschiedenen Gebieten der Physik und Ingenieurswissenschaften zu lösen sind. In dieser Arbeit werden Wavelets zur Diskretisierung partieller Differentialgleichungen verwendet. Neben einer kleinen Einführung in die Welt der Wavelets wird der von Vasilyev, Paolucci and Sen (1995) vorgeschlagene Wavelet Kollokationsalgorithmus beschrieben. Die grundlegende Idee der Multiskalenanalyse ist die, daß eine Funktion als Linearkombination von Wavelets unterschiedlicher Skala an verschiedenen Orten approximiert werden kann. Eine Adaption kann erreicht werden, indem nur Wavelets verwendet werden deren Koeffizient größer als ein wählbares Limit ist. Diese Eigenschaft erlaubt eine lokale Gitterverfeinerung um beliebig kleine Skalen aufzulösen, ohne die Zahl der Gitterpunkte drastisch zu erhöhen. Die Leistungsfähigkeit der Methode steile Gradienten aufzulösen wird anhand einiger Beispiele demonstriert.

Abstract

The application of wavelets to the solution of partial differential equations arising in different areas of physics and engineering is a new field of research. In this work partial differential equations are discretised using wavelets as basis functions. Beside a short introduction to wavelets the wavelet collocation algorithm developed by Vasilyev et al. (1995) is described. The basic idea behind the multilevel approximation is that a function can be approximated as a linear combination of wavelets having different scales and locations. Adaption is achieved by retaining only those wavelets whose coefficients are greater than a given threshold. This property allows local grid refinement up to an arbitrary small scale without a drastic increase of collocation points. The capabilities of the method are demonstrated with several examples which show the advantages of the method in resolving locally steep gradients.

Contents

1 Introduction

The theory and application of wavelets, developed over the last fifteen years, has generated a tremendous interest in many areas of research. Few other theoretical developments in mathematical sciences have been applied to such diverse fields of disciplines. Most applications of wavelets have been focused on analyzing data and using wavelets as a tool for data compression or for image processing.

The application of wavelets to the solution of partial differential equations (PDE) arising in different areas of physics and engineering is a new field of research. Partial differential equations are discretised using wavelets as basis functions. Especially in the field of turbulence and combustion research the method seems promising. In Griebel and Koster (2000) an adaptive wavelet-based solver for the unsteady Navier-Stokes equations is described. Vasilyev and Bushe (1998) use wavelets for the direct numerical simulation of non-premixed turbulent flames. A survey of the usage of wavelets in solving PDE's is given in Vasilyev, Yuen and Paolucci (1997).

In many physical phenomena fast transitions and strong gradients appear in the solution e.g. shock wave formation in compressible gas or combustion fronts in flames. A characteristic feature is that the complex behavior occurs in a small region of space and is possibly intermittent in time. This makes a numerical resolution difficult using adaptive finite differences, spectral methods or finite volumes. The numerical grid has to be adapted following several criteria. Sometimes the grid adaption needs more computing time than the solution of the equations. In such cases the use of wavelets can be advantageous.

Recently Vasilyev et al. (1995) have developed a dynamically adaptive multilevel wavelet collocation algorithm for partial differential equations in multiple dimensions. The basic idea behind the multilevel approximation is that a function can be approximated as a linear combination of wavelets having different scales and locations. Adaption is achieved by retaining only those wavelets whose coefficients are greater than a given threshold. This property allows local grid refinement up to an arbitrary small scale without a drastic increase of collocation points. Therefore high resolution computations are carried out only in those regions where the sharp transients occur.

The algorithm is stable, robust and very accurate. It handles general boundary conditions. The method naturally contains the adaption of collocation points during the solution by using only one simple criteria for the coefficients. At the the same time by this criteria the accuracy of the algorithm is fixed.

In the following the method proposed by Vasilyev is described. Some new examples presented in this report show the advantages of the method applied to problems in combustion of a packed bed.

2 Methodology

In this chapter the general idea behind the wavelets is illustrated. We shall introduce and explain the terms scaling-function, wavelets and multiresolution analysis. Our aim is to provide a practical tool, therefore we will not go to much into mathematical details. For comprehensive proofs and discussions the reader is refered to Daubechies (1992) , Kaiser (1994) or Chui (1992). Also recommendable are the software wavelet toolkits of Mathematica He (1996) or Matlab.

2.1 Wavelets are Small Waves

To make a long story short, wavelets $\Psi(t)$ oscillate and their integral yield zero:

$$\int_{-\infty}^{\infty} \Psi(t)dt = 0. \tag{2.1}$$

An additional property of wavelets is their orthogonality. Wavelets are orthogonal to themselfs, translated by multiples of their wavelength τ namely:

$$\int_{-\infty}^{\infty} \Psi(t)\Psi(t - n\tau)dt = 0, \text{ with } n \in \mathbb{Z}, z = 0, 1, 2, \cdots \tag{2.2}$$

Haar (1910) introduced a wavelet which fulfills both conditions. Figure 1 shows this wavelet.

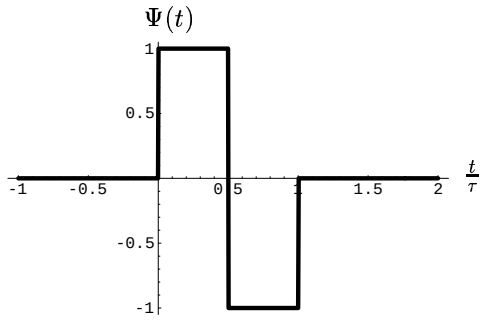

Figure 1: Haar wavelet

This wavelet has a zero integral. Now we translate the function $\Psi(t)$ by an integer number and plot both functions, see figure 2.

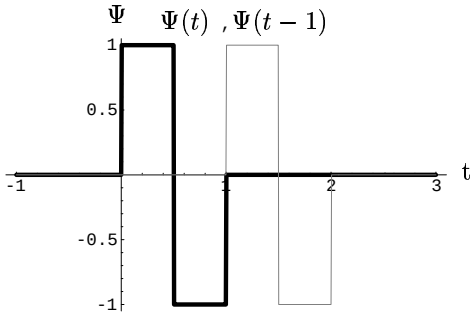

Figure 2: Haar wavelet and translated Haar wavelet are orthogonal.

It is obvious that the Haar wavelet also fulfills the orthogonality condition 2.2. For the application in numerical simulation the wavelets don't have to be orthogonal but orthogonality often provides conveniences in computations.

Wavelets can be used as basis functions in a series expansion similar to the use of sine and cosine functions in Fourier-transform. In contrast to the sine and cosine functions the wavelets are non-zero only in a bounded region, see figure 1. This fact is called 'local support' in time. The sine and cosine have a 'nonlocal support' in time. On the other hand sine and cosine function can be assigned to one single frequency, which means that they have a perfect localization in frequency space. Wavelets do not have this perfect frequency localization but they are localized in the frequency space, i.e. they have a finite support in frequency space.

The localization of wavelets in time and space is profitable for analyzing signals, especially those where frequencies vary in time.

Analyzing signals in time can also be realized by doing a short-time Fourier-transform using windows in order to analyze subsequent sections of the signal. The wavelet transform gets rid of the fixed window length which has to be chosen before doing a short-time Fourier-transform. So one does not have to rely on human experience in choosing the window-length.

To understand the wavelet transform we first introduce the subject of multiresolution analysis.

2.2 Multiresolution Analysis

Multiresolution analysis refers to the simultaneous approximation of different resolutions for a given function. Any function or signal can be decomposed in a coarse approximation and finer details. The addition of all approximation levels results in the complete original signal. The distinction between smooth part and details is specified by the sampling-resolution. At a fixed sampling resolution all details below that resolution are not determinable. Now we can think of a successive increase of the sampling-resolution in specified stages. Each stage of resolution is called a level. At each higher level, finer details appear and we get a better approximation of the signal. Finally we end at the original signal when the number of resolution levels goes to infinity.

Following this heuristic description we can make the description more accurate. We can assemble the original function when we sum up all detail levels d_k to the function at a given resolution f_j. This is described in the following equation 2.3:

$$f(t) = f_j(t) + \sum_{k=j}^{\infty} d_k(t). \tag{2.3}$$

The resolution level is labeled by integers $j \in \mathbb{Z}$.

2.2.1 First Requirement

Functions with specific properties can generate a so called function space. In wavelet-theory the function spaces (e.g. the space of all real square integrable functions $L^2(\mathbb{R})$) are used to explain the multiresolution analysis.

We can look on this abstract function spaces in a similar way as to the decomposition of the single function in equation 2.3. Analogous we can view a function space as being composed of a sequence of subspaces. We can define an approximation space V_j analogous to f_j and a detail space W_k analogous to d_k. We can write the decomposition like this:

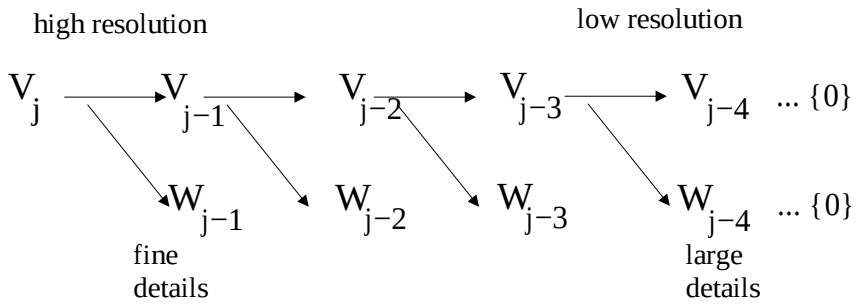

Figure 3: Function space composed of approximation spaces and detail spaces.

If we look at figure 3 we realize that any subspace $V_{j-1}, V_{j-2}, \cdots$ is included in V_j since any information in the low resolution must be included in the finer resolutions. This is the first requirement of a multiresolution analysis. In a short form it can be written as:

$$L^2(\mathbb{R}) \supset \cdots V_2 \supset V_1 \supset V_0 \supset V_{-1} \cdots \{0\}. \tag{2.4}$$

2.2.2 Second Requirement

The difference between two approximation spaces lies in the details at a given level j. This can be written as

$$V_j = V_{j-1} + W_{j-1}. \tag{2.5}$$

To fulfill this requirement information of the detail space W_{j-1} must not be included in V_{j-1}, i.e.

$$V_{j-1} \cap W_{j-1} = \{0\}. \tag{2.6}$$

This corresponds to a vanishing inner product of V_{j-1} and W_{j-1}.

$$\langle V_{j-1}(t), W_{j-1}(t) \rangle = \int_{\infty}^{-\infty} V_{j-1}(t) \cdot W_{j-1}(t) dt = 0 \tag{2.7}$$

Two spaces that satisfy this properties are called orthogonal. Together with condition 2.4 we recover that the set union $\bigcup$ of all approximation spaces gives the space of all square integrable functions,

$$\bigcup_j V_j = L^2(\mathbb{R}). \tag{2.8}$$

Since the only detail which is included in every resolution is the zero function we get for the intersection $\bigcap$ of all spaces:

$$\bigcap_j V_j = \{0\}. \tag{2.9}$$

Equations 2.8 and 2.9 are the second requirement for a multiresolution analysis.

2.2.3 Third Requirement (Dilation Invariance)

The third requirement deals with the scale of the function spaces. All the spaces V_j are scaled (dilated or squeezed) versions of one space, say e.g. V_0. If $f(t)$ contains no details with scales smaller than say e.g. $1/2^j$. The next finer scale $f(2t)$ is

obtained by squeezing $f(t)$ by a factor of 2. Then $f(2t)$ contains no details smaller than $1/2^{j+1}$. In a short form one can write:

$$af_j(t) \in V_j \Longleftrightarrow af_j(2t) \in V_{j+1}, \qquad \text{with} \quad a = 2^{\frac{j}{2}}. \tag{2.10}$$

The scaling factor a scales the amplitude of the functions spaces. Condition 2.10 is also called dilation invariance. For dilation mostly the (also called dyadic) factor 2 is used.

2.2.4 Fourth Requirement (Shift Invariance)

The fourth condition treats the translation of the function spaces used. If $f(t) \in V_0$ then also the translates $f(t-k) \in V_0$. This is valid for all spaces V_j. One can write

$$f_j(t) \in V_j \Longleftrightarrow f_j(t-k) \in V_j \qquad \text{with} \quad k \in \mathbb{Z}, k = 0, 1, 2, \cdots . \tag{2.11}$$

Condition 2.11 is also called shift invariance.

2.2.5 Fifth Requirement, Scaling Function

The final requirement states that there exists a function, called 'scaling function $\Phi(t)$' with the property that the shifted versions of the same function $\Phi(t-k)$ form an orthonormal basis of the function space V_0. The scaling function $\Phi(t)$ is orthogonal to its own translated versions, shifted by integers. The purpose of the scaling-function is to provide an orthogonal basis for the approximation spaces.

Similar to the formation of vector spaces as a linear combination of vectors we can form function spaces as a linear combination of (scaling-) functions Φ

$$V_0 = \sum_{i=1}^{i=k} b_i \Phi_i(t). \tag{2.12}$$

This last requirement as a general rule is not so easy to fulfill.

2.2.6 The Haar Scaling Function

A function space which satisfies all five properties can be built out of box-functions. We consider the space V_0 of all functions which are constant on unit intervals e.g.

$$V_0 = \begin{cases} 1 & for \;\; [k, k+1) \;\; k \in \mathbb{Z}. \\ 0 & otherwise \end{cases} \tag{2.13}$$

It is obvious that we can use 'smaller' building blocks V_j to assemble the functions in V_0

$$V_j = \begin{cases} 1 & for \;\; [\frac{k}{2^j}, \frac{k+1}{2^j}) \;\; j, k \in \mathbb{Z}, \; j = 0, 1, 2, \cdots . \\ 0 & otherwise \end{cases} \tag{2.14}$$

For dilation we choose here powers of 2. Powers of 2 are chosen for computational convenience. Clearly we can build the functions in V_0 out of functions in e.g. V_1 (or V_2 or V_3 ...) but note, not vice versa! The Haar multiresolution analysis (Haar 1910) is using this kind of scaling functions.

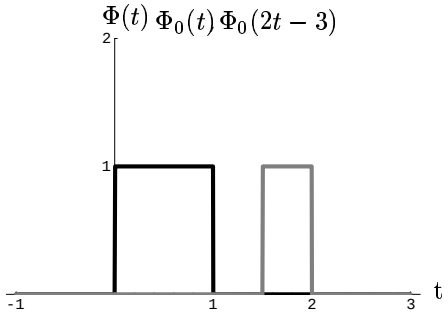

Figure 4: The Haar scaling function $\Phi_0(t)$ and $\Phi_0(2t-3) = \Phi(t-3)$.

In figure 4 we realize that the function $\Phi_0(t)$ can be built out of two blocks of $\Phi_0(2t) = \Phi_1(t)$ namely $\Phi_1(t)$ and $\Phi_1(t-1)$. $\Phi_1(t-3)$ is plotted for better visibility.

It is easy to see that the scaling functions at each level are orthogonal to themselves since there is no overlap between two of them shifted by integers. Furthermore we can see that this function-spaces are orthonormal and satisfy additional to equation 2.7 the condition:

$$\langle V_j(t-n), V_j(t-m)\rangle = \int_{\infty}^{-\infty} V_j(t-n)\cdot V_j(t-m)dt = \delta_{n,m} \quad \text{with} \quad \delta_{n,m} = \begin{cases} 1 & for\ m = n \\ 0 & otherwise \end{cases} \tag{2.15}$$

This example can be seen as the first of a whole family of multiresolution analyses called spline multiresolution analysis. The second member of this family would be the multiresolution analysis of piecewise linear functions as shown in figure 5.

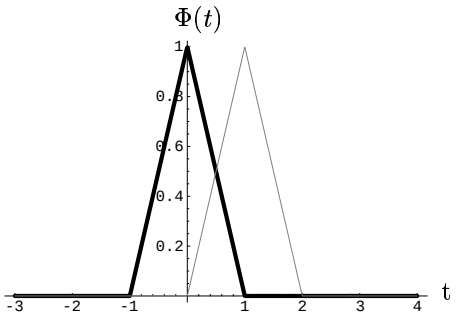

Figure 5: Piecewise linear functions $\Phi(t)$ and $\Phi(t-1)$ (non-orthogonal).

Note that the system generated by piecewise linear functions is not orthogonal, but it is possible to get an orthogonal scaling function by an orthogonalization procedure in Fourier space

$$\hat{\Phi}_{orth}(\omega) = \frac{\hat{\Phi}(\omega)}{\sqrt{\sum_{l=-\infty}^{\infty} |\hat{\Phi}(\omega + 2\pi l)|^2}}, \tag{2.16}$$

where $\hat{\Phi}(\omega) = \int_{-\infty}^{\infty} \Phi(t) e^{-i\omega t} dt$ is the Fourier transform of Φ. We get the orthogonal scaling function by a transforming back from Fourier space:

$$\Phi_{orth}(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{\Phi}(\omega)_{orth}\, e^{-i\omega t} d\omega \tag{2.17}$$

Since equation 2.16 is often a predefined procedure in wavelet packages like mathematica or matlab we will not explain it in detail. For further explanations see e.g. Chui (1992) or He (1996).

The orthogonal scaling function is shown in the following picture.

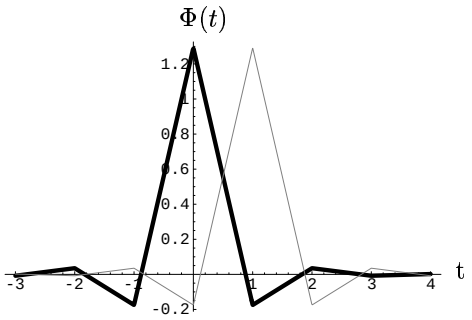

Figure 6: Piecewise linear scaling function $\Phi(t)$ and $\Phi(t-1))$(orthogonal).

Note that the system of orthogonal scaling functions is also orthonormal and satisfies equation 2.15, although this is not obvious from the figure 6.

2.3 The Dilation Equation

In this section we show the consequences of the requirements of the multiresolution analysis. We know that the approximation spaces V_j form a nested sequence (equation 2.4) to get a successively better approximation. The dilations and translations of the scaling function Φ in the space V_0 generate all basis functions for each space V_j. The dilations and translations of the scaling function are realized in the following equation

$$\Phi_{j,k}(t) = 2^{j/2}\Phi(2^j t - k), \qquad j,k \in \mathbb{Z}. \tag{2.18}$$

As mentioned before for reason of computational efficiency in most cases powers of 2 are used for dilation (binary dilation).

Since $V_0 \in V_1$, the functions in V_0 can be expressed as a linear combination of functions in V_1. The functions in V_1 describe a basis in 'coarser' subspaces (V_0, $V_{-1}\cdots$). This can be shown for the scaling function Φ_0

$$\Phi(t) = \sum_k h_k \Phi_{1,k}(t) = \sqrt{2}\sum_k h_k \Phi(2t-k) \tag{2.19}$$

This equation is known as two-scale equation, dilation-equation or refinement equation since it relates the scaling functions of two successive scales. Since the scaling functions in V_1 are orthogonal and form a basis, see section 2.2.5, we can obtain the coefficients h_k by evaluation of the inner product of Φ and Φ_1. To explain this, we can look on the analogon for the inner product in vector space, the scalar product. Any vector in $\mathbb{R}^n$ can be written as a linear combination of the orthonormal basis vectors in $\mathbb{R}^n$. To get the linear factors we compute the scalar product of the vector with the basis vectors.

In function space we have to compute the inner product to get the linear factors:

$$h_k = \langle \Phi_{1,k}, \Phi \rangle = \sqrt{2}\int_{-\infty}^{\infty} \Phi(t)\Phi(2t-k)dt. \tag{2.20}$$

If we integrate both sides of equation 2.19 we get together with $\int_{-\infty}^{\infty} \Phi(t)dt = 1$, see equation 2.15 (orthonormal), an additional condition for the coefficients

$$\sum_k h_k = \sqrt{2}\ . \tag{2.21}$$

As an example we can again take the Haar scaling function. In the following picture we plotted $\Phi_1(t)$ and $\Phi_1(t-1)$.

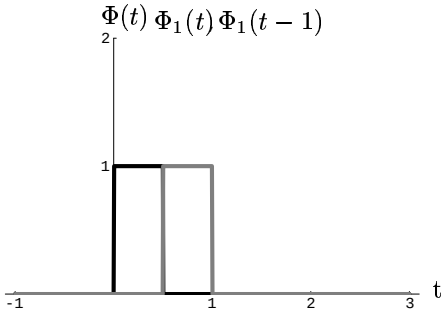

Figure 7: Haar scaling function $\Phi(2t) = \Phi_1(t)$ and $\Phi(2t-1) = \Phi_1(t-1)$.

From figure 7 we see that $\Phi(t) = \Phi(2t) + \Phi(2t-1)$. Therefore in equation 2.19 the h_k are $h_0 = h_1 = 1/\sqrt{2}$. Equation 2.21 is also satisfied. In most cases we have to compute the inner product since the result is not as obvious as in this case.

2.4 The Wavelet Equation

Now we take a closer look on the detail space W_j . Similar to the existence of a orthonormal basis in approximation space V_j, see section 2.2.5, one can show that there exists also a orthonormal basis of the detail space W_j.

The function $\Psi(t)$ which generates all these basis functions is called mother wavelet. The dilations and translations of the mother wavelet Ψ in the space W_0 generate all basis functions for the spaces W_j

$$\Psi_{j,k}(t) = 2^{j/2}\Psi(2^j t - k), \qquad j,k \in \mathbb{Z} \tag{2.22}$$

How do we get this mother wavelet $\Psi(t)$? It is now helpful to take again a look on figure 3. The mother wavelet 'lives' in the detail space W_0 and this detail space is part of the next finer approximation space V_1, i.e. $W_0 \subset V_1$. Therefore we can use a linear combination or superposition of the basis functions in V_1 to express any function in detail space W_0

$$\Psi(t) = \sum_k g_k \Phi_{1,k}(t) = \sqrt{2} \sum_k g_k\ \Phi(2t-k). \tag{2.23}$$

This equation is called the wavelet equation. It relates the mother wavelet in the space W_0, $\Psi(t)$, to the scaling function at the next finer scale V_1. The equation is similar to the dilation equation. Since the scaling function is orthogonal we can obtain the coefficients g_k by computing the inner product (once we know the wavelet function $\Psi(t)$)

$$g_k = \langle \Phi_{1,k}, \Psi \rangle = \sqrt{2} \int_{-\infty}^{\infty} \Psi(t)\Phi(2t-k)\ dt. \tag{2.24}$$

With condition $\int_{-\infty}^{\infty} \Psi(t)dt = 0$ (eqn. 2.1) we get integrating equation 2.23

$$\sum_k g_k = 0. \tag{2.25}$$

Up to now the wavelet function itself is not known, but it is possible to derive a simple relation between the coefficients of the dilation equation and the wavelet equation which enables us to draw the wavelet once we have the coefficients of the orthogonal scaling function

$$g_n = (-1)^n h_{1-n} \quad . \tag{2.26}$$

For the, somewhat longish, derivation of this relation we refer to (He 1996) or Daubechies (1992). We can take the coefficients of the Haar scaling function ($h_0 = h_1 = 1/\sqrt{2}$, with $g_1 = -h_0$ and $g_0 = h_1$) and using equation 2.23 we can draw the wavelet, see figure 8.

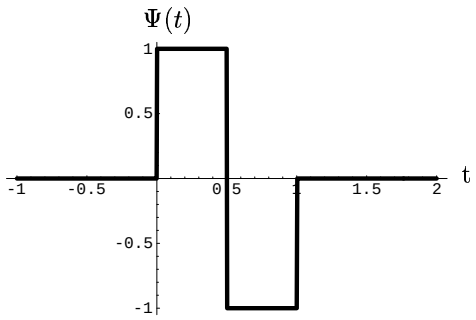

Figure 8: Haar wavelet

Now we have collected all common basics to deal with wavelets and scaling functions.

Much effort was spent in constructing wavelets with special properties for special purposes like the famous Daubechies wavelets, Shannon wavelets, Mayer wavelets and others. In the next chapter we will show some of them. They all obey the requirements of a multiresolution analysis.

2.5 Short Summary with Examples

In the previous section we saw that scaling function and wavelet functions are quite similar. The most important difference to keep in mind is that the scaling function lives in the approximation space and the wavelet in the detail space. Let us shortly summarize in the following table the most important properties of a multiresolution analysis:

scaling function $\Phi(t)$	mother wavelet $\Psi(t)$
approximation space V_j	detail space W_j
nested spaces: $V_j \subset V_{j+1}$	nested spaces: $W_j \subset V_{j+1}$
orthonormal basis: $\Phi_{j,k}(t) = 2^{j/2}\Phi(2^j t - k)$	$\Psi_{j,k}(t) = 2^{j/2}\Psi(2^j t - k)$
dilation equation: $\Phi(t) = \sqrt{2}\sum_k h_k \Phi(2t-k)$	wavelet : $\Psi(t) = \sqrt{2}\ \sum_k g_k\ \Phi(2t-k)$
$h_k = \langle \Phi_{1,k}, \Phi \rangle = \sqrt{2}\int_{-\infty}^{\infty} \Phi(t)\Phi(2t-k)dt$	$g_k = \langle \Phi_{1,k}, \Psi \rangle = \sqrt{2}\ \int_{-\infty}^{\infty} \Psi(t)\Phi(2t-k)\ dt$

relation of coefficients

$$g_n = (-1)^n h_{1-n} \qquad n = 0, 1, \cdots$$

The most common way to construct a wavelet is to start with a scaling function, which is made orthogonal in Fourier space, see equation 2.16. With the dilation equation we compute the coefficients h_k and then the wavelet coefficients g_k from equation 2.26. Now we draw the wavelet from the wavelet equation 2.23.

The following figure 9 shows how we get the mother wavelet from a piecewise constant scaling function.

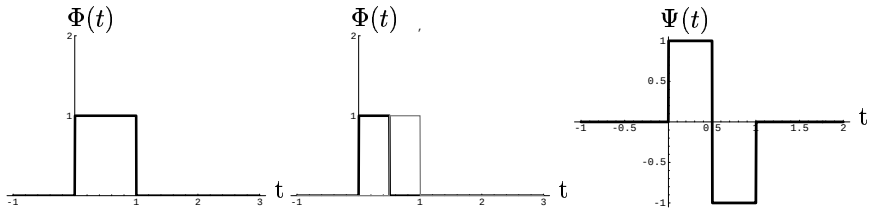

Figure 9: Scaling function $\Phi(t)$,$\Phi(2t)$ and Haar mother wavelet.

Since the scaling function is already orthogonal, we can proceed directly to the step where the scaling function of the zero-level ($j = 0$) is constructed out of scaling functions of the first level ($j = 1$). This is the essence of the dilation equation which relates two successive scales. By looking on the picture and the dilation equation we see that the coefficients must be $h_0 = h_1 = 1/\sqrt{2}$. With the coefficient relation we can now evaluate the wavelet coefficients $g_1 = -h_0$ and $g_0 = h_1$ and we can draw the wavelet.

The next picture shows the steps with a piecewise linear and quadratic scaling function.

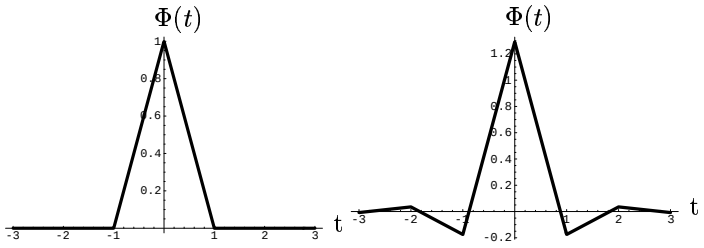

Figure 10: Linear scaling function $\Phi(t)$,orthogonal linear scaling function.

In figure 10 we see the linear and the orthogonal linear scaling function evaluated following equation 2.16.

In the next figure 11 we give an idea of the dilation equation. In this picture we take 5 wavelets of scale $j = 1$. From the inner products of $\Phi_0(t)$ and $\Phi_1(t)$, see equation 2.20 we get the coefficients h_k to reproduce the scale $j = 0$. The number of the coefficients depends on the desired accuracy, but it has an upper limit which is related to the support of the wavelet. The more coefficients the higher the accuracy to reproduce the lower level scaling functions out of the scaling functions of the higher level.

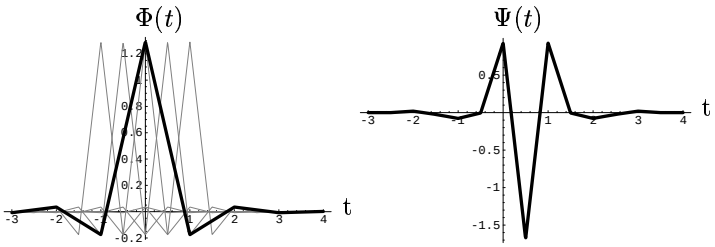

Figure 11: Linear scaling function $\Phi(2t)$, mother wavelet.

From the coefficients h_k we again evaluate the wavelet coefficients g_k and we can draw the mother wavelet, see right side of figure 11.

Figure 12 shows a piecewise quadratic scaling function and the corresponding orthogonal scaling function.

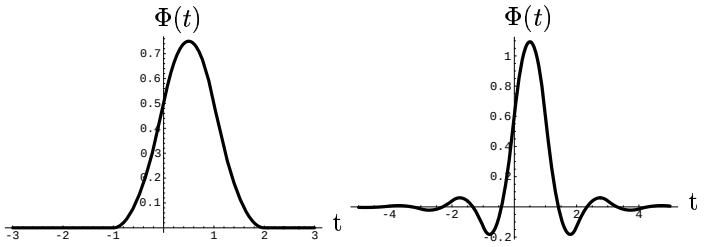

Figure 12: Quadratic scaling function $\Phi(t)$,orthogonal quadratic scaling function.

In figure 13 we show the dilation relation and the wavelet.

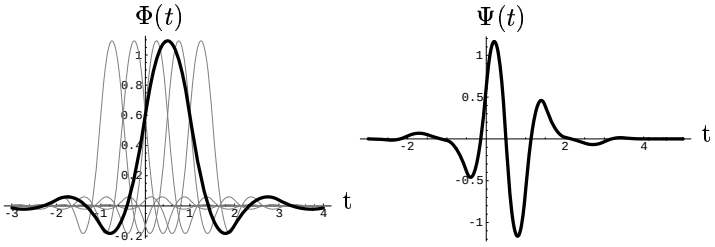

Figure 13: Dilation relation of scaling function, wavelet.

We see that the number of continuous derivations of the scaling function rises from zero (Haar wavelet) to two. These examples are the first of a family of multiresolution analyses called spline multiresolution analysis.

2.6 Wavelets with Additional Properties, Daubechies Wavelets, Coiflets

Ingrid Daubechies was the first to construct a wavelet with a specified number of vanishing moments additional to the requirements of a multiresolution analysis. A moment of a function $f(x)$ is defined as follows:

$$m_k = \int_{-\infty}^{\infty} f(x) x^k dx \qquad \text{with } k = 0, 1, 2, \cdots \tag{2.27}$$

The condition that the first N moments vanish is that the integral in equation 2.27 is zero,

$$\int_{-\infty}^{\infty} f(x)x^k dx = 0 \qquad \text{with } k = 0, 1, 2, \cdots, N-1. \tag{2.28}$$

If we have one vanishing moment, i.e. $m_0 = 0$, the function $f(x)$ is orthogonal to constant functions. With two vanishing moments ($m_{0,1} = 0$) the function is orthogonal to constant and linear functions.

This is an amazing property. As an example we look at some Daubechies wavelets with vanishing moments ($m_{k=0,1,2} = 0$):

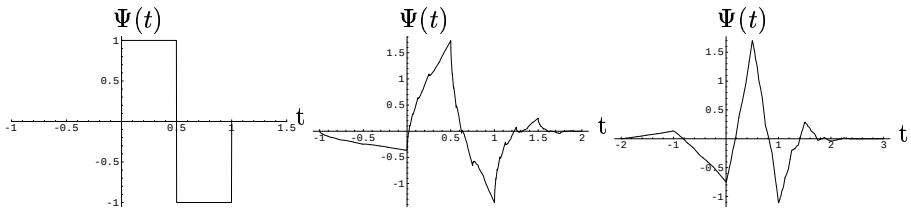

Figure 14: Daubechies wavelets with one, two and three vanishing moments.

Note that the Daubechies wavelet with one vanishing moment is the Haar wavelet, see figure 9. It fulfills the requirements of a multiresolution analysis (section 2.5) and it is orthogonal to constant functions.

The practical effect of wavelets with vanishing moments is that the information of a signal can be concentrated in a smaller amount of coefficients, i.e. we can compress the information of a signal in few coefficients. In numerical simulation application wavelets with five or six vanishing moments are commonly used to reduce the amount of collocation points needed.

For details of the derivation of the Daubechies wavelets with specified vanishing moments we refer the reader to e.g. Daubechies (1992) or any other wavelet introduction.

In the case of the Daubechies wavelets the vanishing moment condition is only valid for the wavelet and not for the scaling function. Coifman suggested that the vanishing moment condition should be valid for the scaling function $\Phi(t)$ and the wavelet $\Psi(t)$. These functions are called 'Coiflets'. Figure 15 shows a Coiflet.

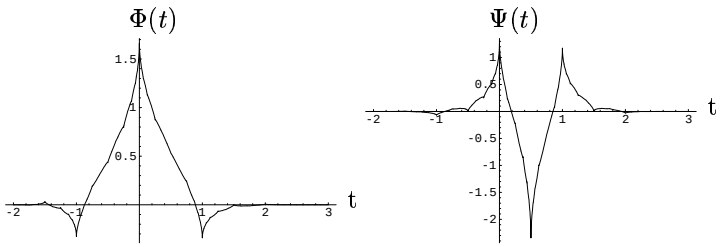

Figure 15: Coiflet scaling function and wavelet with two vanishing moments.

2.6.1 The Mexican Hat Wavelet

In many cases we can use the so called 'Mexican-hat wavelet' for the approximation. The advantage of this wavelet is its explicit analytical description. It is defined as:

$$\Psi(t) = (t^2 - 1)e^{-\frac{1}{2}t^2} \qquad \text{Mexican hat wavelet.} \tag{2.29}$$

This wavelet is one from a family called Gaussian wavelets defined by:

$$\Psi^n(t) = (-1)^n \frac{d^n}{dx^n} e^{-\frac{1}{2}t^2}. \tag{2.30}$$

Figure 16 shows the different wavelets derived from equation 2.30.

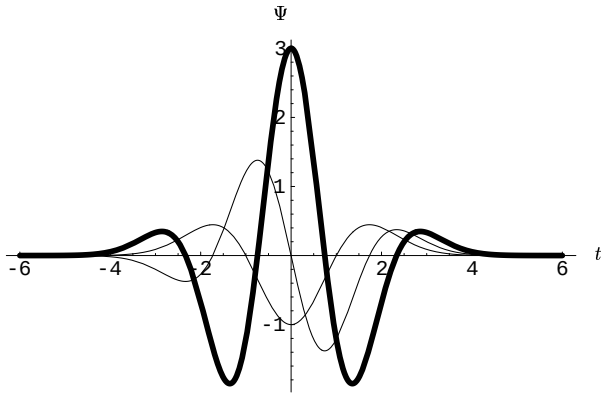

Figure 16: Mexican hat wavelets of order 2 (bold line),3 and 4.

These so called wavelets do not fulfill all the properties of a wavelet. The wavelets are not orthogonal to themselves, shifted by integers. The rest of the conditions for a multiresolution-analysis can be fulfilled.

In this context often so called 'frames' or 'wavelet-frames' are introduced to proof that the reconstruction mechanism is numerically stable, although the used wavelet is not orthogonal. The 'frames' are sets of non-orthogonal basis-functions with specified properties (we can also think of vectors) which we can nevertheless use, within some defined limits, to expand functions in function space (or vectors). The Mexican-hat wavelets constitute such a frame. For a more detailed discussion we refer to Daubechies (1992).

2.6.2 Wavelets and Autocorrelation Functions

In the last section we introduced the 'Mexican-hat' wavelet which is non-orthogonal and symmetric. The wavelets do not need to be orthogonal for our purposes. In some applications non-orthogonal symmetric wavelets are preferred because of their improved 'shift-invariance'. The lack of shift-invariance is the main disadvantage of most orthogonal wavelets, which means that the wavelet transform of a signal and of a time shifted version of the same signal are not simply shifted versions of each other. The property of shift-invariance simplifies the analysis of signals from scale to scale.

For this reason autocorrelation functions of orthogonal wavelets are used since they are also symmetric and shift-invariant, which simplifies the finding of zero-crossings. In the present work, following Vasilyev et al. (1995) the autocorrelation function of the Daubechies scaling function with five vanishing moments is used. The scaling function is shown in figure 17

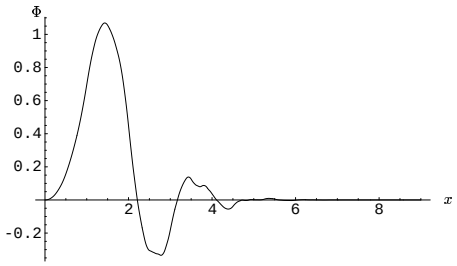

Figure 17: Scaling function of Daubechies wavelet with five vanishing moments.

In the following we show the algorithm and the basic principle. For details of the derivation we refer to Beylkin and Saito (1992).

Let $\mathbf{\Phi(x)}$ be the autocorrelation function of the used scaling function $\Phi(x)$.

$$\mathbf{\Phi}(x) = \int_{-\infty}^{\infty} \Phi(y)\Phi(y-x)dy \tag{2.31}$$

For convenience of explanation we use the variables x and y instead of the time t. The scaling function satisfies the two-scale equation (see also eqn. 2.19):

$$\Phi(x) = \sqrt{2} \sum_{k=0}^{2N-1} h_k \Phi(2x-k), \tag{2.32}$$

where N is the number of vanishing moments of the corresponding wavelet (see also section 2.6). The upper boundary $2N-1$ comes from the finite support of the scaling function, i.e $\mathbf{\Phi} = 0$ for $|x| > 2N-1$. If we substitute equation 2.32 in equation 2.31 the following recursive relation can be derived:

$$\mathbf{\Phi}(x) = \mathbf{\Phi}(2x) + \frac{1}{2} \sum_{l=1}^{N} a_{2l-1} (\mathbf{\Phi}(2x-2l+1) + \mathbf{\Phi}(2x+2l-1)) \tag{2.33}$$

where:

$$a_k = \begin{cases} 2 \sum_{l=0}^{2N-1-k} h_l h_{l+k} & \text{for } k = 1, 3, \cdots, 2N-1 \\ 0 & \text{for } k = 2, 4, \cdots, 2N-2 \end{cases} \tag{2.34}$$

The values of the autocorrelation $\mathbf{\Phi}(x)$ at the collocation points $2^{-j}k$, see equations 3.66, can be obtained recursively from :

$$\mathbf{\Phi}(2^{-j-1}k) = \mathbf{\Phi}(2^j k) + \frac{1}{2} \sum_{l=1}^{N} a_{2l-1} \left(\mathbf{\Phi}(2^{-j}k - 2l + 1) + \mathbf{\Phi}(2^{-j}k + 2l - 1) \right) \tag{2.35}$$

Note that we know a priori that in the case of orthonormal scaling functions Φ the autocorrelation $\mathbf{\Phi}(x)$ at integer-numbered positions is zero exept at $x = 0$ where the autocorrelation has the value one. We are now able to calculate the autocorrelation at any collocation point. Figure 18 shows the autocorrelation of the scaling function printed in figure 17:

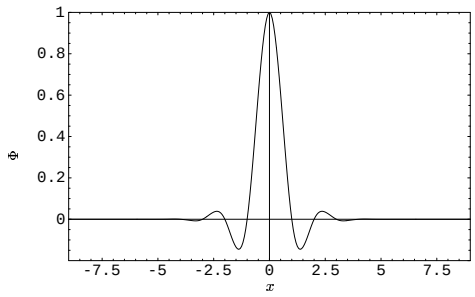

Figure 18: Autocorrelation of Daubechies Scaling function.

For later usage we additionally need the derivatives $\mathbf{\Phi}^{(n)}(x)$ of the autocorrelation function $\mathbf{\Phi}(x)$. At first we need the values at integer positions $x = k(k = 0, \pm 1, \ldots, \pm(2N-1))$ of $\mathbf{\Phi}^{(n)}(x)$ which can be obtained by solving the system of equations:

$$\mathbf{\Phi}^{(n)}(k) = 2^n \mathbf{\Phi}^{(n)}(2k) + 2^{n-1} \sum_{l=1}^{N} a_{2l-1} \left(\mathbf{\Phi}^{(n)}(2k-2l+1) + \mathbf{\Phi}^{(n)}(2k+2l-1) \right) \tag{2.36}$$

The system of equations 2.36 is linear dependent. The decrease of the rank is one. We have to replace one redundant equation by an additional independent condition:

$$\sum_{l=-2N+1}^{2N-1} l^n \mathbf{\Phi}^{(n)}(l) = (-1)^n n! \tag{2.37}$$

Finally the values at the collocation points can be obtained by solving the recursive relation 2.38 and using the values at the integer positions from the solution of the equation system 2.36 and 2.37.

$$\mathbf{\Phi}^{(n)}(2^{-j-1}k) = 2^n \mathbf{\Phi}^{(n)}(2^j k) + 2^{n-1} \sum_{l=1}^{N} a_{2l-1} \left(\mathbf{\Phi}^{(n)}(2^{-j}k-2l+1) + \mathbf{\Phi}^{(n)}(2^{-j}k+2l-1) \right) \tag{2.38}$$

For the 'zero derivative' $n = 0$ we recover equation 2.35. We are now able to draw the derivatives of the autocorrelation printed in figure 18. Figure 19 shows the first and second derivatives of the autocorrelation function.

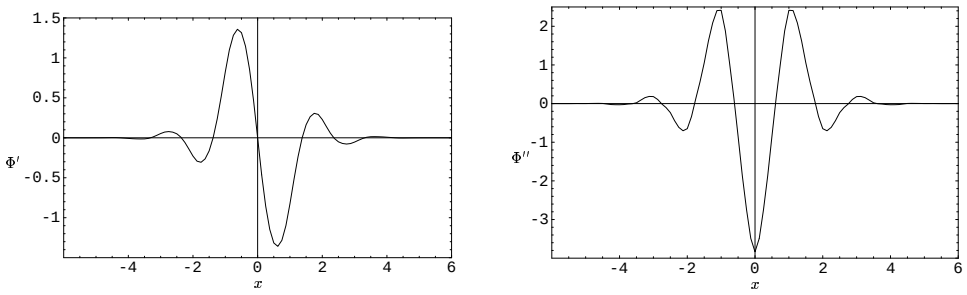

Figure 19: First and second derivative of autocorrelation of Daubechies scaling function.

We can compare the autocorrelation wavelet and its derivatives 18 and 19 with the the Gaussian 'Mexican hat ' wavelet and its derivatives from equation 2.30 and

figure 16. We realize that the autocorrelation wavelet looks similar but has a smaller support which is to the best advantage in the case of localized structures.

2.7 Two- and Threedimensional Wavelets

In this report we treat one-dimensional problems, since it is easier to show the wavelet properties. However the application of wavelets for two- and threedimensional problems is already underway, see e.g Vasilyev and Paolucci (1996b) or Griebel and Koster (2000). A twodimensional wavelet can be generated by a tensor product of two one-dimensional wavelets, see the wavelet equation 2.22, since both wavelets $\Psi(x)$ and $\Psi(y)$ fulfil the wavelet equation:

$$\Psi^j_{k,l}(x,y) = \Psi^j_k(x) \times \Psi^j_l(y). \tag{2.39}$$

The same is valid for scaling functions. Once we have a one-dimensional wavelet or scaling function we can proceed straightforward. Figure 20 shows the twodimensional autocorrelation 'wavelets' of the Daubechies scaling function of the previous section.

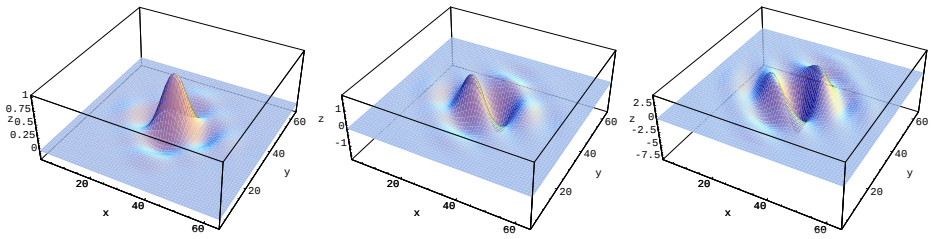

Figure 20: From left to right: Twodimensional autocorrelation of Daubechies scaling function. First derivative and second derivative.

For the derivative of the two-dimensional wavelet we have to apply the product rule:

$$\Psi^{j'}_{k,l}(x,y) = \Psi^{j'}_k(x) \times \Psi^j_l(y) + \Psi^j_k(x) \times \Psi^{j'}_l(y). \tag{2.40}$$

The generalization to three dimensions is also straightforward, but is is not possible any more to plot these functions.

2.8 The (Fast) Discrete Wavelet Transformation

In section 2.2 on multiresolution we saw that we can decompose a function in smooth parts and details. More generally, any square integrable function can be expanded in terms of scaling functions and wavelets. Such decomposition is called a wavelet transform. In a continuous form we can write for the wavelet transformation:

$$W_\Psi f(a,b) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt. \tag{2.41}$$

The scale a is a measure for the frequency and the position b is a measure for the time. In the discrete wavelet transform the scale of the wavelet is not changed continuously but in fixed (dyadic) steps. This becomes clear if we look at the wavelet equation 2.22.

As shown in the section on multiresolution analysis 2.2 we can decompose a function in a coarse approximation and finer details. We can assemble the original

function when we add all details to the function at a given resolution j. We repeat equation 2.3

$$f(t) = f_j(t) + \sum_{k=j}^{\infty} d_k(t).$$

Let us now assume that our function is known up to a specified resolution, say $2^{-(j+1)}$. We can now assemble our function out of details and the coarsest approximation, say f_0 in the following way

$$f_{j+1}(t) = f_0(t) + \sum_{k=0}^{j+1} d_k(t). \tag{2.42}$$

Equation 2.42 describes the assembly of the function following the graphical description in figure 3 from low resolution to high resolution.

The decomposition of a function goes the way in figure 3 from high resolution to low resolution. Suppose we can decompose our function f_{j+1}

$$f_{j+1}(t) = f_j(t) + d_j(t). \tag{2.43}$$

With the knowledge of multiresolution analysis we can write in terms of scaling functions and wavelets

$$f_{j+1}(t) = f_j(t) + d_j(t) = \sum_k f_k^j \Phi_{j,k}(t) + \sum_k d_k^j \Psi_{j,k}(t). \tag{2.44}$$

In equation 2.44 we have written the approximation $f_j(t)$ as a linear combination of coefficients f_k^j and the corresponding scaling function $\Phi_{j,k}(t)$ and the details as a linear combination of coefficients d_k^j with the wavelet $\Psi_{j,k}(t)$. We obtain these coefficients by computing the inner product:

$$f_k^j = \langle \Phi_{j,k}, f_{j+1} \rangle = \int \Phi_{j,k}(t) f_{j+1}(t) dt \qquad d_k^j = \langle \Psi_{j,k}, f_{j+1} \rangle = \int \Psi_{j,k}(t) f_{j+1}(t) dt \tag{2.45}$$

Now we have decomposed our function of scale $j+1$ in approximation and details of the next coarser scale j. We can continue to decompose f_j in f_{j-1} and d_{j-1}. Finally we have decomposed our function in terms of coefficients of successively finer scales of the wavelets. In terms of the function spaces V and W we can write:

$$V_{j+1} = W_j \oplus V_j = W_j \oplus (W_{j-1} \oplus V_{j-1}) = W_j \oplus W_{j-1} \oplus W_{j-2} \oplus \cdots \oplus W_{j-J} \oplus V_{j-J} \tag{2.46}$$

In practice the algorithm is somewhat modified so that the wavelet and the scaling function do not enter the algorithm. Only the coefficients of the dilation equation and the wavelet equation are needed. Since $V_{j-1} \subset V_j$ we can build scaling functions in the coarser space V_{j-1} out of scaling functions in the next finer space V_j. Shortly we can write :

$$\Phi_{j,k}(t) = \sum_l \langle \Phi_{j+1,l} \Phi_{j,k} \rangle \Phi_{j+1,l}(t), \qquad \text{with} \quad l \in \mathbb{Z} \tag{2.47}$$

We introduce the index l since we have to build every 'building block' $\Phi_{j,k}$ in equation 2.44 with a finite number of functions of the next finer scale. Within limits we can choose l freely. We can rearrange the inner product in equation 2.47 and substitute $(u = 2^j t - k)$:

$$\langle \Phi_{j+1,l} \Phi_{j,k} \rangle = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} \Phi(2^j t - k) 2^{\frac{j+1}{2}} \Phi(2^{j+1} t - l) dt = \sqrt{2} \int_{-\infty}^{\infty} \Phi(u) \Phi(2u - (l - 2k)) du. \tag{2.48}$$

We recognize that with equation 2.20, derived from the dilation equation, we obtain the coefficients h. In this case :

$$\langle \Phi_{j+1,l} \Phi_{j,k} \rangle = h_{l-2k} \tag{2.49}$$

Analogous we get :

$$\langle \Phi_{j+1,l} \Psi_{j,k} \rangle = g_{l-2k} \tag{2.50}$$

Finally we can write:

$$\Phi_{j,k}(t) = \sum_l h_{l-2k} \Phi_{j+1,l}(t) \qquad \text{and} \qquad \Psi_{j,k}(t) = \sum_l g_{l-2k} \Phi_{j+1,l}(t) \tag{2.51}$$

With the above relation we can derive a description for the coefficients f_k^j and d_k^j in equation 2.44.

$$f_k^j = \langle \Phi_{j,k} f_{j+1} \rangle = \sum_l h_{l-2k} \langle \Phi_{j+1,l}, f_{j+1} \rangle = \sum_l h_{l-2k} f_l^{j+1} \tag{2.52}$$

So once we have the coefficients $f_l^{j+1} = \langle \Phi_{j+1,k} f_{j+1} \rangle$ in the first decomposition, we can evaluate the coefficients of all coarser spaces only with the knowledge of the coefficients h of the dilation equation. Note that any two scales are connected with the same coefficients h.

Similarly we have for the detail spaces:

$$d_k^j = \langle \Psi_{j,k} f_{j+1} \rangle = \sum_l g_{l-2k} \langle \Phi_{j+1,l}, f_{j+1} \rangle = \sum_l g_{l-2k} f_l^{j+1} \tag{2.53}$$

Following equation 2.44 we can express f_{j+1} in terms of f_k^j and d_k^j. We can repeat this procedure until we reach the coarsest level. Using the f_k^j as the new f_k^{j+1} for decomposition we get f_k^{j-1} and d_k^{j-1} etc. .

Assuming that the number of the coefficients in the dilation equation is L and the number of data points of the original function is N this algorithm needs $O(2LN)$ operations following He (1996). In analogy to the fast Fourier transform this algorithm is also called the 'fast wavelet transform'.

Let us shortly summarize the wavelet transform algorithm. Assume that we use a known scaling function and the wavelet. From the conditions of the multiscale analysis (see section 2.5) we know the coefficients h_k and g_k of the dilation equation and the wavelet equation.

We first have to compute the inner product of the scaling function and the function $f(t)$ defined on a fixed number of sampling points

$$f_k^{j+1} = \langle \Phi_{j+1,k}, f_{j+1} \rangle = \int \Phi_{j+1,k}(t) f_{j+1}(t) dt. \tag{2.54}$$

With this result we can compute the coefficients of the next coarser scales. The coefficients f_k^{j+1} are plugged in the right side of the recursive equations (2.52, 2.53)

$$f_k^j = \textstyle\sum_l h_{l-2k} f_l^{j+1} \tag{2.55}$$

$$d_k^j = \textstyle\sum_l g_{l-2k} f_l^{j+1} \tag{2.56}$$

This process is continued until we reach the coarsest level, say j_0. As a result we have the coefficients f_k^j and d_k^j of all resolution levels $j = j+1, \cdots j_0$. Note that the wavelets Ψ are never used, only the scaling function and the coefficients of the wavelet- and dilation equation. With this result we can assemble the original signal from the coarsest level j_0 with the following equation

$$f_{j+1}(t) = \sum_k f_k^{j+1} \Phi_{j+1,k}(t) = \sum_{l=j_0}^{j} \sum_k d_k^l \Psi_{l,k}(t) + \sum_k f_k^{j_0} \Phi_{j_0,k}(t). \tag{2.57}$$

2.9 Wavelets and Filters

In this section we describe the connection between the fast wavelet transform and the usage of filtering in signal processing. This is useful since we will see that the principle of wavelets can be explained very clear.

In most cases a signal coming from an experimental setup is filtered to exclude undesirable influences from sources outside the experiment, e.g. the 50 Hz signal coming from power supply or noise. Therefore an incoming signal is often decomposed into different frequency bands called channels. These channels can then be processed separately and more efficiently.

The decomposition of the signal can be done by an adjustable filter or by several filters called filter bank.

Mathematically a filter can be described by a linear transformation that transforms an input signal. For a continuous function we can write

$$\bar{f}(x) = \int f(y)c(y-x)dx. \tag{2.58}$$

This is called a convolution. [1] For a discrete input signal $f(x_n) = f_n$ and the linear filter c we can write for the output signal $\overline{f}$

$$\overline{f}_n = \sum_m c_m \overline{f}(x_{m-n}). \tag{2.59}$$

With Fourier-transforms $\mathcal{F}$ and the inverse Fourier-transform $\mathcal{F}^{-1}$ we can easily calculate the result.

$$\overline{f} = \mathcal{F}^{-1}(\mathcal{F}(c) \cdot \mathcal{F}(x)). \tag{2.60}$$

The filter c is often called the impulse response since it is the output signal when the input is a unit impulse (Dirac delta function). If we have a finite number of nonzero filter coefficients it is called a finite impulse response filter (FIR). The Fourier transform of the filter coefficients is called the frequency response function or transfer function. The frequency response shows us the effect of the filter in frequency space. From the frequency response we can easily identify high-pass, lowpass or bandpass filters.

To see the similarity of filters and wavelets we show the following example. We can think of two filters lowpass c and highpass d separating the original signal $f_n(x)$ in two datasets:

$$\overline{f}_{y_n} = \sum_m c_m f(x_{m-n}) \qquad \overline{f}_{z_n} = \sum_m d_m f(x_{n-m}). \tag{2.61}$$

We notice that the amount of data points ($\overline{f}_{y_n}$ and $\overline{f}_{z_n}$) is doubled due to filtering but the information must be the same as in the original data set. With special properties of the filters (halfband filters) this redundancy of the data can be removed by deleting every second data point in the filtered signal. The operation is called downsampling or decimation by 2. The justification of the downsampling can be shown with the Shannon sampling theorem. For the proof, see Daubechies (1992). After downsampling we get

$$\overline{f}_{y_{2n}} = \sum_m c_{2n-m} f(x_m) \qquad \overline{f}_{z_{2n}} = \sum_m d_{2n-m} f(x_m) \tag{2.62}$$

If we compare this with the expression for the wavelet decomposition (equation 2.55) we realize that wavelet decomposition is nothing else than a repeated filtering

[1] The convolution of a function with itself is an auto-correlation, with a other function a cross-correlation.

with a lowpass and highpass FIR-filter. Therefore we can call the coefficients h and g in equation 2.55 also the filter coefficients of a lowpass and highpass filter. We can draw the wavelet decomposition 2.55 as a cascade of half-band filters.

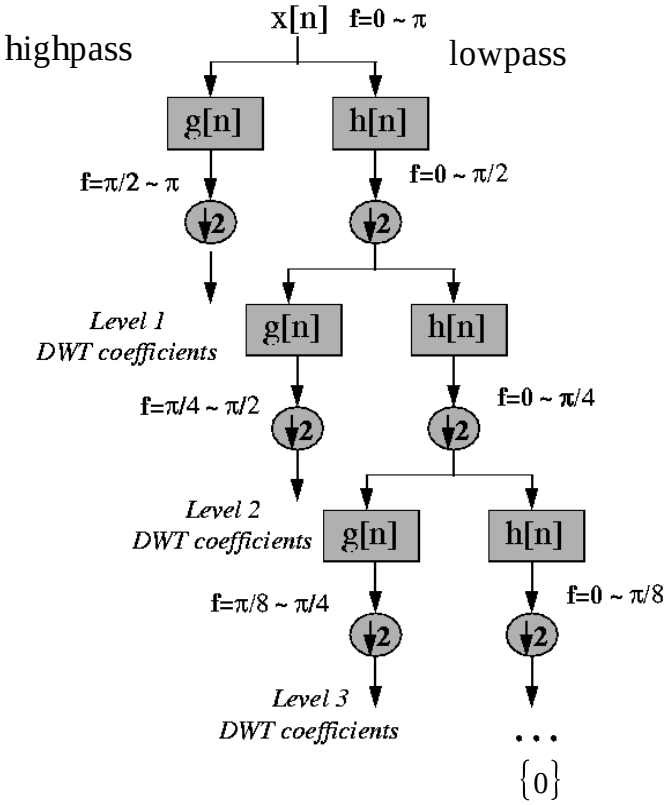

Figure 21: Graphical representation of discrete wavelet transform.

Once we know the connection between wavelet transform and filtering we can explain some properties of wavelets and scaling functions. It is interesting to see and compare the impulse response and the frequency response of different scaling functions and wavelets.

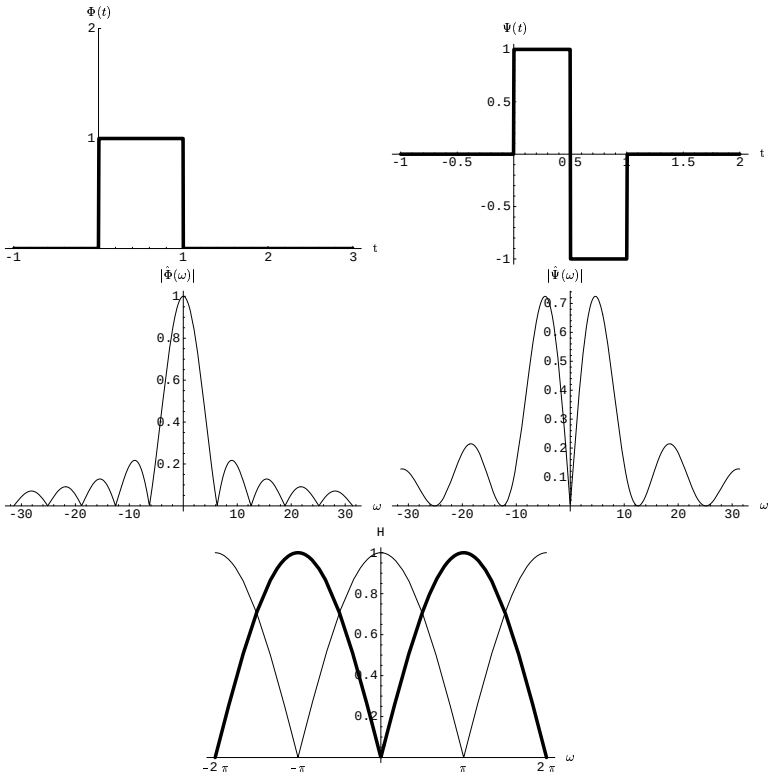

Figure 22: First row: Haar scaling function, wavelet, second row: Fourier-transforms, third row: frequency response of wavelet (bold) and scaling function.

In figure 22 we plot the Fourier transforms of the Haar wavelet and scaling function. We realize that due to the sharp gradients the Haar wavelet has no good localization in Fourier space but the localization in time is very good. In the frequency response we observe that the scaling function represents the lowpass filter and the wavelet (bold line) the highpass filter. We realize also that approximately half of the frequencies are filtered out by one of the filters (so called halfband filters). The frequency regions of the filters have a wide overlap.

The next scaling functions and wavelets in figure 23 and 24 are build out of splines. We realize that the localization in Fourier space becomes better as is the Haar case. In figure 24 we have a nearly perfect half band filter. In figures 25 and 26 we have Daubechies wavelets with two and twelve vanishing moments.

There is no final answer what wavelet to use for the application given. For numerical simulation there seems to be a trend to use wavelets with 5 or 6 vanishing moments. On the other hand the wavelets used for numerics do not have to be orthogonal but symmetry of the wavelet is advantageous. Therefore also autocorrelations of wavelets are used which are not orthogonal.

For signal decomposition orthogonal filters are profitable, since then it is possible to simply add the different resolution levels omitting the highest levels for the elimination of disturbances.

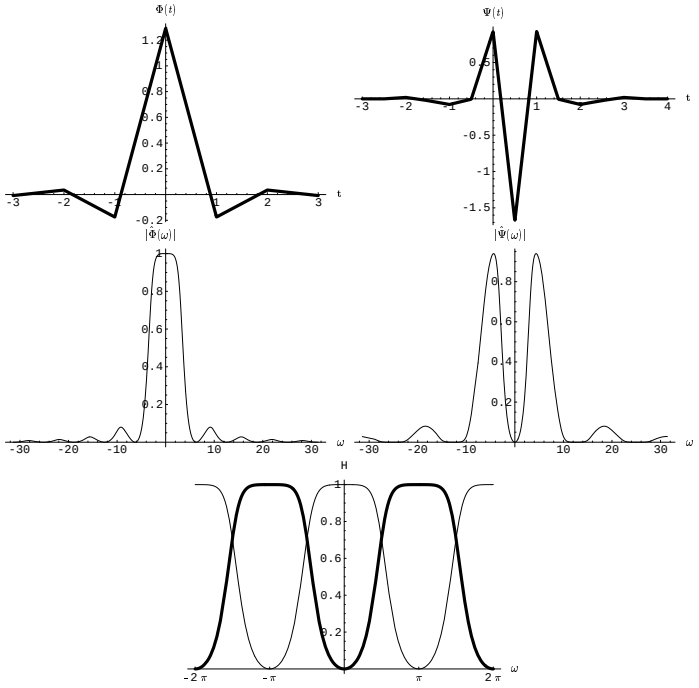

Figure 23: First row: Spline order 1 scaling function, wavelet, second row: Fourier-transforms, third row: frequency response of wavelet (bold) and scaling function.

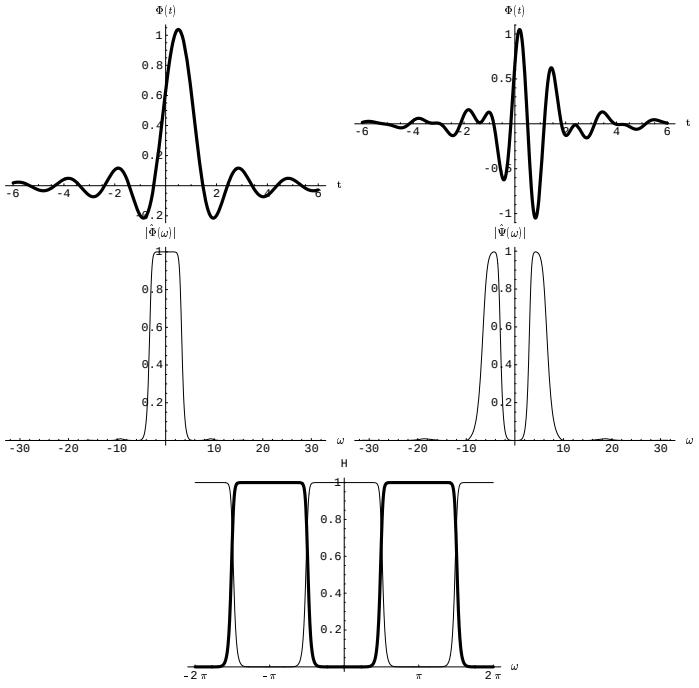

Figure 24: First row: Spline order 8 scaling function, wavelet, second row: Fourier-transforms, third row: frequency response of wavelet (bold) and scaling function.

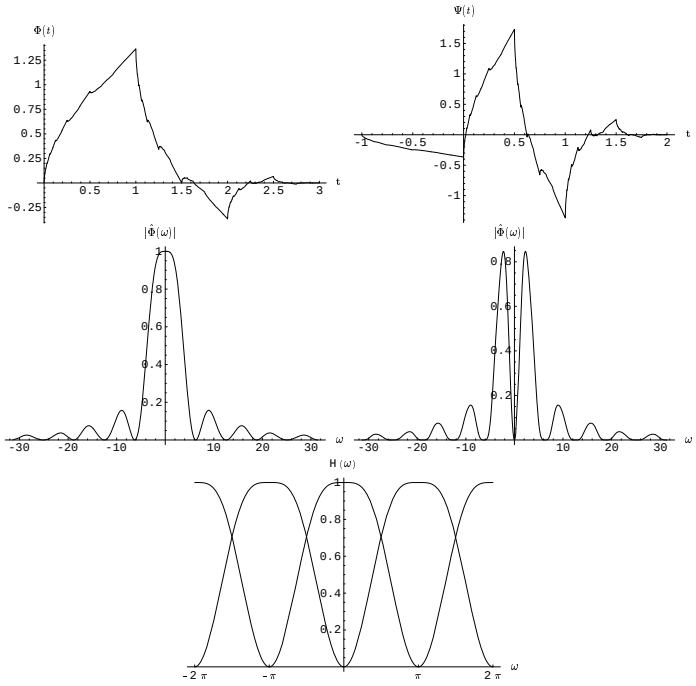

Figure 25: First row: Daubechies order 2 scaling function, wavelet, second row: Fourier-transforms, third row: frequency response of wavelet (bold) and scaling function.

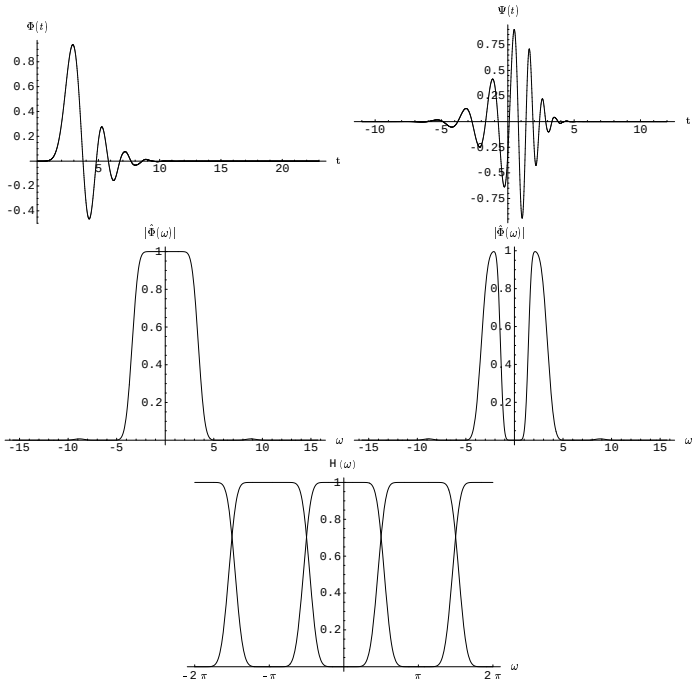

Figure 26: First row: Daubechies order 12 scaling function, wavelet, second row: Fourier-transforms, third row: frequency response of wavelet (bold) and scaling function.

2.10 Examples of Wavelet Transforms

The wavelet transform is useful for noise-reduction in signals. If we have a non stationary signal where frequencies vary with time it is possible to find out where in the signal the variation appears.

Consider the following signal in figure 27. We have added disturbances near $x = 200$ and $x = 700$ which we would like to remove from the signal with a minimal influence to the rest of the signal.

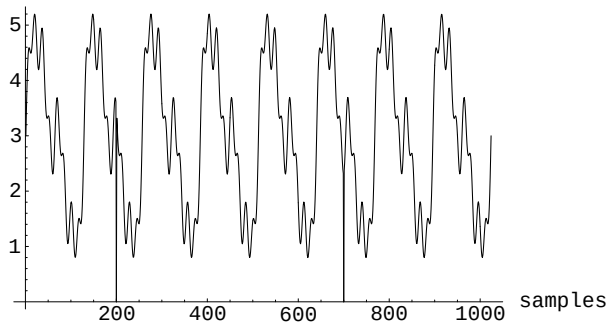

Figure 27: The function $f(t) = 0.4sin(64t)+0.4sin(32t)+0.8sin(16t)+1.6sin(8t)+3$ with additional disturbances.

Figure 28 shows the multi-resolution-decomposition of this signal carried out with a spline filter of order 2.

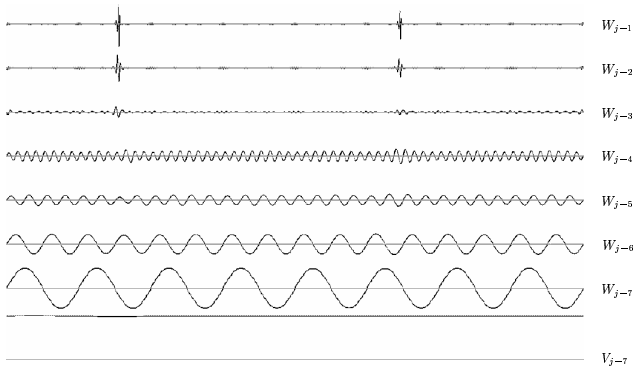

Figure 28: Multi-resolution-decomposition of signal.

The plot shows the details of the signal at several levels. The undermost curve (V_{j-7}) shows the coarsest approximation of the signal. It is the mean part of the signal, i.e. the part of the signal represented by a constant.

Following the decomposition in figure 3 and equation 2.46 we can assemble the original signal by simply adding the details($W_{j-7}, W_{j-6}, W_{j-5}, W_{j-4} \cdots$) to the coarsest approximation V_{j-7}. This is possible since the decomposition is orthogonal.

The curves $W_{j-1}, W_{j-2}, W_{j-3}$ are the levels where the disturbances are located. We can simply omit these levels to remove the disturbances from the signal. The result is shown in figure 29.

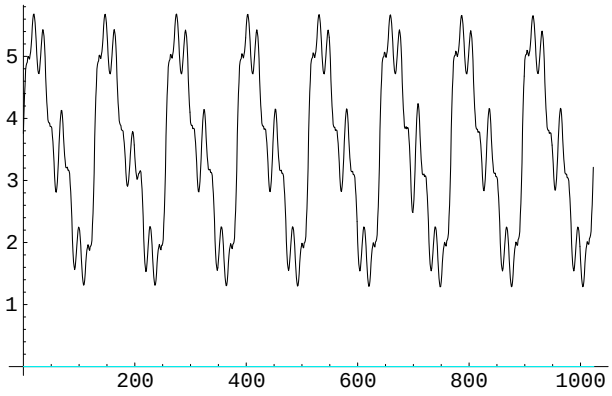

Figure 29: Signal with removed disturbances.

With this procedure we can specify where the disturbances are located in the signal and remove them.

Another example where the frequency changes in time is the so called chirp signal. We consider the function:

$$f(t) = sin(50\pi t^2). \tag{2.63}$$

This function is shown in figure 30.

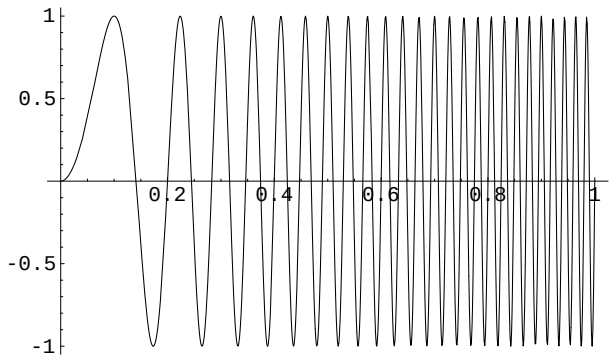

Figure 30: The chirp function.

The multi-resolution-decomposition of this signal, see figure 31 reflects the change of frequency in time.

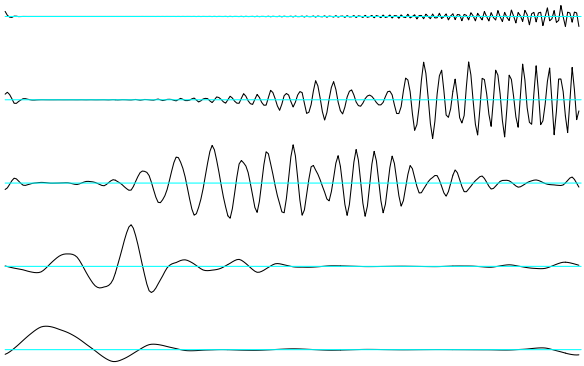

Figure 31: The multi resolution decomposition of the chirp function.

Since we can assign the wavelets to certain frequency ranges it is possible to plot the frequency-ranges of the wavelets over the time in a so called phase-space-plot.

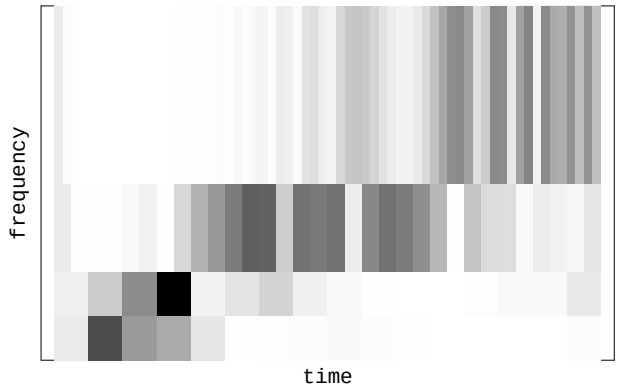

Figure 32: Phase-space-plot of chirp function.

We can see from figure 32 that the frequency changes roughly linear in time. The darkness of the rectangles is related to the magnitude of the wavelet coefficients. The rising aspect ratio of the rectangles results from the fact that the localization in Fourier space of the scaled wavelets (for the high frequencies) becomes worse. This can be explained as follows. In figure 33 we realize that scaled narrow wavelets have a good localization in time space but they take a broad range of wave numbers in Fourier space.

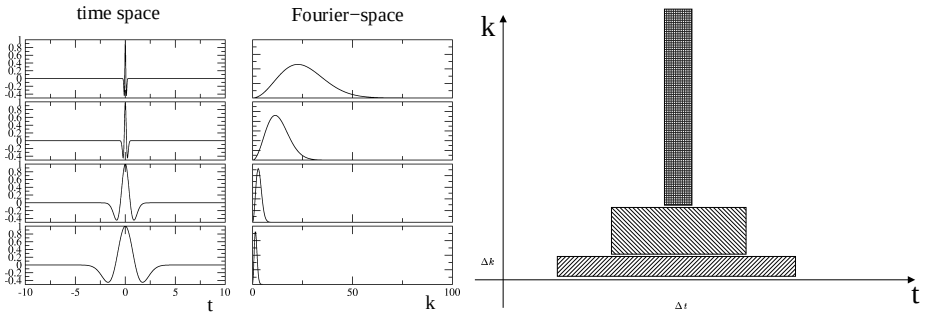

Figure 33: Scaled wavelet in time- and Fourier space, phase plot

In a phase-diagram we plot the wave-number k versus the time, see figure 33. We see a typical property of the wavelet transform. The high frequencies which are captured by small wavelets can be located very accurate in time but not in frequency whereas the low frequencies are captured by the large wavelets which have a good localization in frequency but a poor localization in time. Such a behavior is advantageous in analyzing signals since mostly the disturbances have high frequencies such as peaks or noise whereas the part of the signal which is analyzed contains the lower frequencies which are captured more exact by the broader wavelets. So we can localize the disturbances exactly and remove them which was demonstrated in the first example. In the next example the frequency varies more complicated in time. But we identify immediately how the frequencies vary with time, see figure34.

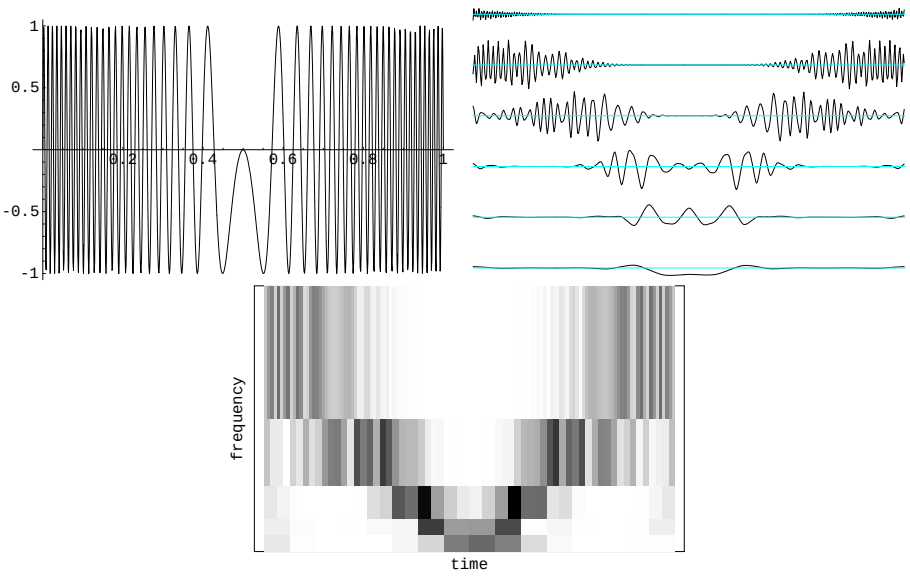

Figure 34: Signal, multiresolution decomposition and phase space plot.

In many practical applications where frequencies vary in time wavelet transformation is a useful tool. In medical science wavelets are used to analyze the heart beat(EKG) or brain currents (EEG) to detect Alzheimers disease. Also the analysis and manipulation of sound is an interesting topic, e.g. to detect a beginning damage in machines. The simulation of human vision and pattern recognition seems promising with wavelets.

It is also possible to compress signals by wavelet transform. We can omit e.g. wavelet coefficients at every level below a certain threshold. This is the guiding principle in image compression.

The application of wavelets is a quite large topic and we refer to textbooks on wavelets for a more complete overview. This completes our short introduction to wavelet transform. In the next section we will deal with the solution of differential equations using wavelets.

3 Differential Equations and Wavelets, Results

A common problem in numerical simulations of partial differential equations is the large range of temporal and spatial scales present in the solution of e.g. the Navier-Stokes equations.

One can think of turbulent flows where the flow is governed by very localized small structures. In magnetohydrodynamic flows the very thin Hartmann layers appear. In microscale flows the electric double-layer may become important. In the field of combustion chemical reactions occur in very thin layers which travel as a front through the domain. Many other examples exist as well.

All characteristic scales have to be resolved by the numerical grid. As a consequence the used grids have to be adapted to the solution to avoid an excessive number of grid-points.

Adaptive grids in finite differences, finite elements or finite volumes often require a significant effort for implementation and computation. The grid is refined or relaxed following several conditions. Often more computing time is spent in the grid computation than in the solver for the equations.

In the wavelet-approach the scales of the wavelets are adapted to the size of localized processes or structures. The adaption of wavelets due to scaling makes them ideally suited for adaptive methods in numerical simulation. The multilevel wavelet collocation method following Vasilyev et al. (1995) naturally leads to an adaption of collocation points by taking only wavelets whose coefficients exceed a specified threshold.

In the following we will explain this algorithm. Additionally we show some examples.

3.1 The Wavelet Collocation Method

The collocation method proposed by Vasilyev et al. (1995) is an approach for solving partial differential equations with arbitrary boundary conditions. The algorithm uses the idea of collocation with functions similar to wavelets, which are placed on the collocation points.

The algorithm can be used with any function which has local support in physical space and in wave-number space. The exact definitions for a wavelet do not have to be fulfilled. In most cases functions which have more similarities to scaling functions are used. The functions are commonly referred to as 'wavelets' although this is not exact following the definitions in the last section.

3.1.1 Distribution of Collocation Points

The first thing to do is to construct a dyadic distribution of collocation points in the interval of the function which we want to reconstruct with wavelets. The dyadic distribution is not the only possible one but it is often chosen for computational convenience. In principle we have a discretization of the continuous wavelet transform, see equation 2.41, by discretization of the parameters 'a' for the scale of the wavelet and 'b' for the position. Usually a doubly-indexed family of wavelets is created by dilation and translation:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi \left(\frac{t-b}{a} \right). \tag{3.64}$$

The parameter a is replaced by powers of a fixed dilation parameter. In our case this is $a(j) = a^{-j}$, where $j \in \mathbb{Z}$. Sometimes a^j is chosen, but then the numbering of the sequence of the nested subspaces, see figure 3, has to be reversed. The result

remains the same. The parameter b is replaced by $b(j,k) = kb_0a^{-j}$, where $b_0 \in \mathbb{Z}$ is the so called spacing parameter. So we get finally

$$\Psi_{j,k}(t) = a^{j/2}\Psi\left(a^j t - kb_0\right). \tag{3.65}$$

In most cases a dyadic decomposition of scales is chosen, i.e. $a = 2$. If we have finite intervals, e.g. from t_l to t_r, the following is chosen to replace the parameters in equation 3.64:

$$a = 2^{-j}a_0, \quad \text{with} \quad a_0 = 2^{-L}\frac{(t_r - t_l)}{b_0}, \quad L, b_0 \in \mathbb{Z} \tag{3.66}$$

$$b = \frac{(t_r+t_l)}{2} + ab_0k \quad . \tag{3.67}$$

b_0 is sometimes called spacing parameter, since it influences the distance between two wavelets. L is sometimes called scaling parameter, since it influences the number of points at each level and the scale of the wavelet. In most cases $L = 1$ and $b_0 = 1$ is chosen. If we take these values and consider an interval between $t_l = -1$ and $t_r = 1$, we can draw the dyadic distribution of grid points at different levels. We plot the collocation points $b(j,k)$ in figure 35.

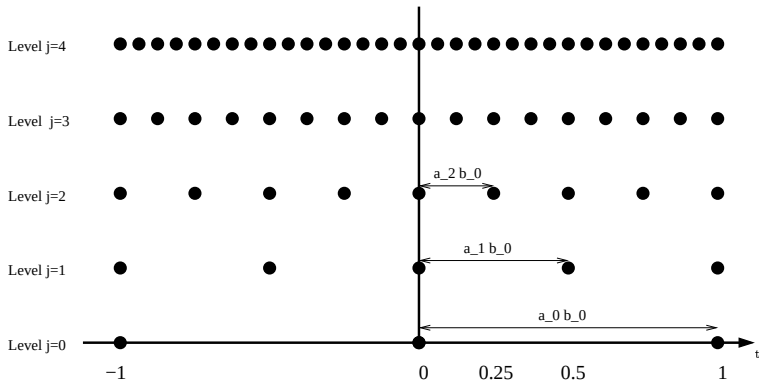

Figure 35: collocation points $b(j,k)$ of the dyadic grid

We realize that this distribution of collocation points, once we have the wavelet coefficients, will show the desired function in coarse and fine resolutions depending on the level. For the justification of other values of L and b_0 we refer to Vasilyev et al. (1995).

The next thing to do is to place the different scaled wavelets on the collocation points and then to find an algorithm to solve for the coefficients.

3.1.2 Approximation of a Function

Using the wavelets, the approximation of a function $u(t)$ can be written as :

$$u^J(t) = \sum_{j=0}^{J} \sum_{k \in \mathbb{Z}_\Omega^j} c_k^j \Psi_k^j(t), \tag{3.68}$$

where Ω is the domain in which the function $u(t)$ is considered. The index $k \in \mathbb{Z}_\Omega^j$ stands for all integer indices of the collocation points placed in the interval between t_l and t_r at the considered level j, see figure 35. Note the similarity of equation 3.68 to the reconstruction formula of the discrete wavelet transform, see equation

2.57. The difference to equation 2.57 is the missing term for the lowest considered level j_0. It can be shown for the algorithm on hand that for a proper choice of the parameter L in equation 3.66, e.g. $L = 1$, all the levels below $j = 0$ can be approximated by a constant. For a proof see Vasilyev et al. (1995).

As an example we want to approximate the following function:

$$u(t) = \frac{1}{2}(t^2 - 1) - sin(\pi t) + \frac{1}{4}sin(4\pi t) + \frac{1}{8}sin(8\pi t) \tag{3.69}$$

Figure 36 shows this function.

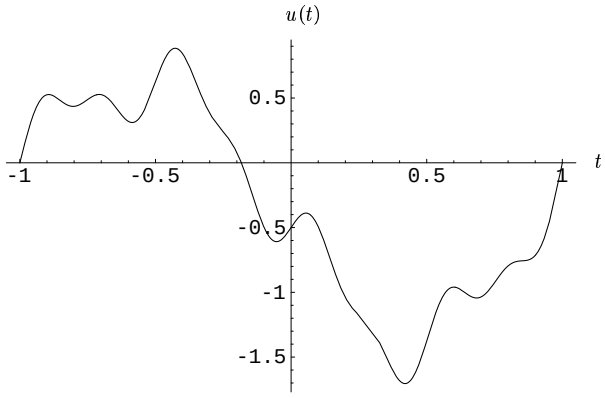

Figure 36: Plot of equation 3.69.

The next task is to place the wavelets on the collocation points. We start with the level $j = 0$, see figure 35. We place three wavelets which are scaled following equation 3.65 on the collocation points at level $j = 0$, see figure 37. In our case we use the 'Mexican hat' wavelet.

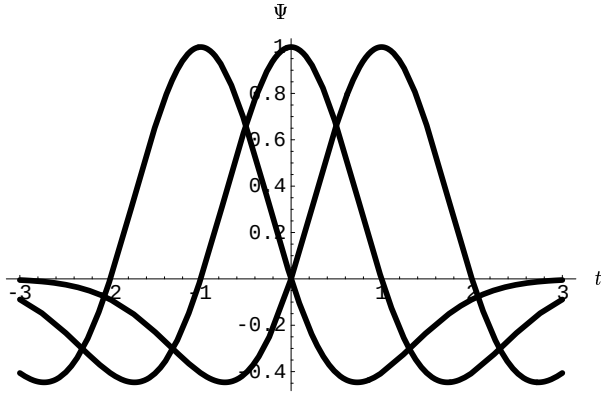

Figure 37: Placement of the 'Mexican-hat' wavelets at the level $j = 0$.

We write equation 3.68 for the level $j = 0$ for the wavelets and the collocation points.

$$u^0(t_i^0) = \sum_{i,k \in \mathbb{Z}_\Omega^0} c_k^0 \Psi_k^0(t_i^0), \qquad \text{with} \qquad i \in \mathbb{Z}_\Omega^0 = \{1, 2, 3\} \tag{3.70}$$

Equation 3.70 can be written as a linear system of equations since we can write the term $\Psi_k^0(t_i^0)$ as a square matrix and the other terms as vectors. The following figure 38 shows this again.

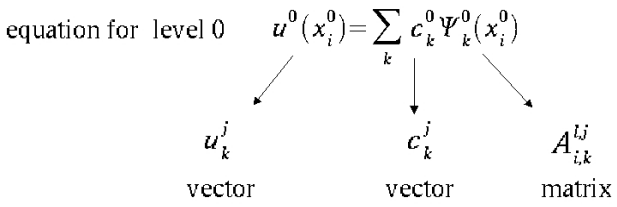

Figure 38: Equation 3.70 for level 0.

The matrix A is constructed out of the values of the wavelets at the different collocation points. The vector c contains the unknown coefficients, the vector u contains the values of the function at the collocation points. In our case it is:

$$A = \begin{bmatrix} 1 & 0 & -0.4 \\ 0 & 1 & 0 \\ -0.4 & 0 & 1 \end{bmatrix}; \qquad c = \begin{bmatrix} c_1^0 \\ c_2^0 \\ c_3^0 \end{bmatrix}; \qquad u = \begin{bmatrix} 0 \\ -0.5 \\ 0 \end{bmatrix} \tag{3.71}$$

The first row of A shows the values of the wavelet at position $t = -1$ at $t = -1, 0, 1$, see figure 37. The second row shows the same with the second wavelet, the third with the third wavelet. With a linear algebraic equation solver we solve the system for the unknown coefficients c

$$Ac = u, \qquad \text{with} \quad c = \begin{bmatrix} 0 \\ -0.5 \\ 0 \end{bmatrix}. \tag{3.72}$$

With the coefficients we can now construct a continuous approximation function for the level $j = 0$

$$u_{approx}^0(t) = c^0 \Psi^0(t). \tag{3.73}$$

Figure 39 shows this approximation function compared with function to approximate.

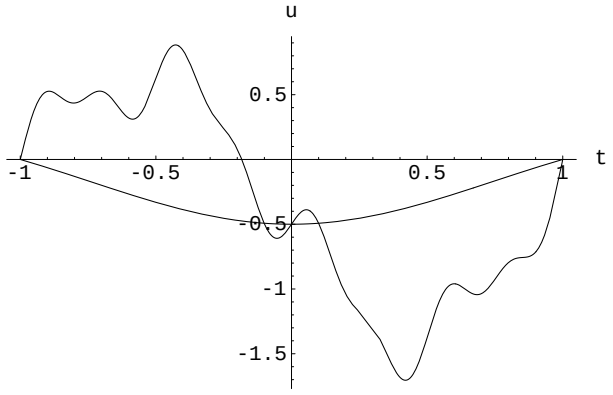

Figure 39: Approximation function for level 0.

We realize that the approximation meets the function at the three collocation points.

In the next level $j = 1$ we have five collocation points, see figure 35. We can assemble the next level out of the approximation of the previous level plus a

correction at the additional collocation points. We make the following ansatz:

$$u^1(t_i^1) = u^0_{approx}(t_i^1) + \sum_{i,k \in \mathbb{Z}^1_\Omega} c_k^1 \Psi_k^1(t_i^1) \tag{3.74}$$

Since $u^0_{approx}(t)$ is a continuous function we can evaluate the values of that function at the collocation points of the next level $j = 1$, then we add the corrections. We can now rearrange equation 3.74

$$\sum_{i,k \in \mathbb{Z}^1_\Omega} c_k^1 \Psi_k^1(t_i^1) = \underbrace{u^1(t_i^1) - u^0_{approx}(t_i^1)}_{\Delta^1 u}. \tag{3.75}$$

The Δ operator is introduced for later simplifications, since it can be expressed separately as a sum of prolongations and restrictions. Again we end with a linear equation system:

$$A^{1,1}_{i,k} c_k^1 = \Delta^1 u \tag{3.76}$$

The continuous approximation function at the level $j = 1$ is given by the superposition of the approximation functions at level $j = 0$ and $j = 1$.

$$u^1_{approx}(t^1) = c^0 \Psi^0(t^1) + c^1 \Psi^1(t^1) \tag{3.77}$$

Figure 40 shows the approximation function for the level $j = 1$:

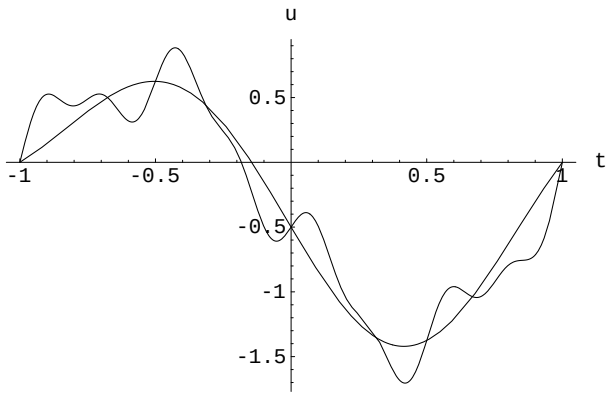

Figure 40: Approximation function for level 1.

The approximation meets the function at five collocation points of the first level, see also figure 35.

From this point the algorithm is just repeated for the next levels. We now write the equations in a more general form:

$$\sum_{i,k \in \mathbb{Z}^j_\Omega} c_k^j \Psi_k^j(t_i^j) = \underbrace{u^j(t_i^j) - u^{j-1}_{approx}(t_i^j)}_{\Delta^j u} \tag{3.78}$$

$$A^{j,j}_{i,k} c_k^j = \Delta^j u \tag{3.79}$$

We can invert the matrix A and write:

$$c_k^j = \underbrace{\left(A^{j,j}_{i,k}\right)^{-1} \Delta^j}_{C^j} u \tag{3.80}$$

Up to now the Δ operator is not exactly defined. It can be constructed out of restrictions and prolongations and will be described later in the text after the definitions of the restriction and prolongation.

Due to the multiresolution character of the algorithm, the following relation of the collocation points t is valid between different resolution levels:

$$t_i^j \subset t_i^{j+1}. \tag{3.81}$$

So we have the same values of the function at different levels of resolution at the same collocation points, i.e.

$$u^l(t_i^l) = u^j(t_m^j) \qquad \text{if} \quad t_i^l = t_m^j \tag{3.82}$$

The relation between the values of the function $u(t)$ from high to low resolution is a restriction. We define a restriction operator which only contains 0 and 1 :

$$R_{i,m}^{l,j} = \begin{cases} 1 & \text{for} \quad t_i^l = t_i^j, \\ 0 & \text{otherwise} \end{cases} \qquad \text{with} \quad 0 \le l \le j \le J. \tag{3.83}$$

J is the highest considered resolution level. Once we have the collocation points and the function $u(t)$ we know all restriction matrices before we begin with the calculation. With the restriction matrix any set of values at a lower resolution l as the given resolution j can be evaluated by :

$$u^j(t_i^l) = u^l(x_i^l) = \sum_{i,m \in \mathbb{Z}_\Omega^j} R_{i,m}^{l,j} u_m^j, \qquad \text{with} \quad 0 \le l \le j \le J, \tag{3.84}$$

Equation 3.84 can be considered as a product of the restriction matrix R and a vector u.

Similar to the restriction operator we can introduce a prolongation operator :

$$u^j(x_i^l) = \sum_{i,m \in \mathbb{Z}_\Omega^j} P_{i,m}^{l,j} u_m^j, \qquad \text{with} \quad 0 \le j \le l \le J, \tag{3.85}$$

The prolongation operator just 'computes' the values of the approximation at level j (u^j) at the collocation points of a higher level l (x_i^l). With some formal rearrangement explained in Vasilyev et al. (1995) we obtain a recursive description of the prolongation operator:

$$P_{i,m}^{l,j} = \begin{cases} \sum\limits_{p \in \mathbb{Z}_\Omega^{j-1}} R_{p,m}^{j-1,j} + \sum\limits_{k \in \mathbb{Z}_\Omega^j} A_{i,k}^{l,j} C_{k,m}^{j,j}, & 1 = j \le l \le J, \; i \in \mathbb{Z}_\Omega^l, \; m \in \mathbb{Z}_\Omega^j \\ \sum\limits_{k \in \mathbb{Z}_\Omega^0} A_{i,k}^{l,0} C_{k,m}^{0,0}, & j = 0, \; 0 \le l \le J, \; i \in \mathbb{Z}_\Omega^l, \; m \in \mathbb{Z}_\Omega^0 \end{cases} \tag{3.86}$$

With the restrictions and prolongations it is now possible to define the delta operator:

$$\Delta_{i,m}^{j,s} = \begin{cases} R_{i,m}^{j,s} - \sum\limits_{p \in \mathbb{Z}_\Omega^{j-1}} P_{i,p}^{j,j-1} R_{p,m}^{j-1,s}, & 1 = j \le s \le J, \; i \in \mathbb{Z}_\Omega^j, \; m \in \mathbb{Z}_\Omega^s \\ R_{i,m}^{0,s}, & j = 0, \; 0 \le s \le J, \; i \in \mathbb{Z}_\Omega^0, \; m \in \mathbb{Z}_\Omega^s \end{cases} \tag{3.87}$$

Finally we can evaluate the matrix C with:

$$C_{k,m}^{j,s} = \sum_{p \in \mathbb{Z}_\Omega^j} \left(A_{k,p}^{j,j} \right)^{-1} \Delta_{p,m}^{j,s}, \qquad 0 \le j \le s \le J, \; k \in \mathbb{Z}_\Omega^j, \; m \in \mathbb{Z}_\Omega^s \tag{3.88}$$

We can now evaluate the coefficients following equation 3.80. With all indices we have

$$c_k^i = \sum_{m \in \mathbb{Z}_\Omega^s} C_{k,m}^{j,s} u_m^s, \qquad 0 \leq j \leq s \leq J, k \in \mathbb{Z}_\Omega^j. \tag{3.89}$$

With equation 3.89 we have the formal mechanism to calculate the coefficients of all levels. In the following figures we show the approximations of the higher levels up to level 4.

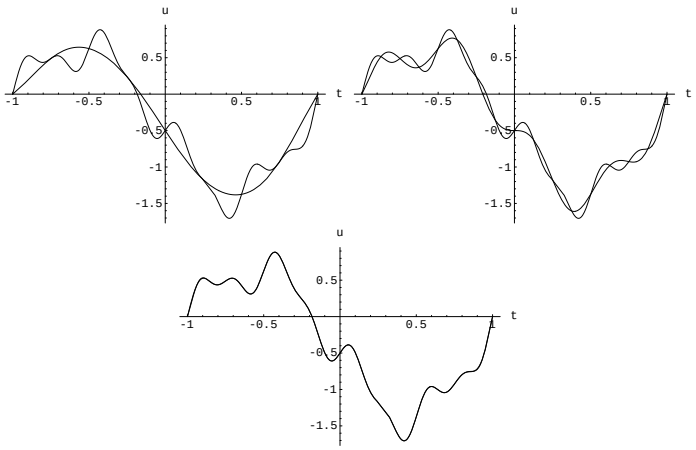

Figure 41: Approximation levels 2,3,4.

We realize that already at level $j = 4$ the difference between the approximation and the original function is not visible any more. Equations 3.86 - 3.89 represent the basic algorithm of wavelet collocation. It is interesting to look at the errors of the approximation compared to the original function 3.69. This is shown in figure 42.

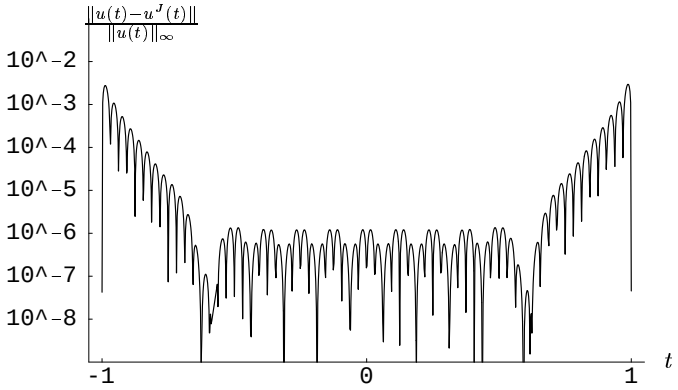

Figure 42: Relative error of approximation.

It can be seen that the relative error of the approximation is not uniformly distributed over the interval. The largest errors occur at the boundaries.

In the following the algorithm is refined by using additional wavelets at the boundaries to enhance the accuracy.

3.1.3 The Introduction of Wavelets at the Boundaries

Up to now we only considered wavelets positioned inside the domain. In order to enhance the accuracy of the approximation at the boundaries we can additionally consider the influence of wavelets placed outside the domain. Due to the finite support there is a finite number of wavelets outside the domain, whose influence inside the domain can be accounted for.

We conclude that for any level of resolution j, there is a subset of integers:

$$k \in \mathbb{Z}_\Omega^j : \qquad -2^{L+j-1} - N_l, \cdots, 2^{L+j-1} + N_r, \tag{3.90}$$

such that the wavelet Ψ_k^j with $k \in \mathbb{Z}_\Omega^j$ affects the interior of the domain. N are the additional wavelets outside the domain, to the left (index l) and to the right (index r). Up to now we placed the wavelets corresponding to the collocation points, see figure 35 and 37.

According to Vasilyev et al. (1995) the wavelets outside the domain are associated with additional collocation points inside the domain. This is shown in the following figure 43

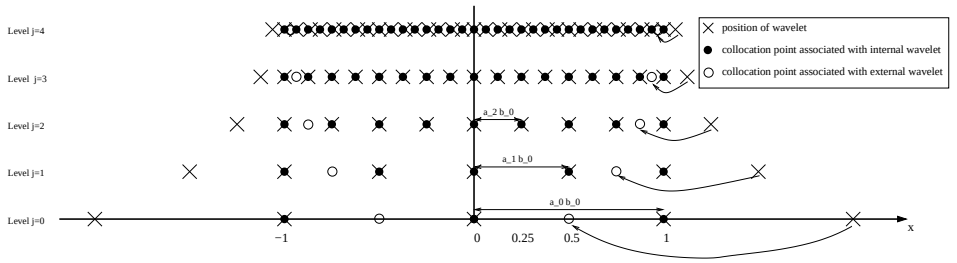

Figure 43: Placement of wavelets outside the domain and associated collocation points inside the domain (marked with arrows). Example with one external wavelet $N = 1$.

The positions of regular collocation points associated with external wavelets at a specified level, are chosen to be collocation points of the next higher scales. This simplifies somewhat the construction of the vectors. We can enhance the accuracy by taking more external wavelets. In figure 44 we show the correlation with two external wavelets.

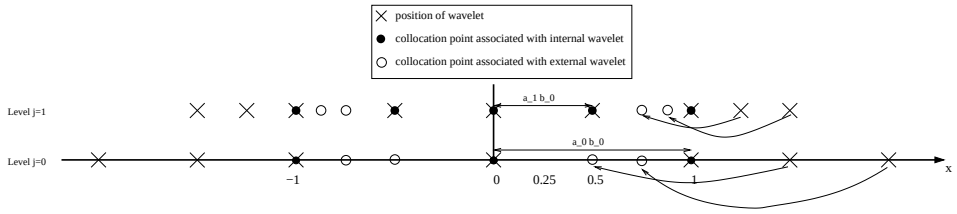

Figure 44: Placement of wavelets outside the domain and associated collocation points inside the domain (marked with arrows). Example with two external wavelet $N = 2$.

Usually not more than two boundary wavelets yield sufficient accuracy. The additional wavelets enter the algorithm at the point where the matrix A is constructed from the placement of the wavelets , see figure 37 and equation 3.71. By adding wavelets outside the domain whose support does not intersect with the domain of

interest the matrix A becomes ill-conditioned and difficult to invert. The condition number of the matrix rises with increasing N.

The enumeration of the collocation points is done in such a way that for any level j ($0 \leq j \leq J$) and $i, k \in \mathbb{Z}^J$, $t_i^j < t_k^j$ if $i < k$. This ensures a subsequent enumeration of the collocation points from one boundary to the other and simplifies the usage of restriction and prolongation operators. One can show that according to equation 3.90:

$$t^j_{-2^{L+j-1}-N_l} = t_l, \qquad t^j_{2^{L+j-1}+N_r} = t_r \quad . \tag{3.91}$$

The algorithm to reconstruct the function with additional boundary wavelets remains as in the previous section. At the beginning of the algorithm the modified vector Ψ and the vector of the collocation points has to be constructed according to the additional wavelets and collocation points.

To show the benefit of additional boundary wavelets we can take again the relative error of the approximation of function 3.69, see figure 45.

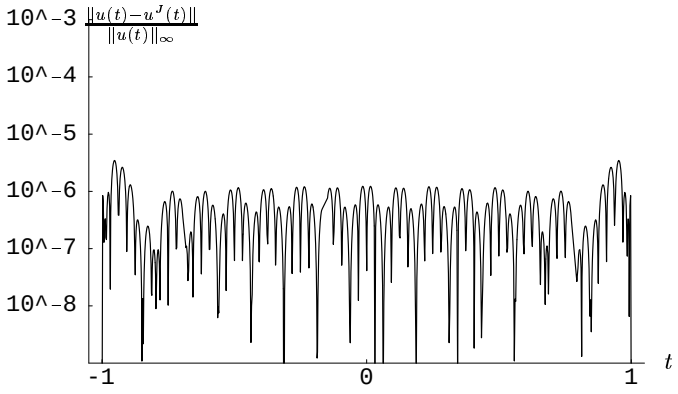

Figure 45: Relative error of approximation with two boundary wavelets.

Compared with figure 42 we have a significant improvement at the boundaries.

In Vasilyev et al. (1995) a detailed reflection for the errors with different numbers of boundary wavelets is given. Also a different choice of the mother wavelet, e.g. the autocorrelation Daubechies wavelet is considered.

3.2 Solving Partial Differential Equations

If we want to solve differential equations, we need to know how to handle derivatives with wavelets. We can introduce derivation-operators to deal with them.

First of all we introduce the interpolation operator. We multiply equation 3.89 with our wavelet $\Psi_k^j(t)$ on hand and obtain:

$$u^J(t) = \sum_{i \in \mathbb{Z}_\Omega^j} I_i(t) u_i^J, \tag{3.92}$$

with

$$I_i(t) = \sum_{j=0}^{J} \sum_{k \in \mathbb{Z}_\Omega^j} \Psi_k^j(t) C_{k,i}^{j,J}, \qquad \text{with} \qquad i \in \mathbb{Z}_\Omega^J. \tag{3.93}$$

The interpolation operator I gives the values not only on the collocation points u_i^J but yields to a continuous representation due to the continuous wavelets for all t.

In a similar way we can define a derivation operator. We multiply equation 3.89 with the m'th derivation of the wavelet and obtain:

$$u^{J(m)}(t) = \sum_{i \in \mathbb{Z}_\Omega^j} D_i^{(m)}(t) u_i^J \tag{3.94}$$

with

$$D_i^{(m)}(t) = \sum_{j=0}^{J} \sum_{k \in \mathbb{Z}_\Omega^j} \Psi_k^{j(m)}(t) C_{k,i}^{j,J}, \qquad \text{with} \qquad i \in \mathbb{Z}_\Omega^J. \tag{3.95}$$

We can proceed as long as the derivation of the wavelet Ψ is continuous to obtain the continuous representation of the derivation of u. Note that $D_i^{(0)}(t) = I_i(t)$.

Now we have all prerequisites to deal with differential equations. We shall show the method through its application to the solution of a second order partial differential equation:

$$\frac{\partial u}{\partial t} = F(t, x, u, u_x, u_{xx}) \qquad \text{with} \quad t = 0, u(x, 0) = u_0(x), \tag{3.96}$$

where F is a linear or nonlinear operator.

Note that we now switch to other dependent variables for the wavelets and all operators. The wavelets are now dependent from the location x. The previous notation was used for a more comprehensible explanation of the similarities to signal processing where the signals are often time-dependent.

We want to solve in the interior of the domain $x_l < x < x_r$ where we have with the appropriate boundary wavelets the following collocation points :

$$x_l = x^j_{-2^{L+j-1}-N_l}, \qquad x_r = x^j_{2^{L+j-1}+N_r} \quad , \tag{3.97}$$

see also equation 3.92. We can now set up a system of differential equations at every collocation point inside the domain. With help of the previously defined derivation operator 3.95 we eliminate the first and second derivations in space and end up with a system of ordinary differential equations in time at every collocation point. These can be solved by standard methods. The equations are solved on all internal points but not on the collocation points of the boundary where the boundary conditions are valid. In case of Neumann or mixed boundary conditions the values at the boundary collocation points are expressed in terms of the values inside the considered domain.

3.2.1 The 1D Heat Equation with Internal Heating

A one-dimensional transient test problem where an analytical solution is known is that of a rod with uniform volumetric heating. This simulation may not show directly the strengths of the wavelet collocation in resolving steep gradients but it gives an impression of the resolution which is necessary to solve this problem. One may compare the effort with other methods. The problem is described by:

$$\frac{\partial T}{\partial t} = \frac{1}{\kappa} \frac{\partial^2 T}{\partial x^2} + \dot{Q} \tag{3.98}$$

The analytical solution in dimensionless form is given in Carslaw and Jaeger (1959). With the help of the derivation operators we can rewrite the heat equation as follows.

$$\frac{\partial}{\partial t} T_i^J(t) = \sum_{k \in \mathbb{Z}_\Omega^J} \frac{1}{\kappa} D_{i,k}^{(2)} T_k^J(t) + \dot{Q}_k, \tag{3.99}$$

$$T_i^J(0) = 0, \tag{3.100}$$

$$T_{\pm(2^{L+J-1}+N)}(t) = 0, \tag{3.101}$$

where the number of equations is $i = -2^{L+J-1} - N + 1, \cdots, 2^{L+J-1} + N - 1$, i.e. the equations are solved on all internal collocation points, at the two boundary points the boundary values are fixed. Since the gradients in the solution are quite smooth it is possible to take only eight wavelets for the simulation of the whole rod. Since the solution is symmetric we show half of the solution in figure 46.

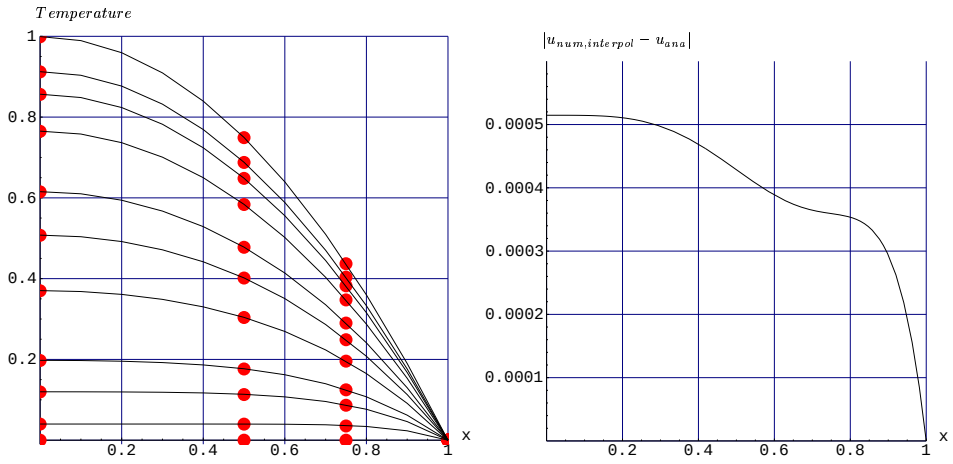

Figure 46: left:Numerical solution (dots) and analytical solution (lines) at different times.The dots also show the distribution of the wavelets on the x-axis. Right: Absolute difference between continuous wavelet-interpolation of numerical solution and analytical solution at time $t = 300$.

Even with such few wavelets no deviation to the analytical solution is visible in figure 46(left). Due to the fact that the wavelets are continuous we can interpolate the values between the collocation points. For interpolation we simply take the wavelet interpolation operator defined in equation 3.93 to compute the values between the collocation points and compare with the analytical solution. In figure 46 (right) we plotted the absolute value of the difference between the interpolated numerical and the analytical solution. We realize that the error at the boundary ($x = 1$) is zero due to the fixed boundary value. The highest error accumulates in the middle at $x = 0$. If we take only two wavelets more the overall error is on the order of 10^{-6}.

It is interesting to note that the algorithm does not fail even if we take only six wavelets for the whole rod and leave the boundary wavelet we used in the present simulation.

3.2.2 Burgers Equation with Low Viscosity

As a test problem for the numerical algorithm we consider Burger's equation with small viscosity:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \qquad \text{with} \quad x \in (-1, 1), t > 0 \tag{3.102}$$

with initial and boundary conditions

$$u(x, t = 0) = -sin(\pi x) \qquad \text{and} \qquad u(\pm 1, t) = 0. \tag{3.103}$$

The analytical solution of this problem is known, see Cole (1951). The solution develops into a saw-tooth wave centered at $x = 0$. Figure 47 shows the solution at several times for a small viscosity $\nu = 10^{-2}/\pi$.

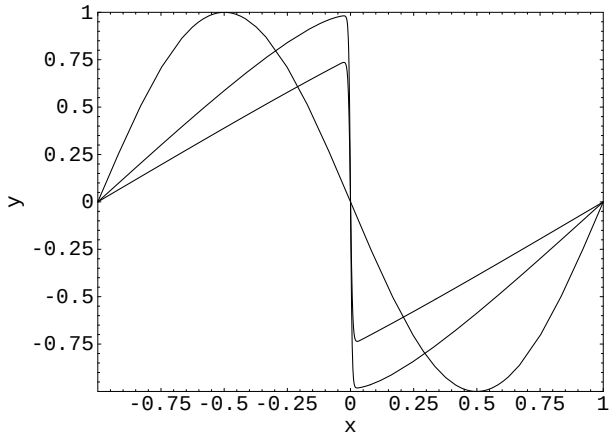

Figure 47: Analytical solution following Cole (1951) at times $t = 0, 1.6/\pi, 3/\pi$.

The gradient at the origin reaches its maximum $|\partial u/\partial x|_{max} = 152.0051616$ at time $t_{max} = 1.60369/\pi$. From the study of Basdevant, Deville, P.Haldenwang, Lacroix, Ouzzani, Peyret, Orlandi and Patera (1986) can be concluded that the performance of a numerical method can be judged from its ability to resolve the large gradient. The largest gradient should appear at the correct time.

With help of the derivation operator 3.95 we can rewrite the Burger's equation as follows:

$$\frac{\partial}{\partial t}u_i^J(t) = \sum_{k\in\mathbb{Z}_\Omega^J}[-u_i^J(t)D_{i.k}^{(1)} + \nu D_{i,k}^{(2)}]u_k^J(t), \tag{3.104}$$

$$u_i^J(0) = -sin(\pi x_i^J), \tag{3.105}$$

$$u_{\pm(2^{L+J-1}+N)}(t) = 0 \tag{3.106}$$

where the number of equations is $i = -2^{L+J-1} - N + 1, \cdots, 2^{L+J-1} + N - 1$, i.e. the equations are solved on all internal collocation points, at the two boundary points the boundary values are fixed.

In the following we show the solution without using boundary wavelets or an adaption of the grid. In figure 45 we show the distribution of the collocation points and the transient evolution of the initial condition 3.105.

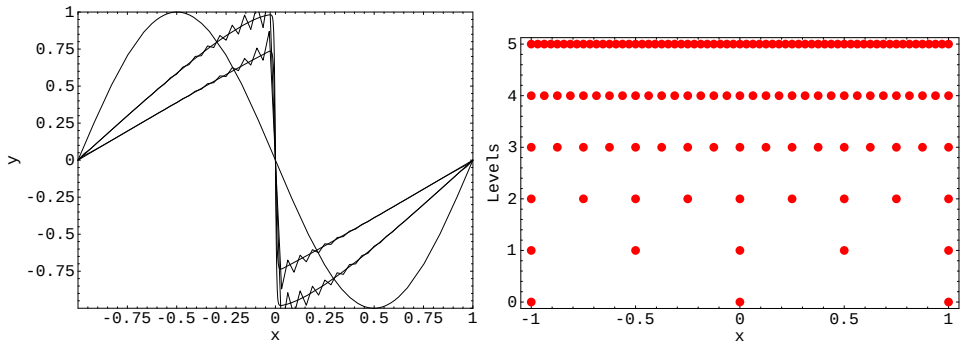

Figure 48: Left: Comparison of analytical solution following Cole (1951) with the wavelet collocation method using the 'Mexican' hat wavelet at times $t = 0, 1.6/\pi, 3/\pi$. Right: Distribution of collocation points.

We realize that we obtain oscillations in the neighborhood of the shock, when the gradient becomes steep. In this example we used five levels to resolve the gradients which is obviously not sufficient. In Vasilyev et al. (1995) a detailed survey of the results is given for different values of the used parameters and boundary wavelets.

Following Vasilyev et al. (1995) we need approximately nine levels of collocation points to resolve the length-scale of the developing shock without oscillations. We need the additional wavelets only in the vicinity of the shock to build the steep gradients. In the rest of the domain the wavelet coefficients will be close to zero and so we will have a big overhead of not necessary computations if we use the algorithm in the present form.

3.2.3 The Dynamically Adaptive Wavelet Collocation Method

The adaptive wavelet collocation method gets rid of non-necessary computations at collocation points where the coefficients will be close to zero. The most straightforward method is to simply omit collocation points and wavelets where the wavelet-coefficient is below a specified threshold. Indeed this is the guiding principle of the adaptive multilevel wavelet collocation following Vasilyev and Paolucci (1996a).

The magnitude of the wavelets of each level is of the order $a(j)^{-1/2}$, see equation 3.64. So we retain only those wavelets whose amplitude satisfy the criteria :

$$|c_k^j| \geq a(j)^{1/2}\epsilon \tag{3.107}$$

By omitting the wavelet the corresponding collocation point is also omitted.

Due to the collocation nature of the algorithm we can omit all collocation points of higher levels which we have already considered at the same position in lower levels, since the wavelet coefficients at these points are zero. This a priori knowledge can be used to avoid an extensive operation count in testing the coefficients.

Since the solution evolves in time we should also include wavelets in adjacent zones since their coefficients may become important at the next time interval. The adjacent zone is defined as follows:

$$|s - j| \leq M, \qquad |x_k^s - x_i^j| \leq C a_j, \tag{3.108}$$

Where C defines the width of the adjacent zone and M determines the extend to which coarser and finer scales are included into the adjacent zone. The considered time-interval Δt must be chosen in a way that the solution does not move beyond the border of the adjacent zone within Δt.

By this method all structures appearing in a solution can be tracked by adding or omitting wavelets following the above criteria. We implemented the collocation point adaption in the algorithm.

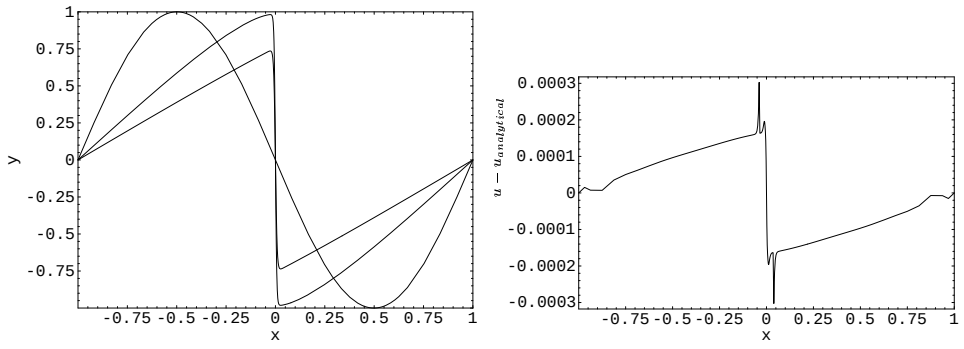

Figure 49: Left: Comparison of analytical solution following Cole (1951) with the wavelet collocation method using the 'Autocorrelation Daubechies' wavelets at times $t = 0, 1.6/\pi, 3/\pi$. No difference is visible. Right: Absolute error at $t = 3/\pi$.

In figure 49 we show the comparison of the adaptive computation with the analytical solution for Burgers equation, see section 3.2.2. We used $\epsilon = 10^{-3}, M = 1, C = 2$ in equations 3.108 for the adaptive computation. We recognize in figure 49 that the curves show no visible deviation compared to the analytical solution. For that reason it is more substantial to plot the difference between the solutions. We realize in figure 49 on the right that the absolute error never exceeds the bounds of the threshold parameter ϵ which controls the accuracy of the method. The biggest errors appear at the positions where the shock develops.

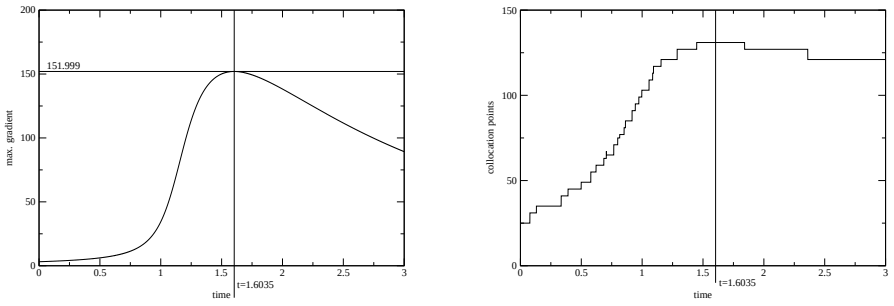

Figure 50: Left: Development of maximum gradient in time of the wavelet computation. Right: Evolution of the number of grid points in time.

In figure 50 we show the development of the maximum gradient and the number of collocation points in time. The maximum value of the gradient appears nearly exactly at the time predicted by the analytical solution of Cole (1951), see section 3.2.2. At the maximum gradient at time $t \approx 1.6035/\pi$ we need the most collocation points. An extensive investigation of the influence of the different parameters on the accuracy of the method is given in Vasilyev and Paolucci (1996a). Figure 51 shows the distributions of collocation points at different times.

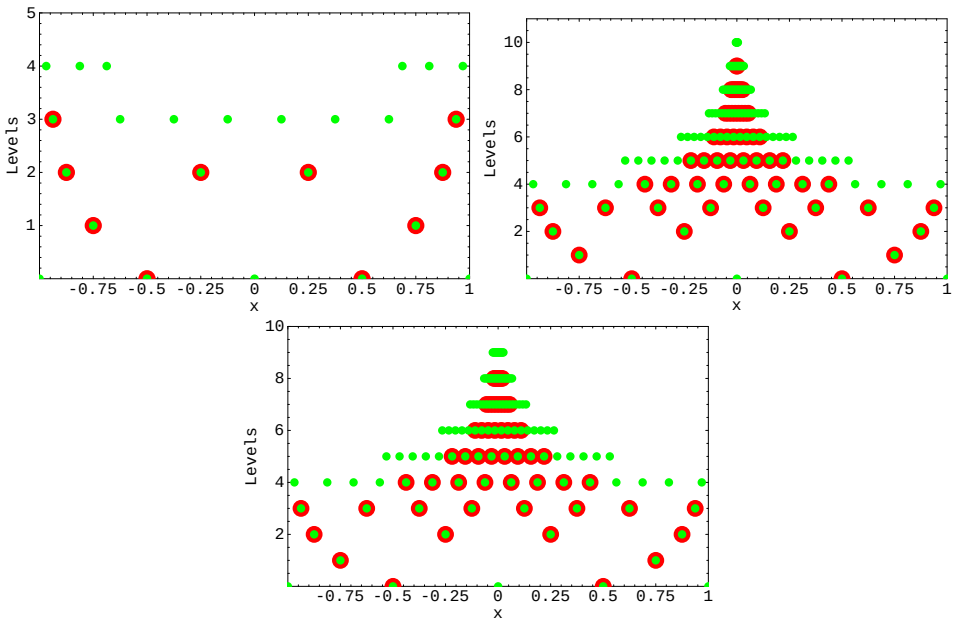

Figure 51: Development of the grid in time. Big dots show the skeleton grid where the wavelet coefficients fulfill equation 3.107. Small dots show the adjacent area. From left to right, $t = 0, 1.6/\pi, 3/\pi$.

In Basdevant et al. (1986) an interesting survey of the accuracy of spectral and finite difference solutions of Burgers equation is given. With regard to the accuracy of the solution the present algorithm requires substantial fewer degrees of freedom than the spectral and finite difference methods investigated in Basdevant et al. (1986).

It should be mentioned however that the number of collocation points is affected by the choice of the wavelet. The algorithm uses fewer collocation points if we use the autocorrelation Daubechies wavelets instead of the Mexican hat wavelet. The reason might be the smaller support of the autocorrelation Daubechies wavelets compared to the Mexican hat wavelet.

The results indicate that the method is suited to solve problems with local steep gradients. The computational grid and the associated wavelets can be adapted efficiently by using only one criteria for the wavelet coefficients.

3.2.4 The Inviscid Burgers Equation

As an ultimate test how far the adaptive algorithm is able to resolve gradients at a fixed number of levels we take Burgers equation without viscosity. We do not change the threshold parameter ϵ, M, C compared to the previous case. For the inviscid case a shock formation can be observed. Following LeVeque (1990) for a smooth initial data $u_0(x)$, e.g. the sinus wave, the wave will break at time

$$T_b = \frac{-1}{min[u_0'(t)]}. \tag{3.109}$$

At the point where the wave breaks the slope is infinity. Since we take a sinus wave as starting condition with minimum slope -1, we expect wave breaking after $t = 1$. Figure 52 shows the development of the maximum gradient and the number of grid points in time.

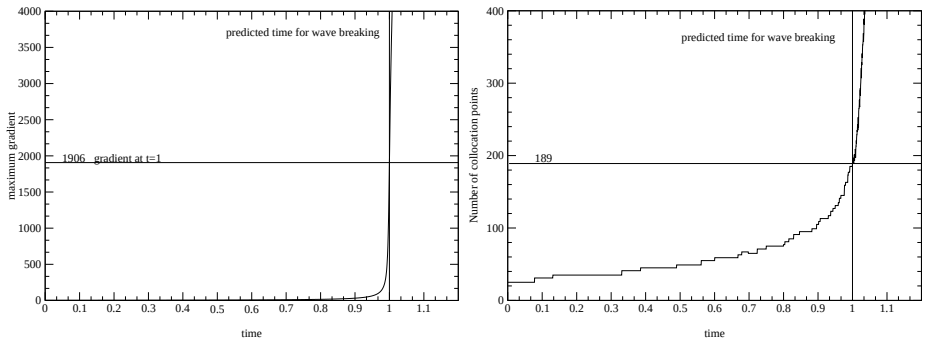

Figure 52: Left: Development of the maximum gradient in time. Right: Development of the grid points in time.

The wavelet collocation method identifies the infinite gradient near $t = 1$. The method shows a gradient of 1906 at $t = 1$. The gradient rises further after $t = 1$ and the method finally begins to oscillate and diverges at $t > 1.036$ since the number of new collocation points is not sufficient any more. Figure 53 shows the development of the solution at three different time steps. The solution at $t = 1.036$ already shows instabilities.

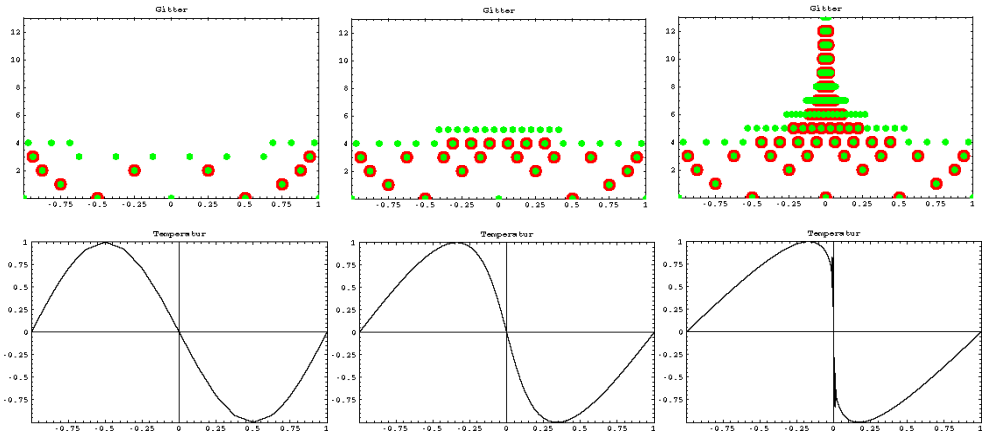

Figure 53: Burgers equation with zero diffusion at three different times, $t = 0, 0.5, 1.036$.

3.2.5 The Modified Burgers Equation

We consider the modified Burgers equation:

$$\frac{\partial u}{\partial t} + (v + u)\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2} \qquad \text{with} \quad x \in (-1, 1), t > 0 \tag{3.110}$$

with initial and boundary conditions

$$u(x, t = 0) = -tanh(\frac{x - x_0}{2\nu}) \qquad \text{and} \qquad u(\pm\infty, t) = \mp 1, \quad v = 1. \tag{3.111}$$

For small viscosity and a piecewise constant initial data we get the so called 'Riemann Problem'. The analytical solution for the shock wave moving with a uniform

velocity v is, see e.g. LeVeque (1990) :

$$u(x,t) = -tanh(\frac{x - x_0 - vt}{2\nu}). \tag{3.112}$$

The shock is transported as given in the analytical solution without visible numerical diffusion; see figure 54.

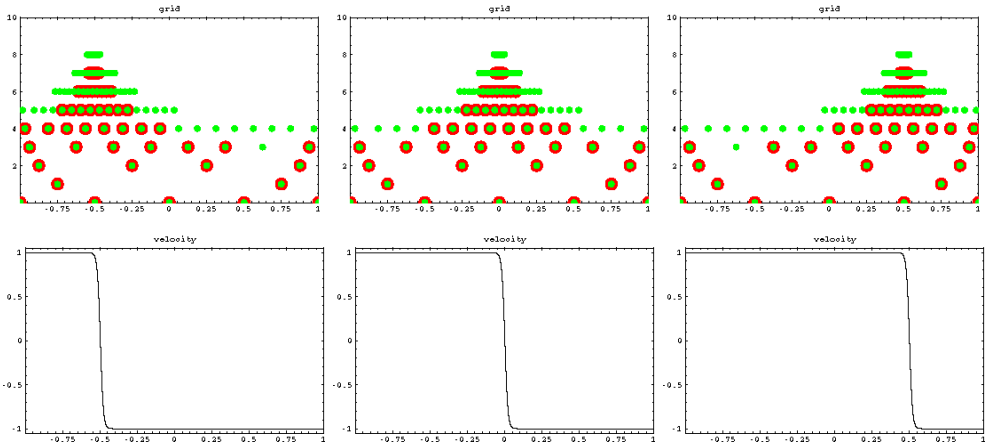

Figure 54: Modified Burgers equation at three different times, $t = 0, 0.5, 1$.

In figure 55 we realize that the number of collocation points oscillates around the average of N=101. The reason for this oscillations lies in the sensitivity of the method whether the shock is located close to a collocation point or between collocation points.

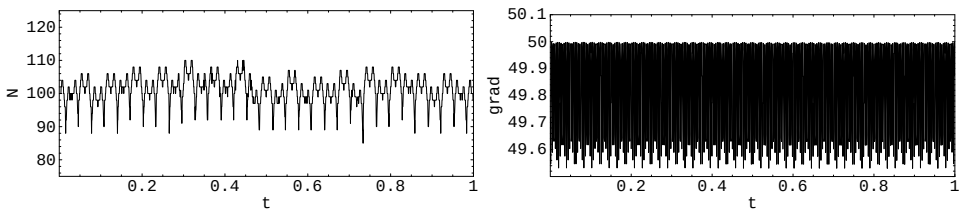

Figure 55: Left: Development of the number of collocation points in time. Right: Development of the maximum amplitude gradient in time.

In figure 55 on the right we plot the maximum gradient of the shock over the time. A decrease of the gradient, due to diffusion with time is not observable. The absolute value of the gradient oscillates to a certain extent between 49.998 and 49.5304 with a mean value of 49.84. The reason for the oscillation of the gradient is again the collocation nature of the algorithm. With the oscillation of the number of collocation points the ability of the algorithm to resolve the gradients also changes slightly.

Finally we can discuss the difference to the analytical solution after the time $t = 1$, see also figure 54. In figure 56 we subtracted the analytical solution (equation 3.112) from the numerical solution.

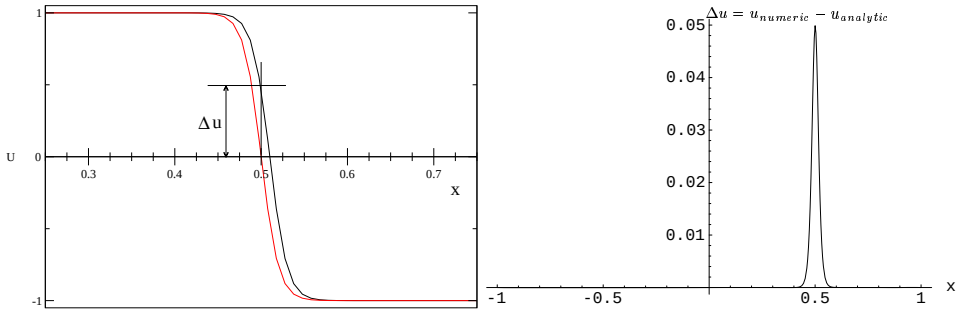

Figure 56: Difference between numerical solution and analytical solution at time $t = 1$. Left picture clarifies how the values on the right figure are evaluated.

We realize that the highest difference appears at the position of the shock. With the present high gradients a slight translation of the numerical solution to the analytical solution gives big differences. Out of the distance between the collocation points near the shock one can estimate that the numerical solution at time $t = 1$ is translated $\Delta x = 0.001$ in front of the analytical solution, which shows that the numerical algorithm transports slightly too fast.

3.2.6 Combustion of a gas

As a simple example we consider premixed gaseous combustion when a single species appears in a small quantity and limits the reaction. Similar examples can be found in the book of Zeldovich, Barenblatt, Librovich and Makhviladze (1985). The problem is governed by the energy equation and the conservation equation for the limiting species.

$$\frac{\partial T}{\partial t} - v\,\frac{\partial T}{\partial x} = -\kappa\,\frac{\partial^2 T}{\partial x^2} + \Omega \tag{3.113}$$

$$\frac{\partial C_O}{\partial t} - v\,\frac{\partial C_O}{\partial x} = -\frac{\kappa}{L}\,\frac{\partial^2 C_O}{\partial x^2} - \Omega \tag{3.114}$$

$$\Omega = C_O Exp[\frac{-N}{T}]\,K \tag{3.115}$$

$$K = Exp[N] \tag{3.116}$$

Where the velocity is $v = 0.7$, the thermal conductivity $\kappa = 0.03$ and the Lewis number $L = 4$. $N = 20$ is the dimensionless activation energy of the global Arrhenius reaction, K is the preexponential factor which we estimate as $K = Exp[N]$. T is the temperature and C_O is the concentration of the limiting component. The velocity v is the speed of the coordinate system relative to a fixed frame of reference. If this speed is equal to the propagation speed of the front the flame front appears at rest.

Due to the exponential terms a very steep combustion front evolves and travels through the domain. The fact that the front is not at rest shows that the estimated velocity of the reference frame does not coincide with the actual flame speed. We choose Dirichlet boundary conditions on both sides and a tanh function to build the initial conditions. Figure 57 shows the step like combustion front at different times.

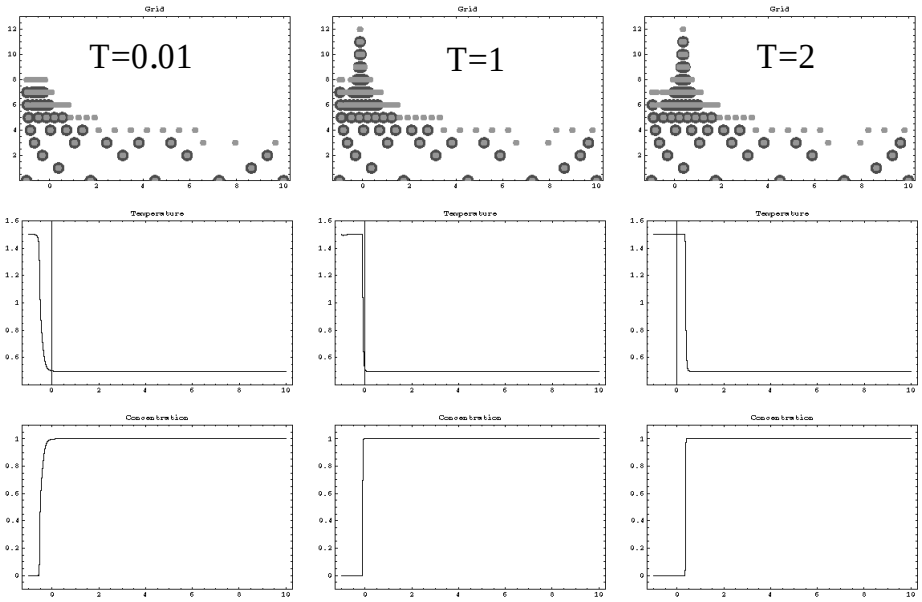

Figure 57: Plots for the grid, temperature and gas concentration at t=0.01 (initial conditions), t=1, t=2

In figure 58 we observe that the front is properly transported without visible diffusion.

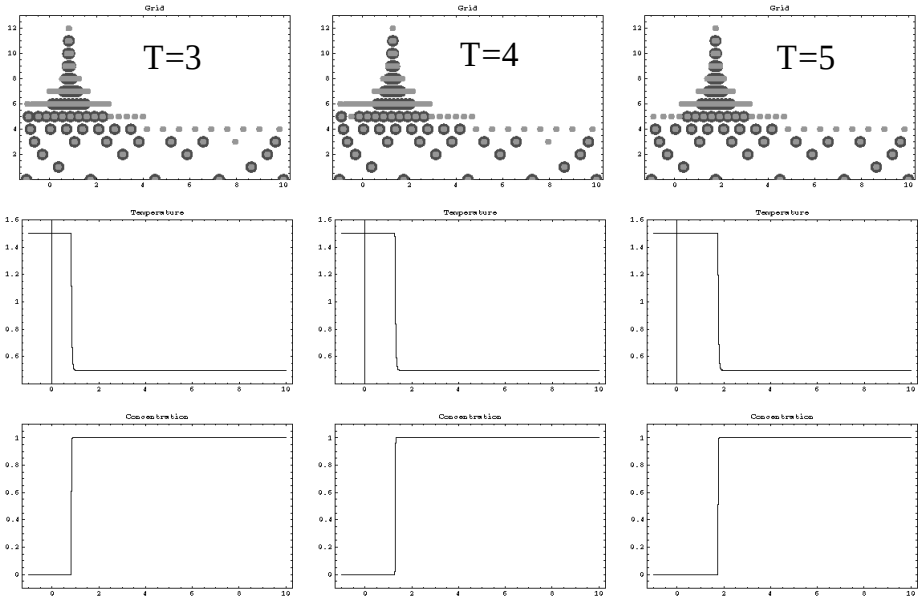

Figure 58: Plots for the grid, temperature and gas concentration at t=3, t=4, t=5

An interesting feature of this system is that instabilities of the front can occur if the Lewis number is sufficiently large. The front speeds up and slows down

periodicly in time. This can be observed in plots of the development of the number of grid points and the steepest gradient, see figure 59.

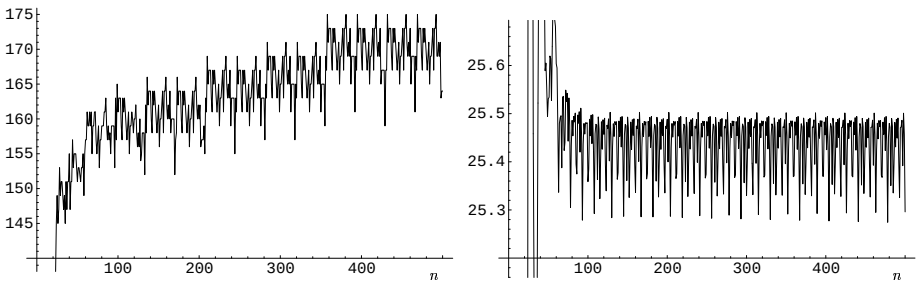

Figure 59: Left: Development of the number of grid points. Right: development of the maximum gradient with timesteps n.

Figure 59 also shows that the number of grid rises until we have a developed state.

3.2.7 Combustion of a solid

As an one dimensional example for the combustion of a solid we consider a gas (oxidizer) flowing through a porous media (fuel).

A similar problem was investigated in Schult, Matkowsky, Volpert and Fernandez-Pello (1995). The oxidizer gas convects with a velocity V_0 through the pores to the reaction zone where a exothermic reaction occurs between the solid and the oxidizer. The gas with oxidizer concentration Y_O flows through the unburnt part to the reaction zone where it reacts with the fuel of concentration Y_S. The burnt gas flows through the hot ash. If the inert part of the the fuel is big and the change of temperature is small, we can assume that the velocity of the gas flow remains constant. The reaction front initially moves in positive x direction. We have unburnt fuel at $x \to \infty$, ash and burnt gas at $x \to -\infty$. The boundary conditions at the unburnt side (index u) are:

$$x \to \infty : \; T = T_u, \; Y_S = Y_{Su}, Y_O = Y_{Ou} \tag{3.117}$$

For the burned side (index b) we have:

$$x \to -\infty : \; \frac{\partial T}{\partial x} = \frac{\partial Y_S}{\partial x} = \frac{\partial Y_O}{\partial x} = Y_S Y_O = 0, \tag{3.118}$$

$$T = T_b, \; Y_S = Y_{Sb}, \; Y_O = Y_{Ob} \tag{3.119}$$

The governing equations are:

$$\frac{\partial Y_S}{\partial t} = -W \tag{3.120}$$

$$\frac{\partial (Y_S \rho_S c_{pS} + \rho_i c_{pi}) T}{\partial t} = \lambda_S \frac{\partial^2 T}{\partial x^2} + W \rho_S \Delta H \tag{3.121}$$

$$\frac{\partial Y_O}{\partial t} + V_0 \frac{\partial Y_O}{\partial x} = D \frac{\partial^2 Y_O}{\partial x^2} - W \frac{\rho_S}{\rho_O} \frac{M_O}{M_S} \tag{3.122}$$

$$W = k Y_S Y_O \; exp(\frac{-Ea}{R\,T}) \frac{\rho_O}{M_O}. \tag{3.123}$$

Y_S is the concentration of the solid fuel, ρ_S and c_{pS} are the corresponding density and heat capacity respectively. The density and heat capacity of the inert part of

the solid fuel are ρ_i and c_{pi}. The temperature of the solid and the oxidizer gas are considered as equal. The heat conductivity of the solid is λ_S. The combustion enthalpy is ΔH which is related to the solid including the inert part of the solid. The concentration of of the oxidizer gas in the gas phase is Y_O and D is the corresponding molecular diffusivity in the gas phase. The reaction rate W is given by a Arrhenius law, where Ea is the activation energy, R is the gas constant, k is the preexponential factor.

We transform to moving coordinates which move at the velocity V from left right. The result is the stationary solution in moving coordinates:

$$y = x - Vt, \qquad \text{therefore } \frac{\partial}{\partial y}\frac{\partial y}{\partial t} = -V\frac{\partial}{\partial y} \tag{3.124}$$

The solution is at rest, so we get stationary equations:

$$-V\frac{\partial Y_S}{\partial y} = -W \tag{3.125}$$

$$-V\frac{\partial(Y_S\rho_S c_{pS} + \rho_i c_{pi})T}{\partial y} = \lambda_S\frac{\partial^2 T}{\partial y^2} + W\rho_S\Delta H \tag{3.126}$$

$$(V_O - V)\frac{\partial Y_O}{\partial y} = D\frac{\partial^2 Y_O}{\partial y^2} - W\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} \tag{3.127}$$

$$W = kY_S Y_O\, exp(\frac{-Ea}{R\,T})\frac{\rho_O}{M_O}. \tag{3.128}$$

Eliminating the reaction term W by adding equations 3.125 and 3.126 yields:

$$-V\frac{\partial Y_S}{\partial y}\rho_S\Delta H = -W\rho_S\Delta H \tag{3.129}$$

$$-V\frac{\partial(Y_S\rho_S c_{pS} + \rho_i c_{pi})T}{\partial y} = \lambda_S\frac{\partial^2 T}{\partial y^2} + W\rho_S\Delta H \tag{3.130}$$

$$-V\frac{\partial Y_S}{\partial y}\rho_S\Delta H - V\frac{\partial(Y_S\rho_S c_{pS} + \rho_i c_{pi})T}{\partial y} = \lambda_S\frac{\partial^2 T}{\partial y^2} \tag{3.131}$$

Similarly by subtracting equations 3.125 and 3.127:

$$-V\frac{\partial Y_S}{\partial y}\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} = -W\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} \tag{3.132}$$

$$(V_O - V)\frac{\partial Y_O}{\partial y} = D\frac{\partial^2 Y_O}{\partial y^2} - W\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} \tag{3.133}$$

$$(V_O - V)\frac{\partial Y_O}{\partial y} + V\frac{\partial Y_S}{\partial y}\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} = D\frac{\partial^2 Y_O}{\partial y^2} \tag{3.134}$$

In order to eliminate the derivatives, we integrate the equations 3.131 and 3.134 across the reaction zone under the simplifying condition that $V_O = const$:

$$-V\rho_S\Delta H Y_S - V(Y_S\rho_S c_{pS} + \rho_i c_{pi})T|_{y=-\infty}^{y=+\infty} = 0 \tag{3.135}$$

$$(V_O - V)Y_O + VY_S\frac{\rho_S}{\rho_O}\frac{M_O}{M_S}|_{y=-\infty}^{y=+\infty} = 0 \tag{3.136}$$

We now introduce the boundary conditions, see equations (3.117), (3.118) and (3.119):

$$-V\rho_S\Delta H Y_{Su} - V(Y_{Su}\rho_S c_{pS} + \rho_i c_{pi})T_u + V\rho_S\Delta H Y_{Sb} + V(Y_{Sb}\rho_S c_{pS} + \rho_i c_{pi})T_b = 0$$

$$(V_O - V)Y_{Ou} + VY_{Su}\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} - (V_O - V)Y_{Ob} - VY_{Sb}\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} = 0$$

From the above relations we evaluate the temperature of the burnt gas T_b and the ratio of the velocities:

$$T_b = \frac{\frac{\rho_S \Delta H(Y_{Su}-Y_{Sb})}{\rho_i c_{pi}} + \left(\frac{Y_{Su}\rho_s c_{pS}}{\rho_i c_{pi}} + 1\right) T_u}{\frac{Y_{Su}\rho_s c_{pS}}{\rho_i c_{pi}} + 1} \tag{3.137}$$

$$\frac{V_O}{V} = 1 - \frac{Y_{Su} - Y_{Sb}}{Y_{Ou} - Y_{Ob}} \frac{\rho_S}{\rho_O} \frac{M_O}{M_S} \tag{3.138}$$

For the special case of stoichiometric combustion we have $Y_{Sb} = Y_{Ob} = 0$ so that:

$$T_b = \frac{\rho_S \Delta H(Y_{Su} - Y_{Sb})}{\rho_i c_{pi}} + \left(\frac{Y_{Su}\rho_s c_{pS}}{\rho_i c_{pi}} + 1\right) T_u \tag{3.139}$$

$$\frac{V_O}{V} = 1 - \frac{Y_{Su}}{Y_{Ou}} \frac{\rho_S}{\rho_O} \frac{M_O}{M_S} \tag{3.140}$$

We introduce dimensionless variables for the stoichiometric case. The following reference values are used: reference velocity V, reference length $\lambda_S/(\rho_i c_{pi} V)$, reference time $\lambda_S/(\rho_i c_{pi} V^2)$, reference temperature $\Delta T = Y_{Su}\rho_S \Delta H/(\rho_i c_{pi})$. The following dimensionless groups appear: $A = Y_{Su}\rho_S c_{pS}/(\rho_i c_{pi})$, $v_O = V_O/V$, $L = \lambda_S/(\rho_i c_{pi} D)$.

For the dimensionless variables we use: time τ, length ξ, temperature $\theta = T/\Delta T$, concentration $C_S = Y_S/Y_{Su}$, $C_O = Y_O/Y_{Ou}$. Now we write the equations 3.120-3.122 in a dimensionless form:

$$\frac{\partial C_S}{\partial \tau} = -\Omega \tag{3.141}$$

$$\frac{\partial (C_S A + 1)\theta}{\partial \tau} = \frac{\partial^2 \theta}{\partial \xi^2} + \Omega \tag{3.142}$$

$$\frac{\partial C_O}{\partial \tau} + v_0 \frac{\partial C_O}{\partial \xi} = \frac{1}{L}\frac{\partial^2 C_O}{\partial \xi^2} - \Omega(1 - v_O)\frac{1 - C_{Ob}}{1 - C_{Sb}} \tag{3.143}$$

$$\Omega = \frac{\lambda_S}{\rho_i c_{pi} V^2}\frac{1}{Y_{Su}} W = K C_S C_O \, exp(\frac{-N}{\theta}) \tag{3.144}$$

$$N = \frac{-E_a}{R\Delta T}, K = k\frac{\lambda_S Y_{Ou}}{\rho_i C_{pi} V^2}\frac{\rho_O}{M_O}. \tag{3.145}$$

From the solution in moving coordinates (3.137),(3.138) we get in dimensionless form:

$$\theta_b = \frac{(1 - C_{Sb}) + (A+1)\theta_u}{C_{Sb}A + 1} \tag{3.146}$$

$$v_O = 1 - \frac{Y_{Su}}{Y_{Ou}}\frac{1 - C_{Sb}}{1 - C_{Ob}}\frac{\rho_S}{\rho_O}\frac{M_O}{M_S} \tag{3.147}$$

In the special case of a stoichiometric combustion we have:

$$C_{Ob} = C_{Sb} = 0 \tag{3.148}$$

$$\theta_b = 1 + (A+1)\theta_u \tag{3.149}$$

The system of equations (3.141) -(3.145) is solved with the wavelet collocation algorithm. For a preliminary testing and demonstration we simply set some typical values for the variables:

$$A = 0.5, \quad C_{Ob} = C_{Sb} = 0, \quad L = 4, \quad V_O = -5, \quad K = 1, \quad N = 2, \quad \theta_u = 0.05. \tag{3.150}$$

For solutions of a real media we should take the real material properties. We set the following initial conditions for temperature and concentrations:

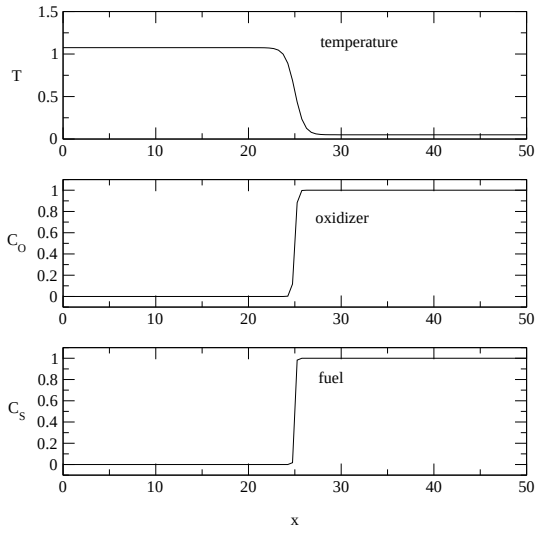

Figure 60: Initial conditions for temperature and concentration of oxidizer and fuel. The results are shown in the following figures:

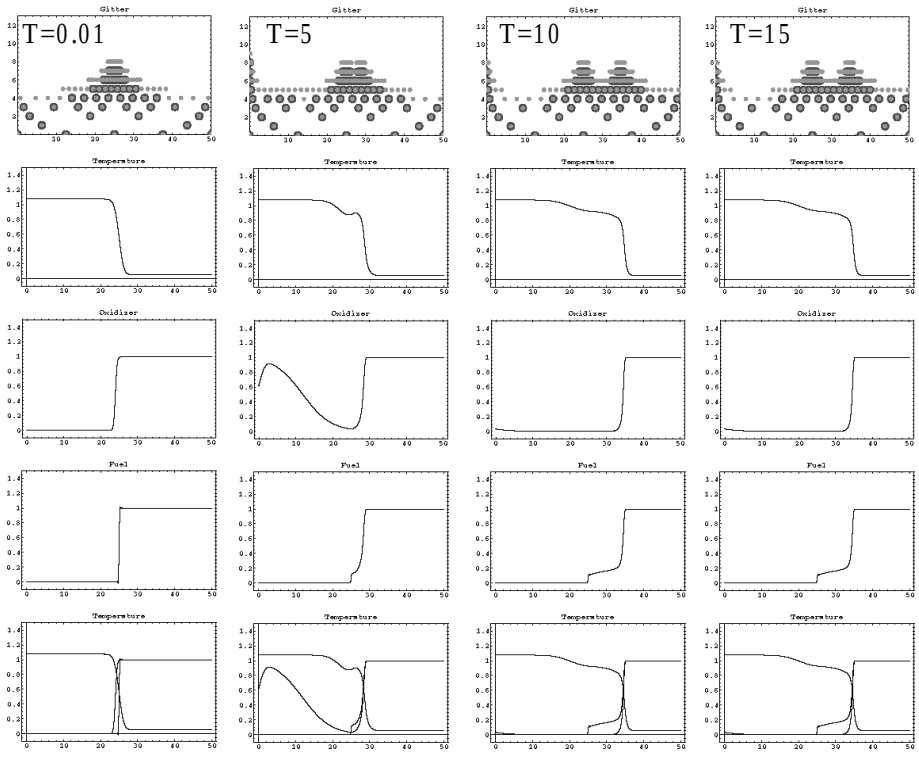

Figure 61: Plots of grid, temperature and concentration of oxidizer and fuel for t=0.01, t=5, t=10, t=15. The undermost picture shows the curves of temperature, oxidizer and fuel in one plot.

In figure 61 we observe steep fronts moving from the left to the right side. Although the fuel equation has no diffusive term the solution can be tracked without visible oscillations. We further observe that the fuel is not burnt completely. This results in a reduced burnt temperature and a reduced conversion of the solid fuel. When the fronts arrive at the right boundary of the computing domain, the direction of the combustion wave reverses and the front moves in the opposite direction. This can be seen in figure 62.

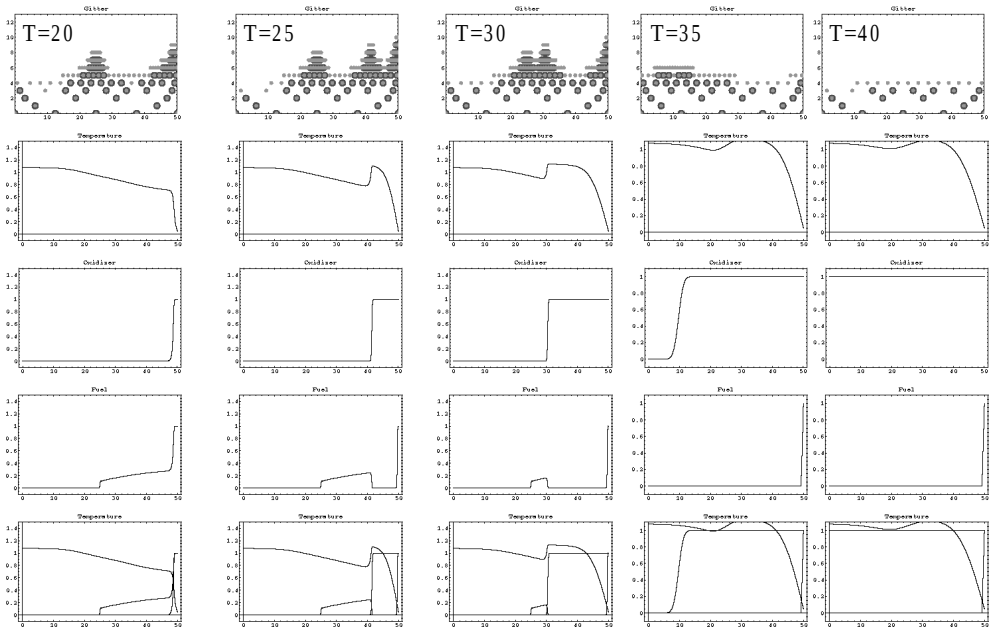

Figure 62: Plots of grid, temperature and concentration of oxidizer and fuel for t=20, t=25, t=30, t=35, t=40. The undermost picture shows the curves of temperature, oxidizer and fuel in one plot.

With the backward traveling wave we can see the so called super-adiabatic effect. The oxidizer gas is now heated from the hot burnt fuel until it reaches at the reaction zone. There due to the exothermic reaction the temperature rises above the adiabatic combustion temperature. When the rest of the fuel is burnt the combustion terminates.

In the next example we consider the case where the oxidizer appears in excess. The values for the constants are:

$$A = 1.0, \quad C_{Ob} = 0.5, C_{Sb} = 0, \quad L = 20, \quad V_O = -100, \quad K = 100, \quad N = 2, \quad \theta_u = 0.05. \tag{3.151}$$

The first results are shown in figure 63. One can see that the gradients are now much steeper than in the previous case. The fuel is not burnt completely. Like in the previous case the front travels to the right boundary and reverses the direction showing the super adiabatic effect (not shown here).

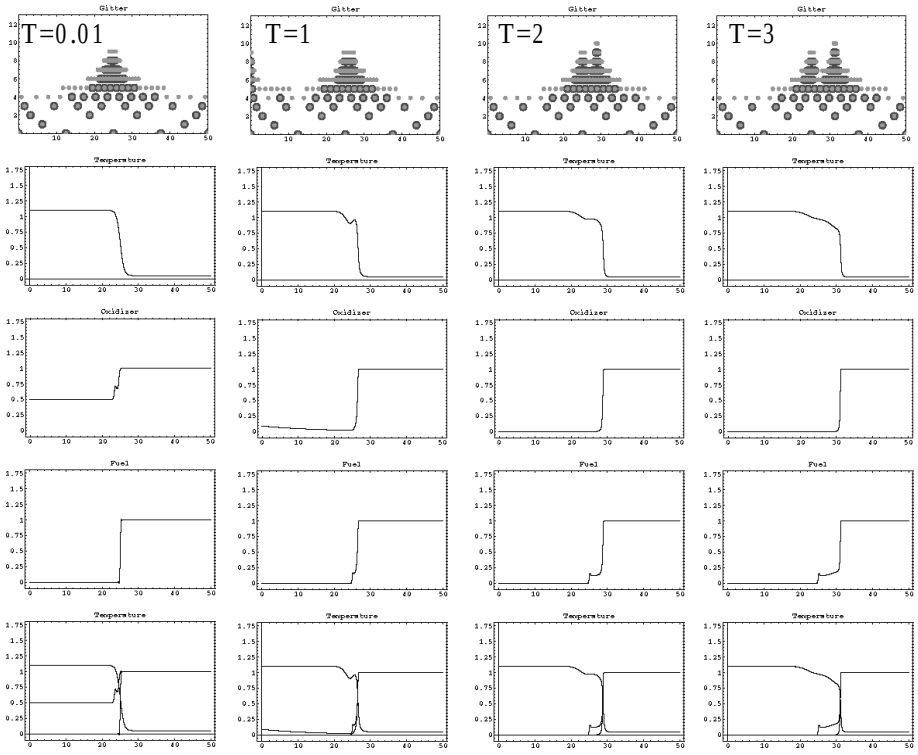

Figure 63: Plots of grid, temperature and concentration of oxidizer and fuel for t=0.01, t=1, t=2, t=3. The undermost picture shows the curves of temperature, oxidizer and fuel in one plot.

4 Conclusions, outlook

The wavelet collocation method following Vasilyev et al. (1995) and Vasilyev and Paolucci (1996a) was successfully implemented in Mathematica notebooks. The examples show the advantages of the method in resolving locally steep gradients. One goal of this work was the simulation of a traveling combustion front through a particle. Here the method could be applied to the best advantage. The particle combustion model solved with the wavelet collocation method is intended for the implementation in the TOSCA software package for a fast and accurate computation of the particle combustion in a packed bed. The work so far gives the framework for a further implementation in any programming language. Other areas where the method could be advantageous is the numerical simulation of microfluid dynamics where the focus of interest is the very thin electric double layer at the walls. Also in the field of magnetohydrodynamics boundary layers, especially the very thin Hartmann-layers, play a decisive role in the physical understanding of such flows.

The method is extendable to two and three dimensions. Publications of e.g. Farge, Kevlahan, Perrier and Schneider (1999), Griebel and Koster (2000), or others show the advantages for the simulation of the unsteady Navier-Stokes equations and turbulence analysis.

Acknowledgements

The authors gratefully acknowledge valuable help by Andreas Class in preparing the last sections of this report.

References

Basdevant, C., Deville, M., P.Haldenwang, Lacroix, J., Ouzzani, J., Peyret, R., Orlandi, P. and Patera, A.: 1986, "spectral and finite difference solutions of the Burgers equation", Computers and Fluids **14**(1), 23–41.

Beylkin, G. and Saito, N.: 1992, Wavelets, their autocorrelation functions, and multiresolution representation of signals, Intelligent Robots and Computer Vision XI: Biological, Neural Net and 3-D Methods, Vol. 1826 of Proceedings of the SPIE, pp. 39–50.

Carslaw, H. S. and Jaeger, J. C.: 1959, Conduction of Heat in Solids, Oxford at the Clarendon Press.

Chui, C. K.: 1992, An Introduction to Wavelets, Vol. 1 of Wavelet Analysis and its Applications, Academic Press, Inc.

Cole, J.: 1951, On a quasi-linear parabolic equation occuring in aerodynamics, Quart. Appl. Math. **9**, 225–236.

Daubechies, I.: 1992, Ten Lectures on Wavelets, Vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia.

Farge, M., Kevlahan, N., Perrier, V. and Schneider, K.: 1999, Wavelets in Physics, Cambridge University Press, chapter Turbulence analysis, modelling and computing using wavelets, pp. 117–200.

Griebel, M. and Koster, F.: 2000, Adaptive wavelet solvers for the unsteady incompressible Navier Stokes equations, in J. Malek, J. Necas and M. Rokyta (eds), Advances in Mathematical Fluid Mechanics, Lecture Notes of the Sixth International School "Mathematical Theory in Fluid Mechanics", Paseky, Czech Republic, September 1999, Springer Verlag. also as Report SFB 256 No. 669, Institut für Angewandte Mathematik, Universität Bonn, 2000.

Haar, A.: 1910, Zur Theorie der orthogonalen Funktionensysteme, Mathematische Annalen **69**, 331–371. In German.

He, Y.: 1996, Mathematica Wavelet Explorer, Wolfram Research.

Kaiser, G.: 1994, A Friendly Guide to Wavelets, Springer-Verlag, New York.

LeVeque, R. J.: 1990, Numerical Methods for Conservation Laws, Birkhäuser.

Schult, D., Matkowsky, B., Volpert, V. and Fernandez-Pello, A.: 1995, Propagation and extiction of forced opposed flow smolder waves, Combustion and Flame **101**, 471–490.

Vasilyev, O. and Bushe, W.: 1998, On the use of a dynamically adaptive wavelet collocation algorithm in dns of non-premixed turbulent combustion, Annual Research Briefs, Center For Turbulence Research, Nasa Ames/Stanford Univ, pp. 65–81.

Vasilyev, O. and Paolucci, S.: 1996a, A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain, J. Comp. Phys. pp. 498–512.

Vasilyev, O. and Paolucci, S.: 1996b, A fast adaptive wavelet collocation algorithm for multidimensional PDE's, J. Comp. Phys. pp. 498–512.

Vasilyev, O., Paolucci, S. and Sen, M.: 1995, A multilevel wavelet collocation method for solving partial differential equations in a finite domain, J. Comp. Phys. **120**, 33–47.

Vasilyev, O., Yuen, D. and Paolucci, S.: 1997, The solution of PDEs using wavelets, Computers in Phys. **11**(5), 429–435.

Zeldovich, Y. B., Barenblatt, G., Librovich, V. and Makhviladze, G.: 1985, The mathematical theory of combustion and explosions, Consultants Bureau.

Index