Doctoral Thesis

Space-Time Adaptive Resolution for Reactive Flows

Author:
Simone Gemini

Supervisor:
Prof. Mauro Valorani

Department of Mechanical and Aerospace Engineering

Sapienza University of Rome, IT

XXXII Ciclo 2016-2019

SAPIENZA

Abstract

Sapienza University of Rome, IT
Department of Mechanical and Aerospace Engineering

Doctor of Philosophy in Aeronautical and Space Engineering

Space-Time Adaptive Resolution for Reactive Flows

by Simone Gemini

Multi-scale systems evolve over a wide range of temporal and spatial scales. The extent of time scales makes both theoretical and numerical analysis difficult, mostly because the time scales of interest are typically much slower than the fastest scales occurring in the system. Systems with such characteristics are usually classified as being stiff. An adaptive mesh refinement method based on the wavelet transform and the G-Scheme framework are used to achieve spatial and temporal adaptive model reduction, respectively, of physical problems described by PDEs. The combination of the methods is proposed to solve PDEs describing reaction-diffusion systems with the minimal number of degrees of freedom, for prescribed accuracies in space and time. Different reaction-diffusion systems are studied with the aim to test the performance and the capability of the combined scheme to generate accurate solutions with respect to reference ones. Several strategies are implemented to improve the performance of the scheme, with minimal loss of accuracy.

Acknowledgements

First of all, a special thanks to my mother. Words cannot express how grateful I am to you for all of the sacrifices that you have made on my behalf.

I would like to express my special appreciation and thanks to my advisor Prof. Mauro Valorani, in particular for his knowledge and patience. I would like to thank Prof. Samuel Paolucci, Dr. Pietro Paolo Ciottoli, Dr. Riccardo Malpica Galassi and Dr. Temistocle Grenga for their precious comments and suggestions.

At the end, I would like to thank Prof. Hong G. Im and Prof. Heinz Pitsch who gave me the opportunity to visit their research centers.

Contents

List of Figures

List of Tables

List of Abbreviations

AMR	**A**daptive **M**esh **R**efinement
ASIM	**A**pproximate **S**low **I**nvariant **M**anifold
CSP	**C**omputational **S**ingular **P**erturbation
DEs	**D**ifferential **E**quation**s**
DoFs	**D**egree**s** **o**f **F**redoom
DWT	**D**iscrete **W**avelet **T**ransform
EoS	**E**quation **o**f **S**tate
FF	**F**ast **F**iber
MRA	**M**ulti **R**esolution **A**nalysis
MSA	**M**ulti **S**cale **A**pproximation
ODEs	**O**rdinary **D**ifferential **E**quation**s**
OS	**O**perator **S**plitting
pWAMR	**p**arallel **W**avelet **A**daptive **M**ultiresolution **R**epresentation
PDEs	**P**artial **D**ifferential **E**quation**s**
RHS	**R**ight **H**and **S**ide
ROM	**R**educed **O**rder **M**odel
RMS	**R**oot **M**ean **S**quare
SIM	**S**low **I**nvariant **M**anifold
SWR	**S**parse **W**avelet **R**epresentation
WAMR	**W**avelet **A**daptive **M**ultiresolution **R**epresentation

List of Symbols

Main symbols for wavelet

$c_{l,k}$	Amplitude of the scaling function $\phi_{l,k}(x)$
$d_{l,k}$	Amplitude of the wavelet function $\psi_{l,k}(x)$
g_k	Filter coefficients defining the wavelet function
h_k	Filter coefficients defining the scaling function
J	Maximum resolution level
J_0	Minimum resolution level
L_w	Length of the wavelet support
N_w	Number of vanishing moments
N_A	Number of active grid points
N_B	Number of neighbouring grid points
N_E	Number of essential grid points
N_N	Number of non-essential grid points
N_T	Total number of grid points
N_0	Maximum index of the coarsest grid level
$\phi(x)$	Mother scaling function
$\psi(x)$	Mother wavelet function
$\phi_{l,k}(x)$	Scaling function at the resolution level l and location k
$\psi_{l,k}(x)$	Wavelet function at the resolution level l and location k
$\bar{p}$	Wavelet order
$\{V_l\}$	Sequence of subspaces whose scaling functions are basis
$\{W_l\}$	Sequence of subspaces whose wavelet functions are basis
Δx_{min}	Minimum grid size
ε	Wavelet threshold parameter
ε_a	Absolute value of the wavelet threshold parameter
ε_r	Relative value of the wavelet threshold parameter
$\mathcal{V}_E$	Set of essential grid points
$\mathcal{V}_B$	Set of neighbouring grid points
$\mathcal{V}_A$	Set of active grid points

Main symbols for the G-Scheme

t	Time
$\vec{x}(t)$	State vector
$\vec{f}$	Nonlinear vector field
$\Delta\vec{x}$	Perturbation vector
$\Delta\xi^i(\tau)$	Curvilinear coordinates
$J_{\vec{f}}$	Jacobian matrix of the nonlinear vector field
$\vec{a}_i(\tau)$	Right eigenvectors of $J_{\vec{f}}$
$\vec{a}^j(\tau)$	Left eigenvectors of $J_{\vec{f}}$
$A(\tau)$	Matrix enclosing the right eigenvectors of $J_{\vec{f}}$

$B(\tau)$	Matrix enclosing the left eigenvectors of $J_{\tilde{f}}$
λ_i	Eigenvalues of the Jacobian matrix
$\mathcal{T}_x$	Tangent space
$\mathbb{E}$	Invariant subspace
$\mathbb{H}$	Slow subspace (Head)
$\mathbb{A}$	Active subspace (Heart)
$\mathbb{T}$	Fast subspace (Tail)
$\Delta\vec{\xi}^{r}_{\text{SIM}}$	Algebraic correction in the slow reduced model
$\Delta\xi^{s}_{FF}$	Algebraic correction in the fast reduced model

Main symbols for the coupled scheme

π_w	Efficiency of the wavelet compression
π_{gs}	Efficiency of the G-Scheme
π_O	Overall compression degree

Main symbols for the Arneodo model

n_{eq}	Number of equations
u	First dependent variable - Concentration of a species
v	Second dependent variable - Concentration of a species
α	Parameter involved in the Arneodo system of equations
β	Parameter involved in the Arneodo system of equations
D	Parameter involved in the Arneodo system of equations
Δt	Integration time step
Δx_{min}	Minimum grid size
u^{ε}_{ss}	Adaptive steady state solution - u variable
v^{ε}_{ss}	Adaptive steady state solution - v variable
u^{ref}_{ss}	Reference steady state solution - u variable
v^{ref}_{ss}	Reference steady state solution - v variable
T_{pb}	Period for the periodic bursting behaviour

Main symbols for flamelet

z	Mixture fraction
Δz_{min}	Minimum grid size in the mixture fraction space
Y_i	Mass fraction of the i-th chemical species
X_i	Molar fraction of the i-th chemical species
χ	Scalar dissipation rate
$\dot{\omega}_i$	Production rate for the i-th species
$\dot{\omega}_T$	Rate of enthalpy production
ρ	Density
T	Temperature
p	Pressure
c_p	Constant-pressure specific heat
c_v	Constant-volume specific heat
h	Specific enthalpy of the mixture
h_i	Partial-mass enthalpy of i-th chemical species
N_s	Number of chemical species
t_{ss}	Time to reached the steady state
R_u	Universal gas constant
M_w	Molecular weight
T_c	Critical temperature of pure chemical species

p_c	Critical pressure of pure chemical species
ρ_c	Critical density of pure chemical species
a	Parameter involved in RK-PR Equation of State
b	Parameter involved in RK-PR Equation of State
δ_1	Parameter involved in RK-PR Equation of State
δ_2	Parameter involved in RK-PR Equation of State
$\alpha(T)$	Temperature correction factor
ω	Acentric factor
T/T_c	Reduced temperature
Z_c	Compressibility factor
$M_{w,i}$	Molecular weight of the i-th chemical species
$Z_{c,ij}$	Pseudo-critical parameter (compressibility factor)
$p_{c,ij}$	Pseudo-critical parameter (pressure)
$T_{c,ij}$	Pseudo-critical parameter (temperature)
ω_{ij}	Pseudo-critical parameter (acentric factor)
$\tilde{e}_i$	Partial-density internal energy of the i-th species
e	Mixture internal energy

CHAPTER 1

Introduction

Numerical simulations of mathematical models describing real-life phenomena involving simultaneous contribution of a wide range of spatial and time scales is particularly challenging, despite of the tremendous progress in computer science. Model order reduction allows to cut down the computational complexity of such problems through a reduction of the state space dimension, which is identified by the number of Degrees of Freedom (DoFs). This leads to the concept of Reduced Order Model (ROM), that can be built on assumptions and simplifications using physical insight or derived information. If uniform grid algorithms are used to solve the finest spatial scales, the number of DoFs becomes large with a consequent requirement for expensive computational resources. A possibility to overcome this limitation is represented by adaptive techniques, that provide advantages in terms of computational speed and storage over uniform grid algorithms. In this project a first numerical technique based on the wavelet transform - named WAMR (Wavelet Adaptive Multiresolution Representation) - is adopted to generate dynamically adaptive grids. This method is able to capture any desired accuracy and produce automatically verified solutions for problems described by Partial and Ordinary Differential Equations (PDEs/ODEs). A further opportunity to reduce the complexity of the problem arises looking at the distribution of the time scales of the system dynamics. The presence of fast time scales constrains the slow and active dynamics within an embedding of Slow Invariant Manifolds (SIM). This means that the number of DoFs to describe the system dynamics is extremely lower than that one required for a DNS or a LES. The identification of a SIM and the creation of a ROM describing the slow dynamics constrained to evolve within the SIM is the task of the Computational Singular Perturbation (CSP) method. In the CSP context, an adaptive model reduction framework - named the G-Scheme - was created to achieve multi-scale adaptive model reduction along-with the integration of the Differential Equations (DEs), describing a particular problem. If the gaps between the fast/active and slow/active time scales are particularly large, this numerical technique allows to obtain a model reduction with the consequent possibility to integrate only a set of non-stiff DEs, typically much smaller than the dimension of the original problem. The G-Scheme and WAMR are used and coupled in this work with the aim to propose an innovative idea to solve spatially and temporally stiff systems with the minimal number of DoFs. Therefore, the coupled scheme is particularly useful to solve problems having a strong multiscale characteristics in space and time. Improvements of

the performance of the coupled scheme are proposed and implemented. The techniques are tested by studying reaction-diffusion systems at increasing level of complexity.

1.1 Literature review and work organization

The tremendous progress in computer science and programming allows to study complex problems characterized by phenomena having wide ranges of spatial and temporal scales. When a system of partial differential equations (PDEs) is employed to model such phenomena, high resolution discretization is required for solving the finest spatial scales, such as steep gradients, singularities or near singularities. Accurate numerical simulation of compressible reactive flows on a fixed computational grid is particularly challenging in terms of computational resources. Several techniques providing different degrees of accuracy and simplification have been developed to overcome this limitation. To reduce the number of DoFs, while maintaining solution accuracy, adaptive discretization becomes necessary. A possibility is represented by the use of Adaptive Mesh Refinement (AMR) techniques, that are able to provide advantages in terms of computational speed and storage over uniform grid algorithms, but they lack of an a-priori error control. Conversely, the wavelet-based adaptive methods are able to provide a direct measure of the local error at each grid point, allowing to control spatial grid adaption. The adaptive grid is built looking at the values of the wavelet amplitudes: if they are lower than an a-priori user-prescribed error the grid point is simply removed, otherwise it becomes a point requested for the wavelet representation. The WAMR code [36, 35] is the framework used in this work to create adaptive grids. It is able to generate dynamically adaptive grids able to capture any desired accuracy, producing automatically verified solutions. This method has been implemented by Grenga [16] for massive parallel application on high performance computing and it was also verified [35, 4] for a wide range of test cases, in particular compressible and incompressible flows described by reacting Navier-Stokes equations in primitive variables in 1-, 2- and 3-D geometries.

The purpose of the CSP method is to reformulate an original problem as ROM that describes the slow systems dynamics. The CSP method was first developed by Lam and Goussis [25, 20, 21, 22, 24, 23], and extended with new developments and applications by Goussis, Valorani and Paolucci [47, 48, 50, 15, 53, 51, 54], as well as other investigators [29, 30, 56, 41, 27, 39]. The production of a ROM relies on the availability of a suitable decomposition of the dynamics in slow and fast subspaces. In this context, the G-Scheme framework (developed by Valorani [49]) is considered: it represents an adaptive model reduction framework able to decompose the system dynamics into active, slow, fast and invariant subspaces. The G-Scheme introduces locally a curvilinear frame of reference, defined by a set of orthonormal basis vectors with corresponding coordinates, attached to this decomposition [49]. The evolution of the curvilinear coordinates associated with the active subspace is described by non-stiff DEs, whereas that associated with the slow and fast subspaces is accounted for by applying algebraic corrections [49].

The coupling between WAMR and the G-Scheme is proposed to obtain adaptive model reduction in space and time. The coupled scheme was firstly proposed by Grenga [16] using

pWAMR, namely the WAMR version able to be used on distributed memory parallel computers using the Message Passing Interface (MPI) standard [34].

The thesis is organized as follows. Chapter (2) is entirely dedicated to the description of the basic wavelet theory. Focus is given on the construction of a particular class of wavelet functions and on the way to obtain adaptive grids. The essential features of the G-Scheme framework are illustrated in chapter (3), while chapter (4) is dedicated to the coupling between WAMR and the G-Scheme with the aim to achieve adaptive model reduction in space and time. Operator Splitting (OS) techniques are also introduced to use the G-Scheme in a more efficient way. The coupling between WAMR and the G-Scheme is tested in chapter (5), where a simple reaction-diffusion system is studied: the excellent capability of the coupled scheme to produce accurate solutions with respect to methods where uniform grids and/or standard time integrators (DVODE) are used is shown. A more complex reaction-diffusion system, represented by the unsteady flamelets, is studied in chapter (6). The coupled scheme WAMR/G-Scheme is tested with an OS technique. In chapter (7) the unsteady flamelets are studied in supercritical conditions and parametric analysis with respect the some WAMR parameters are proposed. Finally, suggestions for future work are described. They concern the generation of multi-dimensional tabulated flamelets with WAMR, using the preliminary results obtained in chapter (7).

CHAPTER 2

Wavelet Basic Theory

This chapter is intended to explain the basic concepts of the wavelet theory. The wavelets are multiscale basis functions whose properties can be compared to those ones characterizing the basis functions used in other approaches: the main differences between the wavelet based method, the spectral methods and the finite difference, volume and element methods are briefly discussed in section (2.1). An overview on the wavelet world with the description of some important properties of the wavelet functions is given in section (2.2). The concept and goal of the Multi-Resolution Analysis (MRA) is introduced in section (2.3), where formal definitions of wavelet and scaling functions are also provided together with the role they play in the MRA. The construction procedure of scaling and wavelet functions is described in detail in section (2.4) for a particular type of basis functions, namely the Daubechies wavelets. Another important class of scaling and wavelet functions is represented by the interpolating scaling and wavelet functions. They are widely used in the current work for the inexpensive workload associated with their construction, demonstrated in section (2.5). The fast interpolating wavelet transform is described in section (2.6), while the concept of Sparse Wavelet Representation is shown in section (2.7). Section (2.8) is dedicated to the description of the adaptive algorithm to solve time dependent PDEs. Finally, in section (2.9), a brief overview on the parallel implementation of WAMR is given (taken by Grenga [16]).

Most of the work is taken from the works of Wirasaet [55], Rastigejev [40] and Grenga [16], where complete details concerning the theory and the adaptive algorithm here cited can be found. The author's contribution is provided in the detailed explanations (with examples) of the some important aspects of the wavelet theory. Further details about the author's contribution are explicitly indicated in each section of the chapter.

2.1 Motivation for wavelets

Wavelets are multiscale basis functions that are localized in both physical and spectral spaces. These good localization properties can be contrasted with other approaches

- Spectral methods: they use infinitely differentiable basis functions that are localized in spectral space providing exponential convergence rates for smooth functions. The global

support in physical space leads to full matrices and difficulty in resolving large changes in physical space.

- Finite-difference, finite-volume and finite-element methods: the basis functions have small compact support in physical space, leading to sparse matrices and the ability of resolving large changes in physical space, but the global support in spectral space leads to low-order (algebraic) convergence rates.

Wavelets appear to combine the advantages of both spectral and finite-difference/volume/element basis. One can expect that numerical methods based on wavelets will attain both good spatial and spectral resolution. In addition, the wavelets provide effective compression and local analysis of data.

2.2 Overview on wavelet families and most important properties

This section is intended to show a possible strategy to classify the wavelet functions in different families. Further details can be found in [14].

A large variety of wavelet families is available in the mathematical literature. The choice of a particular wavelet family depends on the intended application. Typically, two important properties must be taken into account to choose a specific wavelet family: the length of the wavelet support L_w in the space and wavenumber domains and the number of vanishing moments N_w.

1. Wavelets having compact support in space and/or wavenumber domain are nonzero only in a finite space and/or wavenumber interval. The support is defined quasi-compact in space and/or wavenumber domain if $L_w = \infty$ and the wavelet function decays sufficiently fast in the space and/or wavenumber domain. Wavelets having compact or quasi-compact support in both space and wavenumber domains are good localized.

2. Each orthogonal wavelet family is characterized by two functions, namely the mother scaling function ϕ and the mother wavelet function ψ. The number of vanishing moments of a wavelet is N_w if and only if the wavelet function is orthogonal to the polynomials up to degree $N_w - 1$ (Eq. (2.1)): the scaling function alone can generate polynomials up to degree $N_w - 1$. More vanishing moments means that the scaling function can represent more complex functions.

Figure (2.1a) shows the Mexican Hat wavelet: it is an example of wavelet function having a quasi-compact support in the space and wavenumber domains, characterized by $N_w = 2$ vanishing moments. It is able to encode polynomials with two coefficients, *i.e.* constant and linear signal components. Figure (2.1b) represents the Shannon wavelet for which the number of vanishing moments is $N_w = \infty$: polynomials with an infinite number of coefficients can be encoded.

$$\int_{-\infty}^{\infty} x^k \psi(x) dx = 0, \qquad 0 \leq k \leq N_w - 1 \tag{2.1}$$

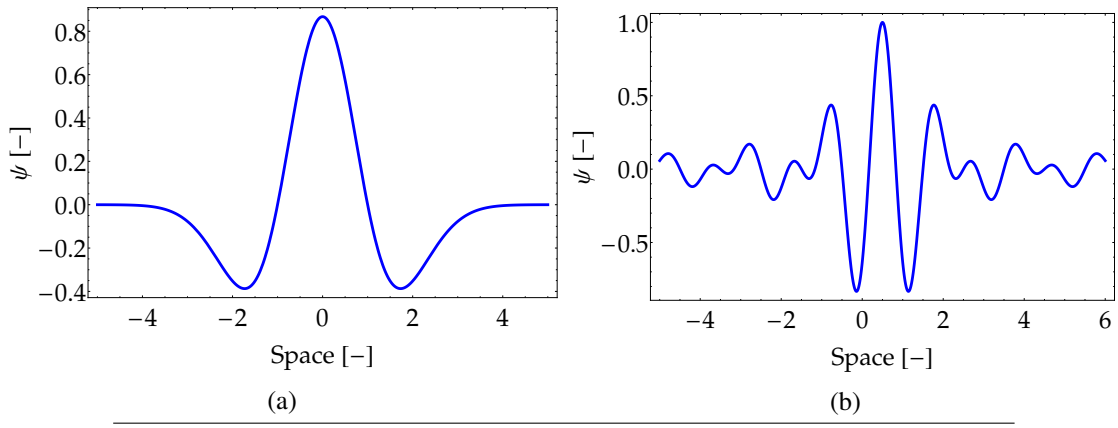


FIGURE 2.1: Example of wavelet functions: (a) Mexican Hat wavelet: the support is quasi-compact in the space and wavenumber domain and the number of vanishing moments is $N_w = 2$. (b) Shannon wavelet: the length of the wavelet support is $L_w = \infty$ in the space domain, while the support is compact in the wavenumber domain. The number of vanishing moments is $N_w = \infty$.

2.3 The concept of Multi-Resolution Analysis

The theory described in this section summarizes part of work of Wirasaet [55] and Rastigejev [40].

The Multi-Resolution Analysis (MRA) or multiscale approximation (MSA) represents a design method of discrete wavelet transform (DWT). The MRA of the Lebesgue space $L_2(\mathbb{R})$ consists in a sequence of levels of details/resolution or nested subspaces $\{V_l\}_{l\in\mathbf{Z}} \in L_2(\mathbb{R})$ such that:

$$\{0\} \subset \ldots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots \subset L_2(\mathbb{R}), \tag{2.2}$$

$$\overline{\bigcup_{l=-\infty}^{+\infty} V_l} = L_2(\mathbb{R}). \tag{2.3}$$

The MRA goal is to expand a function $f(x) \in L_2(\mathbb{R})$ at various levels of detail, satisfying the following properties:

$$f(x) \in V_l \iff f(2x) \in V_{l+1}, \tag{2.4}$$

$$f(x) \in V_l \iff f(x+1) \in V_l. \tag{2.5}$$

The basis functions of the sequence of subspaces $\{V_l\}$ are scaling functions on scale of resolution l and location k, *i.e.* $\{\phi_{l,k}, k \in \mathbb{Z}\}$. They can be written as the dilation and translation of ϕ, Eq. (2.6), where ϕ represents the solution of the dilation equation, (Eq. 2.7)

$$\phi_{l,k}(x) = \phi(2^l x - k), \tag{2.6}$$

$$\phi(x) = \sum_{k=0}^{K-1} h_k \phi(2x - k). \tag{2.7}$$

The K coefficients h_k determines the scaling function of the particular wavelet systems [7, 32, 9]. Note that a closed form of ϕ is unavailable, but it is not necessary since only the coefficients h_k are needed in a practical implementation.

A sequence of subspaces $\{W_l\}_{l\in\mathbf{Z}} \in L_2(\mathbb{R})$ is now considered. They are defined as the complement of V_l in V_{l+1}, Eq. (2.8) and Fig. (2.2),

$$V_{l+1} = V_l \oplus W_l. \tag{2.8}$$

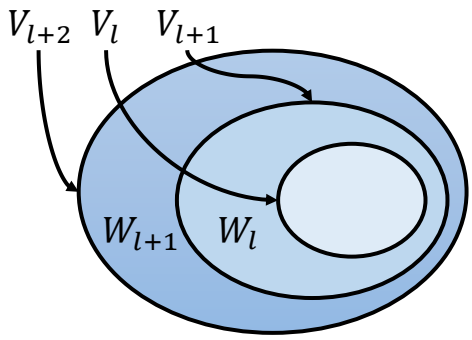


FIGURE 2.2: Relation between the subspaces W_l and V_l, where l is the resolution level.

The basis functions of $\{W_l\}$ are called wavelets on scale of resolution l and location k, *i.e.* $\{\psi_{l,k}, k \in \mathbb{Z}\}$. They can be written as the dilation and translation of ψ, Eq. (2.9), where ψ is the solution of the relation between $\psi(x)$ and $\phi(x)$, Eq. (2.10):

$$\psi_{l,k}(x) = \psi(2^l x - k), \tag{2.9}$$

$$\psi(x) = \sum_{k=0}^{K-1} g_k \phi(2x - k). \tag{2.10}$$

The K filter coefficients g_k determines the particular wavelet system. They are related to the coefficients h_k:

$$g_k = (-1)^k h_{K-k-1}, \qquad k = 0, \ldots, K-1 \tag{2.11}$$

The decomposition of V_l until V_{l_0}, where V_{l_0} denotes the subspace associated with the coarsest scale, yields the MRA of V_l,

$$V_l = V_{l_0} \oplus \left(\bigoplus_{j=l_0}^{l-1} W_j \right). \tag{2.12}$$

Equation (2.12) implies that a function $f^l \in V_l$ can be written either in the single scale representation

$$f^l(x) = \sum_k c_{l,k}\phi_{l,k}(x), \tag{2.13}$$

or, equivalently, in the multiscale scale representation

$$f^l(x) = \sum_k c_{l_0,k}\phi_{l_0,k}(x) + \sum_{j=l_0}^{l-1}\sum_k d_{j,k}\psi_{j,k}(x), \tag{2.14}$$

where $\{c_{l,k}\}$ and $\{d_{l,k}\}$ are denoted as scaling function and wavelet coefficients, respectively. Note that since the collection of $\{\phi_{l_0,k}\}$ and $\{\psi_{j,k}\}_{j=l_0}^{l-1}$ forms a basis of V_l, the coefficients $\{c_{l_0,k}\}$ and $\{d_{j,k}\}_{j=l_0}^{l-1}$ are determined uniquely.

Let $f^l \in V_l$ be an approximation of a function f, and consider the difference of the approximation of f in two successive spaces V_l and V_{l+1}:

$$w^l = f^{l+1} - f^l = \sum_k c_{l+1,k}\phi_{l+1,k} - \sum_k c_{l,k}\phi_{l,k} = \sum_k d_{l,k}\psi_{l,k}. \tag{2.15}$$

Properties 1 and 2 imply that the wavelet coefficient $d_{l,k}$ is large when the function $\psi_{l,k}$ lies in the vicinity of a singularity or, near singularity, of f. Therefore, the number of wavelets required to represent the function f having near singular behaviour, within a prescribed accuracy, can be substantially reduced if one uses the multiscale representation instead of the single scale representation.

2.4 Basis functions construction

The procedure to solve Eq. (2.7) and Eq. (2.10) is now described and more details can be found in *e.g.* [7, 32, 6, 17]. Focus on the Daubechies wavelets is given: they can be used to build the interpolating scaling and wavelet functions, as will be explained in section (2.5).

2.4.1 The filter coefficients

The $\{h_k\}_{k=0,\ldots,K-1}$ filter coefficients of Eq. (2.7) are evaluated considering the following properties:

- the scaling function should be normalised to unity,

$$\int_{-\infty}^{\infty}\phi(x)dx = 1, \tag{2.16}$$

- the translates of the scaling function $\phi(x)$ should satisfy the orthonormality property

$$\int_{-\infty}^{\infty}\phi(x)\phi(x-n)dx = \delta_{0,n} \qquad n \in \mathbb{Z}. \tag{2.17}$$

Using Eq. (2.7) in Eq. (2.16) and Eq. (2.17) one obtains

$$\sum_{k=0}^{K-1} h_k = 2, \tag{2.18}$$

$$\sum_{k=k_0}^{k_1} h_k h_{k-2n} = 2\delta_{0,n} \qquad k = 0, 1, ..., K/2-1, \tag{2.19}$$

where k_0 =max$(0, 2n)$ and k_1 =min$(K-1, K-1+2n)$. Using Eq. (2.1) and Eqs. (2.10)-(2.11), Eq. (2.20) is obtained

$$\sum_{i=0}^{K-1} (-1)^i h_i i^k = 0 \qquad k = 0, 1, ..., K/2-1. \tag{2.20}$$

To find the K filter coefficients, the $K+1$ Eqs. (2.18)-(2.19)-(2.20) need to be solved. It can be shown that the real number of equations is K, given that for $k = 0$ Eq. (2.20) is redundant and can be omitted. Solving Eqs. (2.18)-(2.19)-(2.20) for $K = 4$ the filter coefficients of the scaling function are $h_0 = \frac{1}{4}(1+\sqrt{3})$, $h_1 = \frac{1}{4}(3+\sqrt{3})$, $h_2 = \frac{1}{4}(3-\sqrt{3})$ and $h_3 = \frac{1}{4}(1-\sqrt{3})$ while, from Eq. (2.11), the filter coefficients of the wavelet function are found, $g_0 = \frac{1}{4}(1-\sqrt{3})$, $g_1 = -\frac{1}{4}(3-\sqrt{3})$, $g_2 = \frac{1}{4}(3+\sqrt{3})$ and $g_3 = -\frac{1}{4}(1+\sqrt{3})$.

2.4.2 Scaling and wavelet function construction

Once the set of filter coefficients $\{h_k\}_{k=1,...,K-1}$ is known, the scaling function is easily identified by solving Eq. (2.7).

Considering again the case of $K = 4$, Eq. (2.7) becomes

$$\phi(x) = \sum_{k=0}^{3} h_k \phi(2x-k) = h_0\phi(2x) + h_1\phi(2x-1) + h_2\phi(2x-2) + h_3\phi(2x-3) \tag{2.21}$$

Equation (2.16) and Eq. (2.21) are used to evaluate the scaling function values at the 4 integer points $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ and $x_3 = 3$. In particular, from Eq. (2.21) one has:

$$\begin{aligned} \phi(x_0) &= h_0\phi(2x_0) + h_1\phi(2x_0-1) + h_2\phi(2x_0-2) + h_3\phi(2x_0-3), \\ \phi(0) &= h_0\phi(0) + h_1\phi(-1) + h_2\phi(-2) + h_3\phi(-3), \end{aligned} \tag{2.22}$$

$$\begin{aligned} \phi(x_1) &= h_0\phi(2x_1) + h_1\phi(2x_1-1) + h_2\phi(2x_1-2) + h_3\phi(2x_1-3), \\ \phi(1) &= h_0\phi(2) + h_1\phi(1) + h_2\phi(0) + h_3\phi(-1), \end{aligned} \tag{2.23}$$

$$\begin{aligned} \phi(x_2) &= h_0\phi(2x_2) + h_1\phi(2x_2 - 1) + h_2\phi(2x_2 - 2) + h_3\phi(2x_2 - 3), \\ \phi(2) &= h_0\phi(4) + h_1\phi(3) + h_2\phi(2) + h_3\phi(1), \end{aligned} \tag{2.24}$$

$$\begin{aligned} \phi(x_3) &= h_0\phi(2x_3) + h_1\phi(2x_3 - 1) + h_2\phi(2x_3 - 2) + h_3\phi(2x_3 - 3), \\ \phi(3) &= h_0\phi(6) + h_1\phi(5) + h_2\phi(4) + h_3\phi(3), \end{aligned} \tag{2.25}$$

while from Eq. (2.16) one has:

$$\begin{aligned} \phi(x_0) + \phi(x_1) + \phi(x_2) + \phi(x_3) &= 1, \\ \phi(0) + \phi(1) + \phi(2) + \phi(3) &= 1. \end{aligned} \tag{2.26}$$

Property 1 implies that the scaling function values outside the bounded set $x \in [0, 3]$ are zero, namely $\phi(-1) = \phi(-2) = \phi(-3) = \phi(4) = \phi(5) = \phi(6) = 0$. Therefore, the solution of Eqs. (2.22)-(2.26) is

$$\phi(0) = 0 \quad \phi(1) = 1.366 \quad \phi(2) = -0.366 \quad \phi(3) = 0 \tag{2.27}$$

The scaling function values at the 4 integer points are represented in Fig. (2.3).

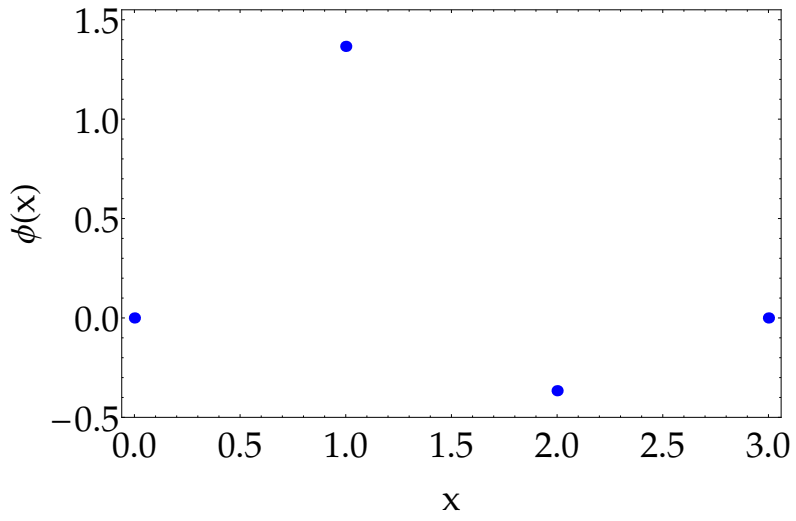


FIGURE 2.3: Daubechies 4 - Scaling function values built at the 4 integer points $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, $x_3 = 3$.

Once $\phi(x)$ is known at 4 integer points $x_0 = 0$, $x_1 = 1$, $x_2 = 2$ and $x_3 = 3$, the function values at the dyadic points are evaluated using again Eq. (2.7): Figure (2.4) shows the results of four consecutive evaluations of the function values at the dyadic points. Figure (2.5a) shows the scaling

function built on a grid composed by 3073 dyadic points. If the procedure is indefinitely repeated, a continuous function is obtained. The corresponding wavelet function is obtained through Eqs. (2.10)-(2.11), and it is represented in Fig. (2.5b) on a grid composed by 3073 dyadic points.

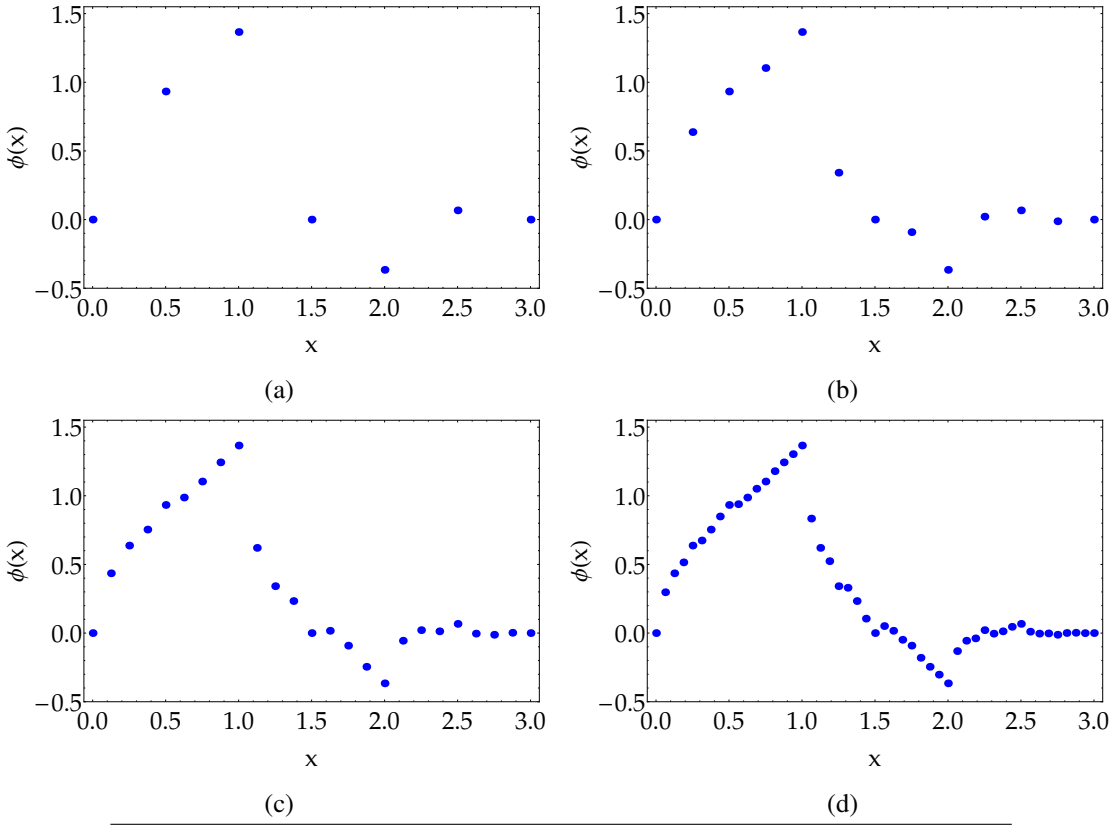


FIGURE 2.4: Daubechies 4 - Scaling function values built at 7 (a), 13 (b), 25 (c) and 49 grid points (d).

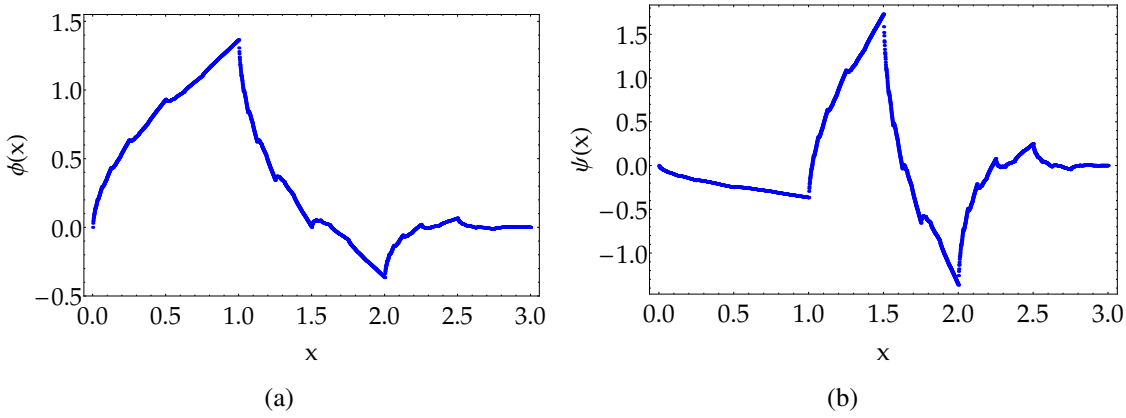


FIGURE 2.5: Daubechies 4 - Scaling function (a) and wavelet function (b) built at 3073 grid points.

The whole procedure to build the scaling and wavelet functions is summarized in Fig. (2.6): once the filter coefficients defining the scaling function are known, the scaling and wavelet

functions are uniquely identified.

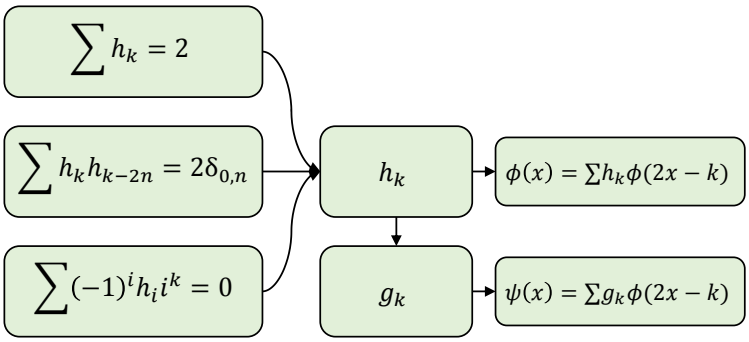


FIGURE 2.6: Solving the system of algebraic equations (2.18)-(2.19)-(2.20), the filter coefficients h_k and g_k are found. They are used to build the scaling function $\phi(x)$ and wavelet function $\psi(x)$.

2.5 One-dimensional interpolating scaling and wavelet functions

The interpolating scaling and wavelet functions represent elementary basis functions. Different strategies are at the base of their construction: they can be found considering the translation and dilation of the auto-correlation of the Daubechies scaling functions [7]. Another approach is the recursive interpolation: starting from the knowledge of functional values on a base grid, the functional values at the midpoints are found through the polynomial interpolation. The procedure can be repeated until a continuous function is obtained. Section (2.5.1) and the first part of section (2.5.2) are dedicated to a rigorous description of the interpolation subdivision scheme, entirely taken by the work of Wirasaet [55]. The example application of the interpolation subdivision scheme in the second part of section (2.5.2) is inserted by the author. The interpolating scaling function properties are described in section (2.5.3) (Wirasaet [55]), while the interpolating wavelet functions are introduced in section (2.5.4) Wirasaet [55].

2.5.1 Interpolation subdivision scheme

The so-called interpolating scaling function that yields the multiresolution analysis on the interval arises naturally from the interpolation subdivision scheme introduced by [8] for data defined for x on an interval. Let $\bar{p}$ be an even positive integer, l an integer such that $l \geq \log_2(\bar{p})$, and

$$V_l = \{x_{l,k} = k2^{-l} \ : \ k \in k_l^0\}, \quad k_l^0 = \{0, 1, \ldots, 2^l - 1, 2^l\} \tag{2.28}$$

be a grid of dyadic points at scale level l. In addition, let $\{f_{l,k} \equiv f(x_{l,k}) \ : \ x_{l,k} \in V_l, \ k = 0, \cdots, 2^l\}$ be a set of function values at the dyadic points of a given function $f(x)$ defined on $x \in [0, 1]$. The interpolation subdivision scheme is a process to interpolate from a given data $\{f_{l,k}\}_{k \in k_l^0}$ to

obtain a function $\widetilde{f}(x)$ defined on $x \in [0,1]$, such that $\widetilde{f}(x_{l,k}) = f_{l,k}$. The scheme begins with determining the interpolated values on the grid V_{l+1}. This is accomplished by computing the interpolated values at odd grid points $x_{l+1,2k+1}$ by means of polynomial interpolation. For the even grid points, since $x_{l+1,2k} = x_{l,k}$, the function values remain unchanged, *i.e.* $\widetilde{f}_{l+1,2k} = f_{l,k}$. More specifically, the determination of $\widetilde{f}_{l+1,2k+1}$ is computed by fitting a polynomial $\pi_{l,k}$ of degree $\bar{p}-1$ to the function values at points in $X_{l,k}$:

$$X_{l,k} = \{x_{l,m} \,:\, m \in \mathcal{X}_{l,k}\}, \tag{2.29}$$

where the index set $\mathcal{X}_{l,k}$ is defined by

$$\mathcal{X}_{l,k} = \begin{cases} \{0,\ldots,\bar{p}-1\} & \text{for} \quad 0 \le k \le \bar{p}/2-2, \\ \{k-\bar{p}/2+1,\ldots,k+\bar{p}/2\} & \text{for} \quad \bar{p}/2-1 \le k \le 2^l-\bar{p}/2, \\ \{2^l-\bar{p}+1,\ldots,2^l\} & \text{for} \quad 2^l-\bar{p}/2+1 \le k \le 2^l-1, \end{cases} \tag{2.30}$$

and the polynomial $\pi_{l,k}$ is given by

$$\pi_{l,k}(x) = \sum_{m=\min \mathcal{X}_{l,k}}^{\max \mathcal{X}_{l,k}} f_{l,m} L_{l,m}^k(x), \tag{2.31}$$

where $L_{l,m}^k$ is the Lagrange polynomial defined by

$$L_{l,m}^k \equiv \prod_{r=\min \mathcal{X}_{l,k}, r\neq m}^{\max \mathcal{X}_{l,k}} \frac{(x-x_{l,r})}{(x_{l,m}-x_{l,r})}, \tag{2.32}$$

so that

$$\widetilde{f}(x_{l+1,2k+1}) \equiv \pi_{l,k}(x_{l+1,2k+1}). \tag{2.33}$$

Since the function values $\{\widetilde{f}_{j+1,k}\}_{k\in k_{l+1}^0}$ are now known, the interpolated values on V_{l+2} can be calculated using the same scheme with $\{\widetilde{f}_{j+1,k}\}_{k\in k_{l+1}^0}$ for the starting data. By applying this scheme iteratively, one obtains the interpolation function $\widetilde{f}(x)$ for $x \in [0,1]$ in the limit $l \to \infty$. For $\bar{p} = 2$, the subdivision scheme yields linear interpolation. For higher $\bar{p}$, the interpolation subdivision scheme defines a function whose continuity increases with $\bar{p}$ [9], *i.e.* with the order of the polynomial used in the interpolation process.

2.5.2 One-dimensional interpolating scaling functions

Let $\phi_{l,k}(x)$ for $k \in \{0,\cdots,2^l\}$ denote a function resulting from applying the subdivision scheme of order $\bar{p}$ to the Kronecker data $\{\delta_{r,k} \,:\, r \in k_l^0\}$ (a set of function values whose entries are zeros except at the point $x_{l,k}$). As discussed above, the function $\phi_{l,k}$ is continuous and its continuity increases as a function of $\bar{p}$.

It can be noticed that the subdivision scheme is linear. Consequently, the interpolation function $\widetilde{f}(x)$ resulting from applying the scheme to $\{f_{l,k}\}_{k\in k_l^0}$ is equivalent to a superposition of those

interpolation functions of the data sets $\{f_{l,r}\delta_{r,k} \ : \ r = k_l^0\}$, $k = 0, \ldots, 2^l$. More precisely, the interpolation function $\widetilde{f}$ is a linear combination of $\phi_{l,k}(x)$,

$$\widetilde{f}(x) = \sum_{k=0}^{2^l} f_{l,k}\phi_{l,k}(x) \tag{2.34}$$

The function $\phi_{l,k}(x)$ is known as the interpolating scaling function [3, 9] (they are linearly independent and hence form a basis).

The interpolating procedure is applied to the Kronecker data. First, the integer $\bar{p}$ is set equal to 4: the scale level has to be larger than $l \geq \log_2(\bar{p}) = \log_2(4) = 2$. Setting $l = 3$, it follows from Eq. (2.28):

$$\begin{aligned} V_3 &= \{x_{3,k} = k2^{-3} \ : \ k \in k_3^0\}, \quad k_3^0 = \{0, 1, \ldots, 2^3 - 1, 2^3\} = \{0, 1, \ldots, 8\}, \\ V_3 &= \{x_{3,k} = k/8 \ : \ k \in k_3^0\} = \{x_{3,0}, x_{3,1}, \ldots, x_{3,8}\} = \{0, 1/8, \ldots, 1\}. \end{aligned} \tag{2.35}$$

This means that the grid at the scale level $l = 3$ is composed of 9 points. The Kronecker delta function values at the 9 dyadic points are $\{f_{3,k} \equiv f(x_{3,k}) \ : \ x_{3,k} \in V_3, \ k = 0, \cdots, 8\} = \{f_{3,0}, f_{3,1}, \ldots, f_{3,8}\} = \{1, 0, \ldots, 0\}$, as Fig. (2.7a) shows, where also the grid points index set associated with the third resolution level is represented. The first step is to find the interpolated values at the new grid points associated with the scale level $l + 1 = 4$, showed in Fig. (2.7b) with the black triangular symbols.

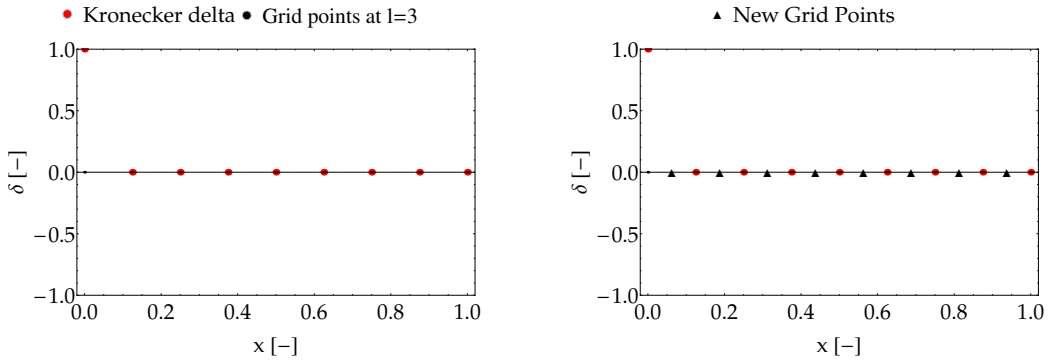


FIGURE 2.7: Kronecker delta values at the dyadic points for the scale level $l = 3$ (a) and $l = 4$ (b).

The procedure described in section (2.5.1) is applied. From the definition of the index set (Eq. (2.30)) the stencils to build the local interpolating polynomials are found:

$$\mathcal{X}_{3,k} = \begin{cases} \{0,1,2,3\} & \text{for} \quad k=0 \rightarrow \{x_{3,0}, x_{3,1}, x_{3,2}, x_{3,3}\} \\ \{0,1,2,3\} & \text{for} \quad k=1 \rightarrow \{x_{3,0}, x_{3,1}, x_{3,2}, x_{3,3}\} \\ \{1,2,3,4\} & \text{for} \quad k=2 \rightarrow \{x_{3,1}, x_{3,2}, x_{3,3}, x_{3,4}\} \\ \{2,3,4,5\} & \text{for} \quad k=3 \rightarrow \{x_{3,2}, x_{3,3}, x_{3,4}, x_{3,5}\} \\ \{3,4,5,6\} & \text{for} \quad k=4 \rightarrow \{x_{3,3}, x_{3,4}, x_{3,5}, x_{3,6}\} \\ \{4,5,6,7\} & \text{for} \quad k=5 \rightarrow \{x_{3,4}, x_{3,5}, x_{3,6}, x_{3,7}\} \\ \{5,6,7,8\} & \text{for} \quad k=6 \rightarrow \{x_{3,5}, x_{3,6}, x_{3,7}, x_{3,8}\} \\ \{5,6,7,8\} & \text{for} \quad k=7 \rightarrow \{x_{3,5}, x_{3,6}, x_{3,7}, x_{3,8}\}. \end{cases} \tag{2.36}$$

Therefore, Eqs. (2.31)-(2.32) allow to build the local interpolating polynomials (Eq. (2.31)):

$$\pi_{3,0}(x) = \sum_{m=\min \mathcal{X}_{3,0}}^{\max \mathcal{X}_{3,0}} f_{3,m} L^0_{3,m}(x) =$$

$$= \sum_{m=0}^{3} f_{3,m} L^0_{3,m}(x) = f_{3,0} L^0_{3,0}(x) + f_{3,1} L^0_{3,1}(x) + f_{3,2} L^0_{3,2}(x) + f_{3,3} L^0_{3,3}(x), \tag{2.37}$$

$$L^0_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,0}, r\neq m}^{\max \mathcal{X}_{3,0}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})} = \prod_{r=0, r\neq m}^{3} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})}, \quad m = 0, \ldots, 3$$

$$\pi_{3,1}(x) = \sum_{m=\min \mathcal{X}_{3,1}}^{\max \mathcal{X}_{3,1}} f_{3,m} L^1_{3,m}(x) =$$

$$= \sum_{m=0}^{3} f_{3,m} L^1_{3,m}(x) = f_{3,0} L^1_{3,0}(x) + f_{3,1} L^1_{3,1}(x) + f_{3,2} L^1_{3,2}(x) + f_{3,3} L^1_{3,3}(x), \tag{2.38}$$

$$L^1_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,1}, r\neq m}^{\max \mathcal{X}_{3,1}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})} = \prod_{r=0, r\neq m}^{3} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})}, \quad m = 0, \ldots, 3$$

$$\pi_{3,2}(x) = \sum_{m=\min \mathcal{X}_{3,2}}^{\max \mathcal{X}_{3,2}} f_{3,m} L^2_{3,m}(x) =$$

$$= \sum_{m=1}^{4} f_{3,m} L^2_{3,m}(x) = f_{3,1} L^2_{3,1}(x) + f_{3,2} L^2_{3,2}(x) + f_{3,3} L^2_{3,3}(x) + f_{3,4} L^2_{3,4}(x), \tag{2.39}$$

$$L^2_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,2}, r\neq m}^{\max \mathcal{X}_{3,2}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})} = \prod_{r=1, r\neq m}^{4} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})}, \quad m = 1, \ldots, 4$$

$$\pi_{3,3}(x) = \sum_{m=\min \mathcal{X}_{3,3}}^{\max \mathcal{X}_{3,3}} f_{3,m} L^3_{3,m}(x) =$$

$$= \sum_{m=2}^{5} f_{3,m} L^3_{3,m}(x) = f_{3,2} L^3_{3,2}(x) + f_{3,3} L^3_{3,3}(x) + f_{3,4} L^3_{3,4}(x) + f_{3,5} L^3_{3,5}(x), \tag{2.40}$$

$$L^3_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,3}, r\neq m}^{\max \mathcal{X}_{3,3}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}} = \prod_{r=2, r\neq m}^{5} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}}, \quad m = 2, \ldots, 5$$

$$\pi_{3,4}(x) = \sum_{m=\min \mathcal{X}_{3,4}}^{\max \mathcal{X}_{3,4}} f_{3,m} L^4_{3,m}(x) =$$

$$= \sum_{m=3}^{6} f_{3,m} L^4_{3,m}(x) = f_{3,3} L^4_{3,3}(x) + f_{3,4} L^4_{3,4}(x) + f_{3,5} L^4_{3,5}(x) + f_{3,6} L^4_{3,6}(x), \tag{2.41}$$

$$L^4_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,4}, r\neq m}^{\max \mathcal{X}_{3,4}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}} = \prod_{r=3, r\neq m}^{6} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}}, \quad m = 3, \ldots, 6$$

$$\pi_{3,5}(x) = \sum_{m=\min \mathcal{X}_{3,5}}^{\max \mathcal{X}_{3,5}} f_{3,m} L^5_{3,m}(x) =$$

$$= \sum_{m=4}^{7} f_{3,m} L^5_{3,m}(x) = f_{3,4} L^5_{3,4}(x) + f_{3,5} L^5_{3,5}(x) + f_{3,6} L^5_{3,6}(x) + f_{3,7} L^5_{3,7}(x), \tag{2.42}$$

$$L^5_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,5}, r\neq m}^{\max \mathcal{X}_{3,5}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}} = \prod_{r=4, r\neq m}^{7} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}}, \quad m = 4, \ldots, 7$$

$$\pi_{3,6}(x) = \sum_{m=\min \mathcal{X}_{3,6}}^{\max \mathcal{X}_{3,6}} f_{3,m} L^6_{3,m}(x) =$$

$$= \sum_{m=5}^{8} f_{3,m} L^6_{3,m}(x) = f_{3,5} L^6_{3,5}(x) + f_{3,6} L^6_{3,6}(x) + f_{3,7} L^6_{3,7}(x) + f_{3,8} L^6_{3,8}(x), \tag{2.43}$$

$$L^6_{3,m}(x) \equiv \prod_{r=\min \mathcal{X}_{3,6}, r\neq m}^{\max \mathcal{X}_{3,6}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}} = \prod_{r=5, r\neq m}^{8} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r)}}, \quad m = 5, \ldots, 8$$

$$\pi_{3,7}(x) = \sum_{m=\min \chi_{3,7}}^{\max \chi_{3,7}} f_{3,m} L^7_{3,m}(x) =$$

$$= \sum_{m=5}^{8} f_{3,m} L^7_{3,m}(x) = f_{3,5} L^7_{3,5}(x) + f_{3,6} L^7_{3,6}(x) + f_{3,7} L^7_{3,7}(x) + f_{3,8} L^7_{3,8}(x), \tag{2.44}$$

$$L^7_{3,m}(x) \equiv \prod_{r=\min \chi_{3,7}, r\neq m}^{\max \chi_{3,7}} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})} = \prod_{r=5, r\neq m}^{8} \frac{(x - x_{3,r})}{(x_{3,m} - x_{3,r})}, \quad m = 5, \ldots, 8.$$

The interpolated function values at the new grid points introduced with the resolution level $l = 4$ can be evaluated from Eq. (2.33), Fig. (2.8). By repeating the whole procedure in the limit $l \to \infty$, the function $\phi_{3,0}(x)$ is obtained, Fig. (2.9). Figure (2.10) shows the interpolating scaling functions $\phi_{3,k}(x), k = 1, \ldots, 4$ built on a infinite number of resolution levels.

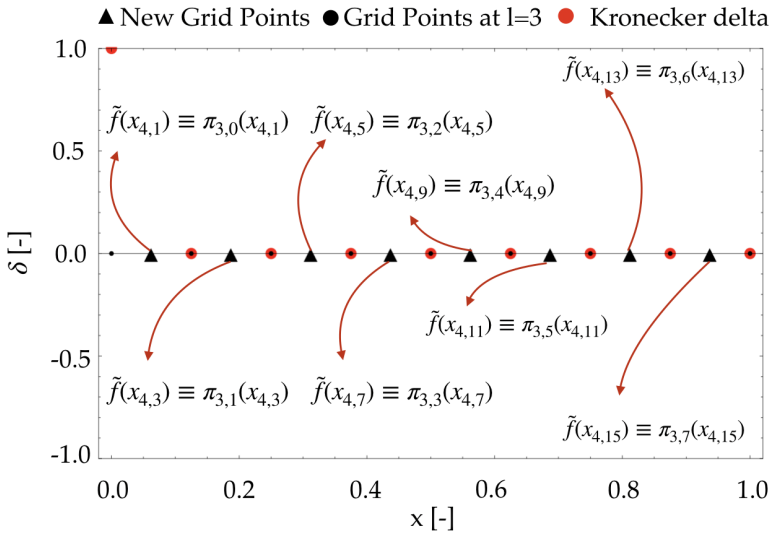


FIGURE 2.8: Interpolated function values at the grid points introduced by the resolution level $l = 4$. The values are evaluated from Eq. (2.33).

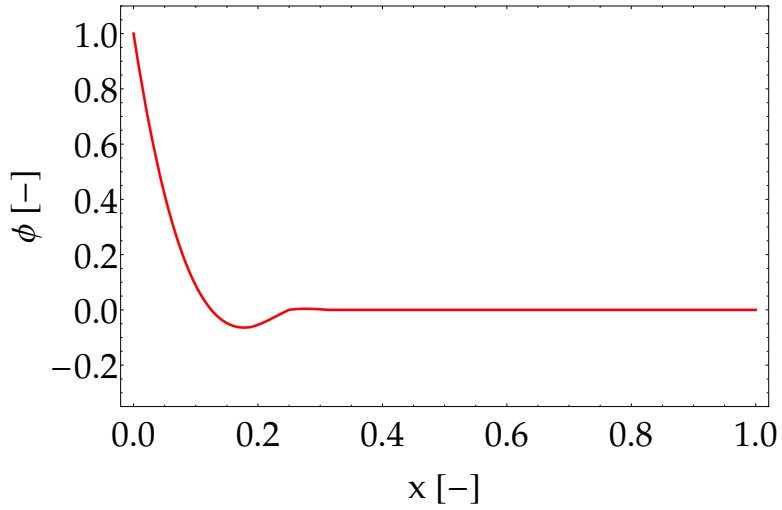


FIGURE 2.9: Interpolating scaling function $\phi_{3,0}(x)$.

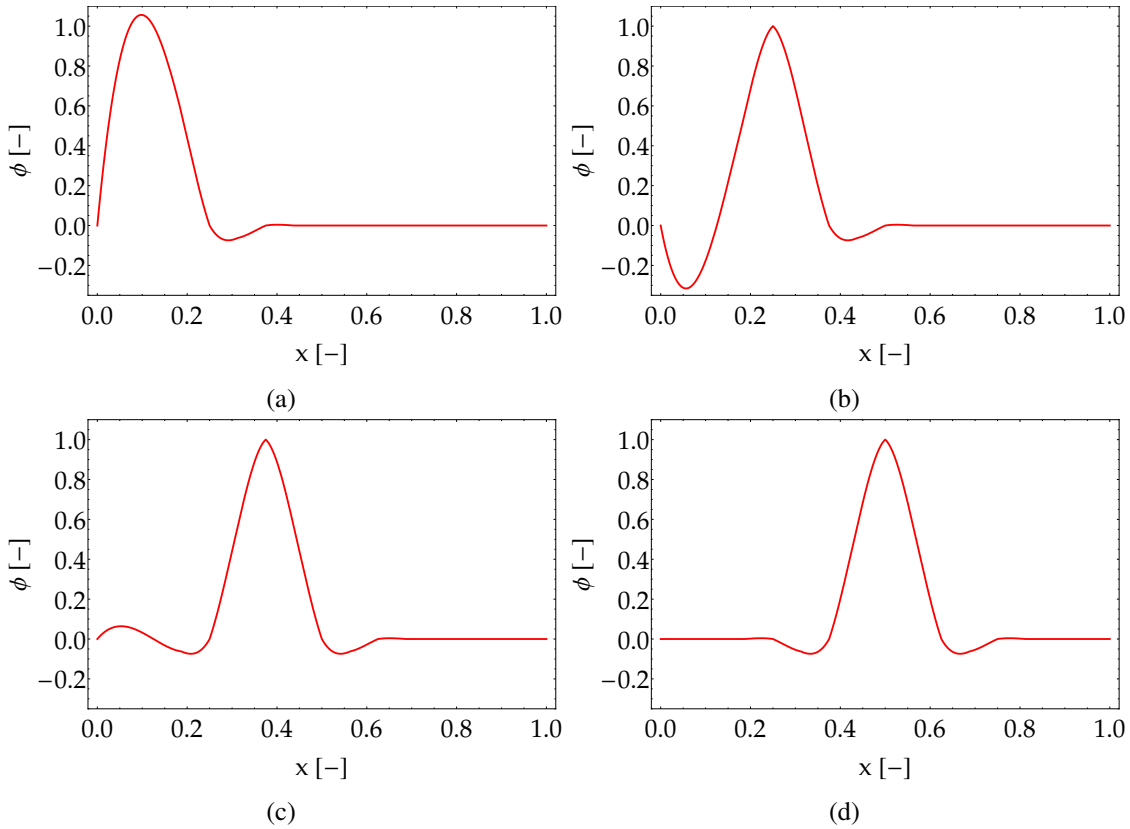


FIGURE 2.10: Interpolating scaling function $\phi_{3,k}(x)$, with $k = 1$ (a), $k = 2$ (b), $k = 3$ (c) and $k = 4$ (d).

2.5.3 Interpolating scaling functions properties

The interpolating scaling function $\phi_{j,k}(x)$ properties are summarized in the following points:

i) $\phi_{l,k}$ satisfied the interpolation property on the dyadic grid V_l, *i.e.* $\phi_{l,k}(x_{l,r}) = \delta_{k,r}$.

ii) The support of $\phi_{j,k}$ is compact: $|\text{supp}\,\phi_{l,k}| = O(2^{-l})$, or more precisely,

$$\text{supp}\,\phi_{l,k} = [\max(0, 2^{-l}(k - \bar{p} + 1)), \min(2^{-l}(k + \bar{p} - 1), 1)] \tag{2.45}$$

iii) The function $\phi_{l,k}$ is the solution of the two-scale relation

$$\phi_{l,k}(x) = \sum_{r=0}^{r=2^{l+1}} h_r^{l,k} \phi_{l+1,r}(x), \tag{2.46}$$

where $h_r^{l,k}$ are the so-called filter coefficients. For $r \in \{\min(0, 2k - \bar{p} + 1), \ldots, \max(2k + \bar{p} - 1, 2^{l+1})\}$, the filter coefficients are given by

$$h_r^{l,k} = \phi_{l,k}(x_{l+1,r}) = \begin{cases} \delta_{r-2k} & \text{for} \quad r \ \text{ even}, \\ L_{l,k}^{(r-1)/2}(x_{l+1,r}) & \text{for} \quad r \ \text{ odd}, \end{cases} \tag{2.47}$$

and all other $h_r^{l,k}$ are zero. Note that $L_{l,k}^m$ is the Lagrange polynomial defined by (2.32).

iv) Polynomials on $[0, 1]$ of degree less than $\bar{p}$ can be written as linear combinations of $\{\phi_{l,k} : k = 0, \ldots, 2^l\}$. More precisely,

$$x^i = \sum_{k=0}^{k=2^l} \left(2^{-l}k\right)^i \phi_{l,k}(x), \quad i = 0, 1, \ldots, \bar{p} - 1. \tag{2.48}$$

It is the property i) that distinguishes the interpolating scaling function from other non-interpolating scaling functions. This property implies that they are linearly independent and thus form a basis. Properties i), iii), and iv) are obvious from the subdivision scheme. Property ii) can be verified using the fact that, for a nonzero data set with finite support on V_l, one iteration of the subdivision scheme adds nonzero layers of size $2^{-j-1}(\bar{p} - 1)$ on both side. Therefore, with infinite iterations, starting with the Kronecker data on V_l with nonzero element at the point $x_{l,k}$, the layer of total width $\sum_{i=1}^{\infty} 2^{-l-i}(\bar{p} - 1) = (\bar{p} - 1)/2^l$ is added to both sides of the point at $2^{-l}k$. Since the subdivision scheme is restricted to the interval $[0, 1]$ (*i.e.* for an interpolated point near a boundary, a set of points participating in the construction of $\pi_{j,k}$ is adjusted accordingly), one observed that (2.45) holds.

2.5.4 One-dimensional interpolating wavelet functions

Let V_l denotes the span of $\{\phi_{l,k}(x) : k \in k_l^0\}$. Here the symbol V_l is overloaded to a space of functions. The specific meaning should be clear from the context. Owing to the interpolation property of the scaling functions, the interpolation operator I_l maps a given function $f(x)$ defined for $x \in [0, 1]$ to the space V_l:

$$(I_l f)(x) \equiv \sum_{k=0}^{k=2^l} f_{l,k}\phi_{l,k}(x) \tag{2.49}$$

where $f_{l,k} = f(x_{l,k})$. Now, consider the complement space

$$W_l = V_{l+1} \ominus V_l. \tag{2.50}$$

In this particular setting, interpolating wavelets which form a basis of W_l can be simply chosen as [9]:

$$\psi_{l,k}(x) = \phi_{l+1,2k+1}(x), \quad k \in k_l^1 = \{0, 1, \ldots, 2^l - 2, 2^l - 1\}. \tag{2.51}$$

With wavelets defined as above, there exists a fast transform (to be described later) that maps function values to wavelet coefficients and *vice versa*.

Further decomposition of the space V_l until reaching V_{l_0} yields results similar to (2.12). Therefore, the interpolation function $I_l f$ can be written in the multi-scale representation

$$(I_l f)(x) = \sum_{k=0}^{2^{l0}} f_{l_0,k}\phi_{l_0,k}(x) + \sum_{j=l_0}^{l-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(x), \tag{2.52}$$

where $d_{j,k}$ are wavelet coefficients. For a large classes of functions, [9] has shown that $f = \lim_{l\to\infty} I_l f$, *i.e.*

$$f(x) = \sum_{k=0}^{2^{l0}} f_{l_0,k}\phi_{l_0,k}(x) + \sum_{j\geq l_0}^{\infty}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}(x), \tag{2.53}$$

which represents the decomposition of f into a superposition of contributions from levels $l \geq l_0$. Information of these contributions is contained in the coefficients $\{f_{l_0,k}\}_{k\in k_{l_0}^0}$ (the function values at the coarsest level) and wavelet coefficients on the finer scales $\{\{d_{l,k}\}_{k\in k_l^1}\}_{l\geq l_0}$.

Note that in the context of interpolating wavelets, each basis function is associated with one grid point. To be more specific, the scaling function $\phi_{j,k}(x)$ is associated with the grid point $x_{j,k}$ of V_j and the wavelet function $\psi_{j,k}$ is associated with the grid point $x_{j,k}^1$ of

$$W_j = \{x_{j,k}^1 = x_{j+1,2k+1} \ : \ k \in k_j^1\}, \tag{2.54}$$

the complement grid of V_j in V_{j+1}. It is obvious that the grid V_l is equivalent to $V_{l_0} \cup W_{l_0} \cup \cdots \cup W_{l-1}$. Due to this decomposition of the grid and the notation used in (2.54), it can be seen that for any given dyadic point $x_{q,k} \notin V_{l_0}$, there is a unique index j, k, where $j < q$, such that $x_{j,k}^1 = x_{q,k}$. In this way, an action on the grid, *e.g.* discarding and including grid points, also implies an appropriate action on the index of the basis functions and *vice versa*.

2.6 Fast interpolating wavelet transform

For notational simplicity, lets denote (2.52) by f^l. Consider the difference of the approximation of f between the two consecutive space V_l and V_{l+1}:

$$w^l = f^{l+1} - f^l = \sum_{k=0}^{2^{l+1}} f_{l+1,k}\phi_{l+1,k} - \sum_{k=0}^{2^l} f_{l,k}\phi_{l,k} = \sum_{k=0}^{2^l-1} d_{l,k}\psi_{l,k}(x). \tag{2.55}$$

From (2.51), (2.47), and the interpolation property (property i)), the wavelet coefficients obtained by evaluating $w_j(x)$ at $x^1_{j,k}$ are given by

$$d_{l,k} = f_{l+1,2k+1} - \sum_{r=0}^{2^l} \underbrace{\phi_{l,r}(x^1_{l,k})}_{=\, h^{l,r}_{2k+1}} f_{l,r}. \tag{2.56}$$

It can be verified that, for each k, there are only p filter coefficients $h^{l,r}_{2k+1}$ having non-zero values and they are those with $r \in \mathcal{X}_{l,k}$ (see (2.30) for the definition of the index set $\mathcal{X}_{l,k}$). From (2.47), (2.32) and (2.31), it can be shown that

$$\sum_{r=0}^{2^l} h^{l,r}_{2k+1} f_{l,r} = \sum_{r=\min \mathcal{X}_{l,k}}^{\max \mathcal{X}_{l,k}} L^k_{l,r}(x^1_{l,k}) f_{l,r} = \pi_{l,k}(x^1_{l,k})\pi_{l,k}(x_{l+1,2k+1}) \tag{2.57}$$

Notice that to compute $d_{l,k}$, one requires only $f_{l+1,2k+1}$ and those $f_{l,r}$ for $r \in \mathcal{X}_{l,k}$ (see figure 2.11 for illustration). In addition, it is evident that the wavelet coefficient indicates how large $f_{l+1,2k+1}$ deviates from the value predicted by the local interpolation polynomial.

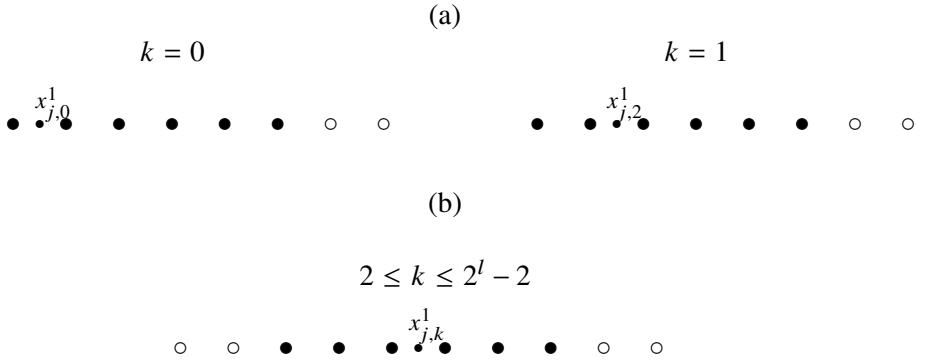


FIGURE 2.11: Points required in the calculation of wavelet coefficients $d_{j,k}$ represented by the small filled circles for $\bar{p} = 6$ (a) $k = 0, 1$, (b) $k = 2, \ldots, 2^{l-1} - 2$. Note that the larger circles denote the points in V_l. The filled circles represent points participating in the calculation of $d_{j,k}$.

By applying (2.56) until reaching the coarsest level, one obtains an algorithm for determining the wavelet coefficients of f^l of any given f. The interpolating wavelet transform mapping the function values to the wavelet coefficients is summarized in the Algorithm 1. Notice that the scaling function coefficients $f_{j,k}$ are not altered by the wavelet transform (they correspond to the function values at their associated points.). Thus, it is not necessary to perform the the wavelet

transform in a top-to-bottom manner (*i.e.* from level $j = l - 1$ down to l_0). This feature is unique to this particular wavelet basis and useful in particular for compression of functional information. The transform being done in the top-to-bottom manner has the advantage in that one can overwrite in place $f_{j+1,2k+1}$ by $d_{j,k}$ and hence require only one vector for storage. For a given set of wavelet

Algorithm 1 *1DFWT*

Given function values $\{f_{l,k}\}_{k\in k_l^0}$
for $j = l - 1$ to l_0 **do**
  for $k = 0$ to $2^{j-1} - 1$ **do**
    $m = 2^{l-j-1}(2k + 1)$
    $f_{l,m} = f_{l,m} - \sum_{r=0}^{2^j} h_{2k+1}^{j,r} f_{l,r2^{l-j}}$
  end for
end for

coefficients, one can obtain the associated function values using the inverse wavelet transform. The procedure starts from the coarsest level $j = l_0$ to obtain $f_{j+1,2k+1}$ by adding $d_{j,k}$ to the value predicted by the local polynomial, *i.e.*

$$f_{j+1,2k+1} = d_{j,k} + \sum_{r=0}^{2^l} h_{2k+1}^{l,r} f_{l,r}. \tag{2.58}$$

By applying (2.58) recursively until reaching level $j = l - 1$, one obtains the function values on the grid level l. It is important to note that the inverse wavelet transform must be performed strictly in the bottom-to-top order since the data $\{f_{j,k}\}_{k\in k_l^0}$ must be available before $f_{j+1,2k+1}$ can be computed. Algorithm 2 summarizes the inverse wavelet transform procedure. Note that this algorithm assumes that the wavelet coefficient $d_{j,k}$ is stored in the entry $f_{l,2^{l-j-1}(2k+1)}$ of the input data.

Algorithm 2 *1DIWT*

Given wavelet coefficients $\{f_{l,k}\}_{k\in k_l^0}$
for $j = l_0$ to $l - 1$ **do**
  for $k = 0$ to $2^j - 1$ **do**
    $m = 2^{l-j-1}(2k + 1)$
    $f_{l,m} = f_{l,m} + \sum_{r=0}^{2^j} h_{2k+1}^{j,r} f_{l,2^{l-j}r}$
  end for
end for

It can be seen that the calculation of one wavelet coefficient requires $\bar{p}$ operations. The number of operations required for the transform or its inverse at the level j is of order $\bar{p}2^j$ and thus the total number of operations is of order $\bar{p}\sum_{j=l_0}^{l-1} 2^j = \bar{p}(2^l - 1) \sim O(2^l)$. This indicates that the cost of the wavelet transform or its inverse is proportional to the total number of data values. The proportionality constant varies linearly with the order of basis.

2.6.1 Remarks

The calculation of wavelets coefficients $d_{l,k}$ requires the associated nonzero filter coefficients $h^{l,r}_{2k+1}, r = \min \mathcal{X}_{l,k}, \ldots, \max \mathcal{X}_{l,k}$. Since the grid is composed of equally distant points, the values of filter coefficients are independent of the level l. For $\bar{p}/2 - 1 \leq k$ and $m \leq 2^l - \bar{p}/2$, the filter coefficients $\{h^{l,r}_{2m+1}\}_{r \in k^0_l}$ are just the shifted versions of $\{h^{l,r}_{2k+1}\}_{r \in k^0_l}$. Furthermore, the filter coefficients associated with $2^l - \bar{p}/2 \leq k \leq 2^l - \bar{p}/2 - 2$ correspond to those of $2^l - k$ with reverse-order entries. Thus, it is sufficient to calculate the filters $\{h^{l,r}_{2k+1}\}$ associated with $k = 0, 1, \cdots, \bar{p} - 1$. Table 2.1 lists, for example, the non-zero filter coefficients $h^{l,r}_{2k+1}, r \in X_{l,r}$, needed for the calculation of wavelet coefficients for $\bar{p} = 6$.

Table 2.1: The vector of nonzero filter coefficients $h^{l,r}_{2k+1}$ For $\bar{p} = 6$.

r	0	1	2	3	4	5
$k = 0$	63/256	315/256	−105/128	63/128	−45/256	7/256
$k = 1$	7/256	105/256	105/128	−35/128	21/256	−3/256
$k = 2$	3/256	−25/256	75/128	75/128	−25/256	3/256

2.7 Sparse Wavelet Representation

Using the one-dimensional interpolating scaling and wavelet functions basis, the one-dimensional wavelet transform of a continuous function $f(x)$, given by Eq. (2.52), is now rewritten identifying J_0 and J as the minimum and maximum resolution levels, respectively,

$$f(x) \approx f^J(x) = \sum_{k=0}^{2^{J_0}} f_{J_0,k}\phi_{J_0,k}(x) + \sum_{j=J_0}^{J-1}\sum_{k=0}^{2^J-1} d_{j,k}\psi_{j,k}(x). \tag{2.59}$$

Because of the basis functions have to satisfy the interpolation property i), the scaling function coefficients $f_{J_0,k}$ correspond to function values at the associated grid points, $f_{J_0,k} = f(x_{J_0}, k)$. The maximum index of the coarsest grid level is $N_0 \geq \bar{p} + 1$. Therefore, each basis function is related to a single point in a regular grid V_J with locations $\{x_k = k(N_0 2^J)^{-1} : k \in k^0_J\}$, with $k^0_J = \{0, \ldots, N_0 2^J\}$.

The wavelet coefficients provide a direct measure of the local approximation error at each grid point. Equation (2.59) is rewritten introducing a user-defined accuracy, called wavelet threshold parameter ε:

$$f(x) \approx f^J(x) = \sum_{k=0}^{2^{J_0}} f_{J_0,k}\phi_{l_0,k}(x) + \sum_{j=l_0}^{J-1}\sum_{\{k:|d_{j,k}|\geq\varepsilon\}} d_{j,k}\psi_{j,k}(x) + \underbrace{\sum_{j=l_0}^{J-1}\sum_{\{k:|d_{j,k}|<\varepsilon\}} d_{j,k}\psi_{j,k}(x)}_{R^J_\varepsilon}. \tag{2.60}$$

Discarding the R^J_ε yields the sparse wavelet representation (SWR) of f, f^J_ε, with the local approximation error being no greater than ε at any point. In this manner any function can be

represented using a minimal number of basis functions while maintaining a prescribed accuracy for a resolution J sufficiently large.

Note that in the context of interpolating wavelets, each basis function is associated with one grid point. To be more specific, the scaling function $\phi_{j,k}(x)$ is associated with the grid point $x_{j,k}$ of V_j and the wavelet function $\psi_{j,k}(x)$ is associated with the grid point $x^1_{j,k}$ of $W_j = \{x^1_{j,k} = x_{j+1,2k+1} : k \in k^1_j\}$, which is the complement grid of V_j in V_{j+1}. The grid V_J is equivalent to $V_{J_0} \cup W_{J_0} \cup \ldots \cup W_{J-1}$. Due to this decomposition of the grid, it can be seen that for any given dyadic point $x_{q,k}$ not belonging to V_{J_0}, there is a unique index j, k, where $j < q$, such that $x^1_{j,k} = x_{q,k}$. In this way, an action on the grid, *e.g.* discarding and including grid points, also implies an appropriate action on the index of the basis and *vice versa*.

2.7.1 Grid adaption strategy

The grid points associated with wavelet amplitudes larger than the wavelet threshold parameter are called *essential* points:

$$\mathcal{V}_E = \{x_{0,k} \oplus \bigcup_{j>0} x_{j,k} : k \in \kappa^0_j\}, \quad \kappa_j = \{k \in k_j : |d_{j,k}| \leq \varepsilon\}. \tag{2.61}$$

The total number of essential points is N_E. To accommodate the possible advection and sharpening of solution features in time, the *neighboring* grid points are considered:

$$\mathcal{V}_B = \bigcup_j \mathcal{N}_{j,k_j}, \quad k_j = \{0, \ldots, N_0 2^J\}, \tag{2.62}$$

with the total number of neighboring points defined as N_B. The neighbors $\mathcal{N}_{j,k_j}$ of a grid point x_{j,k_j} are defined by

$$\mathcal{N}_{j,k_j} = \{x_{j+\tilde{n}_s,k_i-\tilde{n}_l 2^{\tilde{n}_s}}, \ldots, x_{j+\tilde{n}_s,k_i+\tilde{n}_l 2^{\tilde{n}_s}}\}, \tag{2.63}$$

with $\tilde{n}_s$ and $\tilde{n}_l$ are the extent of neighbors in scale and location, respectively. The refinement strategy (2.63) creates a 1-dimensional uniform refinement around a point, with spacing according to scale level $j + \tilde{n}_s$ and extending to $\tilde{n}_l$ points on scale level j. One level in scale and one neighbor in space are typically used, that is $\tilde{n}_s = \tilde{n}_l = 1$. Note this strategy is conservative, albeit quite robust. Alternative definitions which use solution information are possible and may provide additional reduction in the number of neighboring points used compared to (2.63).

The sparse grid $\mathcal{V}_A$ consists of the union of essential points and neighboring points

$$\mathcal{V}_A = \mathcal{V}_E \cup \mathcal{V}_B. \tag{2.64}$$

The solution is advanced only at points in $\mathcal{V}_A$, which are considered to be *active*. The total number of active points is the sum of essential and neighboring point counts, $N_A = N_E + N_B$. To complete stencils needed in various operations on the sparse grid, extra non-essential points are included and their values are determined purely from interpolation. Non-essential points are associated with wavelet coefficients of zero. The number of non-essential points is N_N, making the total number of points $N_T = N_A + N_N$.

Classical finite difference schemes are used to evaluate the derivatives on the adaptive irregular grid. The following procedure is used:

1. the irregular grid $\mathcal{V}$ must be found: it is composed also by the points required to complete the stencils at the different resolution levels to use the classical finite difference schemes. This means that, typically, the grid $\mathcal{V}$ is larger than the irregular grid associated with the SWR;

2. using the fast inverse interpolating wavelet transform, the function values are found on all the points belonging to $\mathcal{V}$;

3. through classical finite difference schemes, the derivatives of order n are evaluated at each grid point of the irregular sparse grid. The i^{th} derivative approximation with respect to x is indicated as $\frac{\partial^i f}{\partial x^i}$;

4. the interpolating wavelet transform is applied to the result to obtain the corresponding wavelet coefficients $d_{j,k}^{(i)}$.

By defining h as the minimum of the spacings between two consecutive stencil points, it can be demonstrated that the finite difference approximation performed in the step 3) is $O(h^n)$. Thus, at a grid point in which its associated stencil has no missing points, the truncation error of the scheme is approximatively $O(h^n)$. Note that, sometimes, the stencil to use the chosen finite difference scheme need to be completed, step 1). For a grid point in which its associated stencil contains points that do not belong to the irregular grid produced by the SWR, an additional error associated with the wavelet interpolation must be taken into account: it is roughly $O(\varepsilon/h^i)$. Estimate shows that the pointwise error of the derivative approximation has the following bound [55] that the following relation holds:

$$\left\| \frac{\partial^i f}{\partial x^i} - D_x^{(i)} f_\varepsilon^J \right\| \leq C N^{-\min((\bar{p}-i),n)/d}, \tag{2.65}$$

where C is a constant depending on f, d (the problem dimension), $\bar{p}$, and mildly on the threshold value ε. The above result suggests that in order not to lose accuracy, a finite difference of order $n \leq \bar{p} - i$ should be employed.

In the project, wavelet functions of fourth and sixth order are used and first and second derivatives are evaluated. To satisfy the last inequalities, difference schemes of order $n = 2$ are employed.

2.8 Dynamically adaptive algorithm to solve time dependent PDEs

The problem described by the following PDE is considered:

$$\frac{\partial u}{\partial t} = F(t, u, u_x, u_{xx}, \ldots), \tag{2.66}$$

where u is the dependent variable, while t and x are the independent variables (time and space, respectively). The initial condition u_{t_0} is built on the initial sparse grid $\mathcal{V}^0$. The dynamically adaptive algorithm consists in the following steps:

1. Equation (2.66) is integrated in time and the approximate solution at the time instant t_{n+1} ($u(t_{n+1})$) is found on the irregular sparse grid $\mathcal{V}^n$ by using the solution from the previous steps $u(t_{n-q})$, $q = 0, \ldots, n$;

2. the new sparse grid ($\mathcal{V}^{n+1}$) is based on the thresholding of the magnitudes of wavelet amplitudes of the current solution, $u(t_{n+1})$;

3. assign $\mathcal{V}^{n+1} \rightarrow \mathcal{V}^n$ and $u(t_{n+1-q}) \rightarrow u(t_{n-q})$, $q = 0, \ldots, n$ and go to step 1.

This thesis uses the Wavelet Adaptive Multi-Resolution Representation (WAMR) code to build dynamically adaptive grid. WAMR is written in Fortran 90: it has been extracted from the parallel version of the algorithm (pWAMR) able to solve the system of Navier-Stokes equations modelling continuum reactive and compressible flows on distributed memory parallel computers using the Message Passing Interface (MPI) standard [34].

One of the most important features of WAMR is the structure of the wavelet threshold parameter. It is defined considering that,

- all the dependent variables of a problem described by PDEs need to be built on the same adaptive grid;

- the order of magnitude of the dependent variables could be extremely different.

Therefore, the wavelet threshold parameter needs to be scaled as follows:

$$\varepsilon = \varepsilon_r \bar{f} + \varepsilon_a, \qquad \bar{f} = |\max(f(x))|, \tag{2.67}$$

where ε_r and ε_a are user assigned relative and absolute wavelet threshold parameters, while $\bar{f}$ represents the maximum value of the single variable defined along the x axis. In this sense, if n_v is the total number of variables, one has a vector of n_v scaled wavelet threshold parameters to be used in Eq. (2.60).

2.9 Parallel implementation

The wavelet theory in the multi-dimensional context and the main features of the pWAMR algorithm are described in the work of Grenga [16]. However, making reference to Fig. (2.12), the structure of the pWAMR implementation (taken by Grenga [16]) is briefly described. On the top of Fig. (2.12) there are the routines that describe the problem that need to be solved including partial differential equations, initial and boundary conditions and the problem parameters. These routines may change substantially from a problem to another. The next lower level contains the core of code. It is divided in three parts: (i) the pWAMR source code that includes all the routines needed to create the dynamically adaptive grid and discretize the equations, and the external

libraries for (ii) the time integration method, and (iii) chemistry and thermal-physical properties. All the above layers use the parallelization specifically realized to allow the usage of massive parallel computation. This level contains the routine that performs the grid synchronization and builds all the communications of data, and the one for load balancing. At the lowest level there is the MPI standard over which all the code is built.

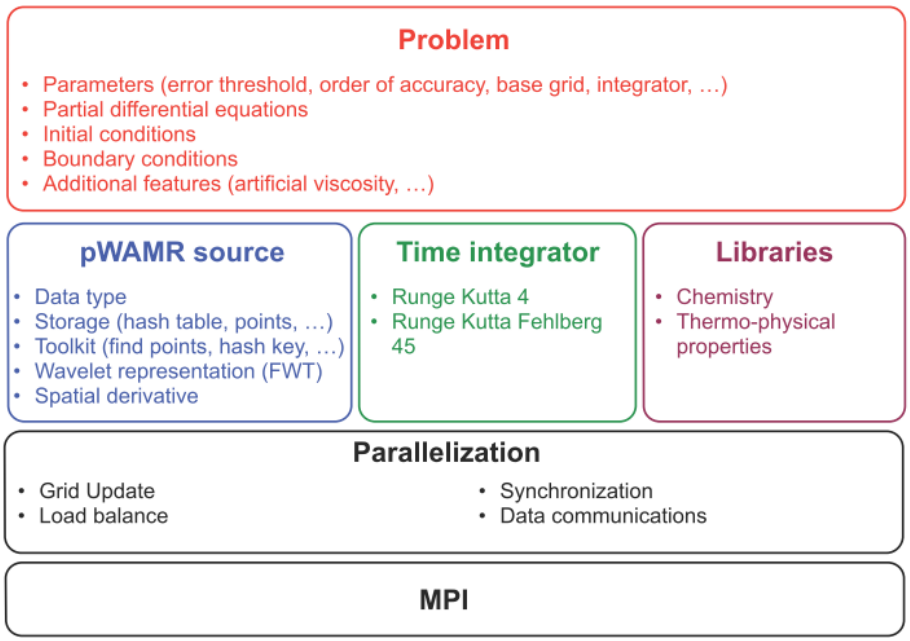


FIGURE 2.12: Structure of the pWAMR implementation [16].

CHAPTER 3

The G-Scheme Framework

The content of this chapter summarizes from the work of Valorani and Paolucci [49] about the G-Scheme as a tool for numerical integration of reactive mixture, where further details can be found. The G-Scheme was created with the aim to try to reduce the complexity related to the integration of the system dynamics developing a wide range of time scales. The idea at the base of the G-Scheme algorithm is that the system dynamics can develop very fast and very slow time scales, separated by a range of active time scales. In the algorithm, the dynamics is supposed to be decomposable into active, slow, fast and invariant (when applicable) subspaces. The G-Scheme introduces locally a curvilinear frame of reference, defined by a set of orthonormal basis vectors with corresponding coordinates, attached to this decomposition. The numerical integration of the dynamics is accomplished by solving a number of non-stiff DEs typically much smaller than the dimension of the original problem. The direct consequence is a saving in computational work. Section (3.1) is dedicated to the description of the main mathematical features behind the G-Scheme algorithm: the G-Scheme decomposition of the system dynamics is introduced in section (3.1.1), while the whole algorithm is presented in section (3.1.2).

One of the problematic of the G-Scheme is the large computational cost related to the eigenproblem evaluation: as will be shown in chapters (5)-(6), the largest part of the workload is due to the basis vectors and Jacobian matrix calculation. This negative aspect of the algorithm was also underlined by Grenga in several works [44, 46, 45]. In particular, with the aim to develop a multiscale adaptive reduction chemistry solver for computationally efficient modeling of a reactive flow, a comparison between an Hybrid Multi-Timescale method and the G-Scheme algorithm was proposed in [44] to study homogeneous autoignition and 1-D premixed spherical propagating flame calculations with detailed chemical kinetics. Grenga [44] underlined that the workload increases significantly when the number of species involved in a detailed chemical mechanism is larger. To overcome the limitations related to the calculation of the eigenproblem, Valorani [31] identified a criterion to decide if and when the reuse of the basis vectors is feasible. The reuse concept allows to increase significantly the computational efficiency of the G-Scheme solver with minimal accuracy losses. In the same work [31], the reuse strategy was tested for autoignition test cases of a methane/air mixture in a constant pressure batch reactor using different mechanisms of increasing size and complexity. As a result, it was proved that the basis reuse makes the solver

$2 \div 5$ times faster, depending on the mechanism. The main features of the reuse concept are summarized in section (3.2).

A tool to identify unambiguously and easily the most energetic scale at a given space location and time instant is represented by the Tangential Stretching Rate (TSR). Valorani [52] constructed a TSR parameter by combining the concepts of stretching rate with the decomposition of the local space in eigen-modes. Section (3.3) shows the guidelines to evaluate the TSR.

Finally, a comparison between the G-Scheme framework and the Approximate Inertial Manifold (AIM) approach (Akram and Raman [2]) is illustrated in section (3.4). The AIM approach is an alternative attempt with respect to the G-Scheme to represent the system dynamics in a subspace smaller than the entire state-space, with the similar goal to produce a considerable simplification in the study of the dynamics of an original systems. The difference and similarities of the two strategies are summarized in section (3.4.1).

3.1 Theory

A Cauchy problem defined by a set of autonomous ODEs is considered:

$$\frac{d\vec{x}(t)}{dt} = \vec{f}(\vec{x}(t)), \qquad \vec{x}(0) = \vec{x}_{t_0}, \tag{3.1}$$

where $t \in (t_0, t_f] \subset \mathbb{R}$ is the time, $\vec{x} \in \mathbb{R}^N$ is the state vector (where N is the dimension of the system), and $\vec{f} : E \subset \mathbb{R}^N \rightarrow \mathbb{R}^N$ is the nonlinear vector field. Considering an arbitrary time t_n, the state vector $\vec{x}(t)$ can be always expressed as the sum of $\vec{x}(t_n)$ and a perturbation vector $\Delta\vec{x}\,(\tau)$, $\vec{x}(t) = \vec{x}(t_n) + \Delta\vec{x}\,(\tau)$. The component-wise representation of the perturbation vector can be written in terms of curvilinear coordinates $\Delta\vec{x}\,(\tau) = \Delta\xi^i\,(\tau)\,\vec{a}_i\,(\tau) = \Delta\xi_j\,(\tau)\,\vec{a}^j\,(\tau)$ related to the sets of orthonormal covariant and contravariant basis vectors $\vec{a}_i\,(\tau)$ and $\vec{a}^j\,(\tau)$. This expansion allows to write Eq. (3.1) in terms of curvilinear coordinates:

$$\begin{aligned} &\frac{d\Delta\xi^k}{d\tau} = \vec{a}^k(\tau) \cdot \vec{f}\left(\vec{x}(t_n) + \Delta\xi^i(\tau)\,\vec{a}_i(\tau)\right) - \vec{a}^k(\tau) \cdot \frac{d\vec{a}_i(\tau)}{d\tau}\,\Delta\xi^i(\tau), \\ &\Delta\xi^k(0) = 0. \end{aligned} \tag{3.2}$$

To write the system (3.2) in the more convenient vector form, the matrices $A(\tau)$, $B(\tau)$ and the vector $\Delta\xi(\tau)$ are introduced and defined as

$$A(\tau) \equiv \left[\vec{a}_1(\tau)\ \cdots\ \vec{a}_i(\tau)\ \cdots\ \vec{a}_N(\tau)\right],\ B(\tau) \equiv \begin{bmatrix} \vec{a}^1(\tau) \\ \cdots \\ \vec{a}^j(\tau) \\ \cdots \\ \vec{a}^N(\tau) \end{bmatrix},\ \Delta\xi(\tau) \equiv \begin{bmatrix} \Delta\xi^1(\tau) \\ \cdots \\ \Delta\xi^j(\tau) \\ \cdots \\ \Delta\xi^N(\tau) \end{bmatrix}, \tag{3.3}$$

where $\vec{a}_i(\tau)$ are column vectors, $\vec{a}^j(\tau)$ are row vectors. The matrices $A(\tau)$ and $B(\tau)$ satisfy the property $A(\tau)\,B(\tau) = B(\tau)\,A(\tau) = I$, where I is the identity matrix. Therefore, the Cauchy

problem (3.1) becomes:

$$\begin{aligned} &d\Delta\xi/d\tau = B(\tau)\,\vec{f}\,(\vec{x}(t_n) + A(\tau)\,\Delta\xi(\tau)) - B(\tau)\,(dA(\tau)/d\tau)\,\Delta\xi(\tau), \\ &\Delta\xi(0) = \vec{0}. \end{aligned} \tag{3.4}$$

3.1.1 The G-Scheme decomposition

In the G-Scheme theory, the sets of orthonormal covariant and contravariant basis vectors $\vec{a}_i\,(\tau)$ and $\vec{a}^j\,(\tau)$ are equal to the right and left eigenvectors of the Jacobian matrix $J_{\vec{f}}$ of the nonlinear vector field, respectively. The characteristic time scales are the reciprocal of the eigenvalues λ_i of $J_{\vec{f}}$. The modes are ordered according to the magnitude of the complex eigenvalues of $J_{\vec{f}}$:

$$0 = \lambda_1 = \cdots = \lambda_E < |\lambda_{E+1}| \leq \cdots \leq |\lambda_{H-1}| \ll |\lambda_H| \leq \cdots \leq |\lambda_T| \ll |\lambda_{T+1}| \leq \cdots \leq |\lambda_N|,$$

where

$$\begin{aligned} 0 = \lambda_1 = \cdots = \lambda_E \quad &\text{identify the } N_E \text{ time scales in the invariant subspace } \mathbb{E}, \\ |\lambda_{E+1}| \leq \cdots \leq |\lambda_{H-1}| \quad &\text{identify the } N_H \text{ slow time scales in the slow subspace (head) } \mathbb{H}, \\ |\lambda_H| \leq \cdots \leq |\lambda_T| \quad &\text{identify the } N_A \text{ active time scales in the active subspace (heart) } \mathbb{A}, \\ |\lambda_{T+1}| \leq \cdots \leq |\lambda_N| \quad &\text{identify the } N_T \text{ fast time scales in the fast subspace (tail) } \mathbb{T}. \end{aligned} \tag{3.5}$$

Through this decomposition, the tangent space $\mathcal{T}_x$ is given by the direct sum of four subspaces:

$$\mathcal{T}_x = \mathbb{E} \oplus \mathbb{H} \oplus \mathbb{A} \oplus \mathbb{T}, \tag{3.6}$$

where the active subspace $\mathbb{A}$ contains all the intermediate, currently active (dynamic) time scales, all scales slower/faster than the active ones are confined in the subspaces $\mathbb{H}/\mathbb{T}$, and, if the system possesses E invariants, $\mathbb{E}$ is the subspace spanned by the directions associated with them. The number of modes in each subspace is defined as $N_E = \dim(\mathbb{E}) = E$, $N_H = \dim(\mathbb{H}) = H - E - 1$, $N_A = \dim(\mathbb{A}) = T - H + 1$, and $N_T = \dim(\mathbb{T}) = N - T$. Note that, because of this ordering, (possibly complex) eigenvalues with both negative and positive real parts can be found in $\mathbb{H}$ and $\mathbb{A}$, whereas we expect the eigenvalues in $\mathbb{T}$ to have dominant negative real part. Indeed, this is the distinguishing feature of the class of problems for which the G-Scheme is expected to perform most effectively. The ratio ϵ_H is a measure of the spectral gap between the slow and active subspaces, while the ratio ϵ_T is a measure of the spectral gap between the active and fast subspaces. The are defined as:

$$\epsilon_H \equiv \frac{|\lambda_{H-1}|}{|\lambda_H|} \ll 1 \quad \epsilon_T \equiv \frac{|\lambda_T|}{|\lambda_{T+1}|} \ll 1, \tag{3.7}$$

Since the G-Scheme approximates the contributions of the very-slow and very-fast time scales with asymptotic corrections, it is expected that the accuracy and efficiency of the scheme will be higher for larger spectral gaps.

The controlling (driving) time scale of the dynamics is the fastest of the active time scales present in $\mathbb{A}$, and will be of the order of the reciprocal of $|\lambda_T|$. The G-Scheme in fact consists of an effective implementation of the merging of slow and fast reduced models. In general, for any arbitrary but fixed accuracy, there are slow and fast time scales whose dynamics are negligible and thus these modes are *not dynamically active*. Nevertheless, these near-frozen and near-equilibrium modes cannot be ignored since they play crucial roles, and thus their influence couple with those of the active modes.

The basic expression of the state vector $\vec{x}(t) = \vec{x}(t_n) + \Delta\vec{x}(\tau)$ can be rewritten noting that the system may have E invariant quantities, represented by ξ^c, with $c = 1, \ldots, E$. In this case the contributions to $\Delta\vec{x}(\tau)$ lying in the invariant subspace $\mathbb{C} = \mathbb{E}$ spanned by the vectors $\vec{a}_c$ can be isolated from those in the orthogonal complement $\mathbb{NC}$ to $\mathbb{C}$ ($\mathbb{R}^N = \mathbb{NC} \oplus \mathbb{C}$) spanned by the vectors $\vec{a}'_c$, where $c' = E + 1, \ldots, N$:

$$\vec{x}(t) = \vec{x}(t_n) + \Delta\vec{x}(\tau) = \vec{x}(t_n) + A(\tau)\Delta\xi(\tau) = \vec{x}(t_n) + \Delta\xi^{c'}(\tau)\,\vec{a}_{c'}(\tau) + \Delta\xi^{c}(\tau)\,\vec{a}_{c}(\tau). \tag{3.8}$$

The last term of Eq. (3.8) is equal to zero because ξ^c is invariant. The term $\Delta\xi^{c'}(\tau)\,\vec{a}_{c'}(\tau)$ can be seen as the sum of slow, active and fast contributions.The final form of state vector is:

$$\vec{x}(t) = \vec{x}(t_n) + \Delta\xi^{h}(\tau)\,\vec{a}_{h}(\tau) + \Delta\xi^{a}(\tau)\,\vec{a}_{a}(\tau) + \Delta\xi^{t}(\tau)\,\vec{a}_{t}(\tau), \tag{3.9}$$

where $\Delta\xi^h(\tau)$, $\Delta\xi^a(\tau)$ and $\Delta\xi^t(\tau)$ are the amplitudes of the perturbation vector along the slow, active and fast directions, $\vec{a}_h(\tau)$, $\vec{a}_a(\tau)$ and $\vec{a}_t(\tau)$, respectively. The decomposition (3.9) induces a partitioning of (3.4) that, after a linearization, is rewritten as:

$$\frac{d}{d\tau}\begin{pmatrix} \Delta\xi^h(\tau) \\ \Delta\xi^a(\tau) \\ \Delta\xi^t(\tau) \end{pmatrix} = \begin{pmatrix} B^h(\tau) \\ B^a(\tau) \\ B^t(\tau) \end{pmatrix}\vec{f}\,(\vec{x}(t_n)) + \Lambda \begin{pmatrix} \Delta\xi^h(\tau) \\ \Delta\xi^a(\tau) \\ \Delta\xi^t(\tau) \end{pmatrix}, \tag{3.10}$$

$$\begin{pmatrix} \Delta\xi^h(0) \\ \Delta\xi^a(0) \\ \Delta\xi^t(0) \end{pmatrix} = \begin{pmatrix} \vec{0}^h \\ \vec{0}^a \\ \vec{0}^t \end{pmatrix}, \tag{3.11}$$

where the matrix Λ is

$$\Lambda = \begin{pmatrix} \Lambda^h_h\,(\vec{x}(t_n),\tau) & \Lambda^a_h\,(\vec{x}(t_n),\tau) & \Lambda^t_h\,(\vec{x}(t_n),\tau) \\ \Lambda^h_a\,(\vec{x}(t_n),\tau) & \Lambda^a_a\,(\vec{x}(t_n),\tau) & \Lambda^t_a\,(\vec{x}(t_n),\tau) \\ \Lambda^h_t\,(\vec{x}(t_n),\tau) & \Lambda^a_t\,(\vec{x}(t_n),\tau) & \Lambda^t_t\,(\vec{x}(t_n),\tau) \end{pmatrix}, \tag{3.12}$$

with $\Lambda^y_x = B^y J_{\vec{f}} A_x \quad y, x = h, a, t.$

3.1.2 The G-Scheme algorithm step-by-step

The G-Scheme algorithm is presented in Fig. (3.1). At any time step, the G-Scheme estimates the contribution of the fastest modes with the aim of finding the size of the fast subspace, $\mathbb{T}$, by comparing the contribution of these modes with the magnitude of the state variables.

The magnitude of the time step used by the G-Scheme is one of the major features and originalities of this method, defined as $\Delta t = \gamma/|\lambda_{T(\vec{x}(t_n))}|, \gamma \leq 1$. It can be several order of magnitude larger than the fastest time scale of the system. This means that, for stiff problems, the time step used by the G-Scheme would be several orders of magnitude larger than that one required by other numerical integration methods. The specific time step used by the G-Scheme depends on the state variables and it can sweep over a wide range of order of magnitude during the time evolution of the process. Indeed, in order to preserve the required accuracy, the used time step is of the same order of the fastest scale of the problem only at the stage of the process wherein the variation of the state variables occurs at the fastest scale.

Once the dimension of $\mathbb{T}$ is determined, the size of the other subspaces can be also evaluated. The near-invariant subspace, $\mathbb{E}$, contains the modes having a ratio between the driving time scale and the time scale smaller than the machine precision. Therefore, their contributions are negligible, or rather are smaller than the machine precision.

The slow modes are identified using the same criteria of the fast modes. The contribution of these modes is compared with the magnitude of the state variables in order to satisfy the required accuracy. It also depends on Δt used.

The set of identified active modes constitutes a set of non-stiff ODE's that is solved for $\tau \in \Omega = (0, \Delta t]$:

$$\frac{d\Delta\xi^a}{d\tau} = B^a(t_n)\,\vec{f}\,(\vec{x}(t_n) + A_a(t_n)\,\Delta\xi^a(\tau))\,, \qquad \Delta\xi^a(0) = \vec{0}^a. \tag{3.13}$$

Equation (3.13) is integrated using the RK4 scheme. Then, the state vector is updated:

$$\vec{x}^a(t_{n+1}) = \vec{x}(t_n) + A_a(t_n)\,\Delta\xi^a\,(\Delta t)\,; \tag{3.14}$$

the head correction is estimated:

$$\Delta\xi^h_{\mathrm{FF}}(\Delta t) = \Delta t\ B^h(t_n)\left[I + \frac{1}{2}\Lambda^h_h\,(\vec{x}(t_n), t_n)\,\Delta t\right]\vec{f}\,(\vec{x}(t_n))\,; \tag{3.15}$$

the state vector is updated:

$$\vec{x}^h(t_{n+1}) = \vec{x}^a(t_{n+1}) + A_h(t_n)\,\Delta\xi^h_{\mathrm{FF}}(\Delta t). \tag{3.16}$$

The tail correction is estimated:

$$\Delta\xi^t_{\mathrm{SIM}(t_n)}(\Delta t) = -\left(B^t(t_n)J(\vec{x}(t_n))A_t(t_n)\right)^{-1}B^t\,(t_n)\,\vec{f}(\vec{x}^h(t_n)); \tag{3.17}$$

Then, it is applied to project the solution onto the subspace obtained using the bases at t_n:

$$\vec{x}^t(t_{n+1}) = \vec{x}^h(t_{n+1}) + A_t(t_n)\,\Delta\xi^t_{\mathrm{SIM}(t_n)}(\Delta t), \tag{3.18}$$

The integration time step can be considered completed. Therefore, the Jacobian matrix $J(\vec{x}^t(t_{n+1}))$, the eigenvalues $\lambda_i(t_{n+1}) = \lambda_i(\vec{x}^t(t_{n+1}))$, and the set of new basis vectors $A(t_{n+1}) = A(\vec{x}^t(t_{n+1}))$ and $B(t_{n+1}) = B(\vec{x}^t(t_{n+1}))$ are calculated. If is was necessary (*i.e.*, if the fast subspace changes) the bases rotation correction must be applied to find the state $\vec{x}(t_{n+1})$. This is done by projecting $\vec{x}^t(t_{n+1})$ located on the manifold evaluated at t_n onto the manifold evaluated at t_{n+1}:

$$\vec{x}(t_{n+1}) = \vec{x}^t(t_{n+1}) + A(t_{n+1})\,\Delta\xi_{\mathrm{SIM}(t_{n+1})}(\Delta t), \tag{3.19}$$

where the basis rotation correction is estimated as:

$$\Delta\xi_{\mathrm{SIM}(t_{n+1})}(\Delta t) = -\left(B^t(t_{n+1})J(\vec{x}^t(t_{n+1}))A_t(t_{n+1})\right)^{-1} B^t\,(t_{n+1})\,\vec{f}(\vec{x}^t(t_{n+1})). \tag{3.20}$$

The new state is reached and a new integration time step can start with the definition of the subspaces.

The parameters $rtol_{head}$, $rtol_{tail}$, $atol_{head}$ and $atol_{tail}$ are introduced to define the relative and absolute values of the total variation of the state variables over the time interval admitted for the tail and head corrections. They are used to form a threshold vector which allows the identification of the subspaces $\mathbb{A}$, $\mathbb{H}$ and $\mathbb{T}$.

3.2 The basis vectors reuse

To avoid a frequent update of the CSP basis Valorani et al. [31] observed that an effective and computationally inexpensive sensor of the CSP basis rotation is the norm of the directional derivative of the vector field $\vec{f}$ evaluated at the state vector $\vec{y}$ along a $\vec{p}$ direction that perturbs all the entries of the state vector itself. Hence, during the numerical integration of a set of DEs using the G-Scheme, it is tempting to update the CSP basis only when the norm of $\vec{f}_{\vec{p}}(\vec{y})$ is larger that a prescribed threshold (typically 35%). This strategy allows to have a temporal update of the CSP basis only when rotations of the CSP basis are significant, namely when the effects of nonlinearities are relevant.

3.3 Tangential Stretching Rate

The Tangential Stretching Rate (TSR) represents a useful tool to identify unambiguously and easily the most energetic scale at a given space location and time instant. The complete theory behind the construction of this parameter can be found in [52], while the main features are reported in this Section.

The Cauchy problem (3.1) is again considered and the small perturbation vector $\vec{v}(t)$ is introduced:

$$\vec{v}(t) := \lim_{|\hat{\varepsilon}|\to 0} \frac{\vec{x}_2(t) - \vec{x}_1(t)}{|\hat{\varepsilon}|} \tag{3.21}$$

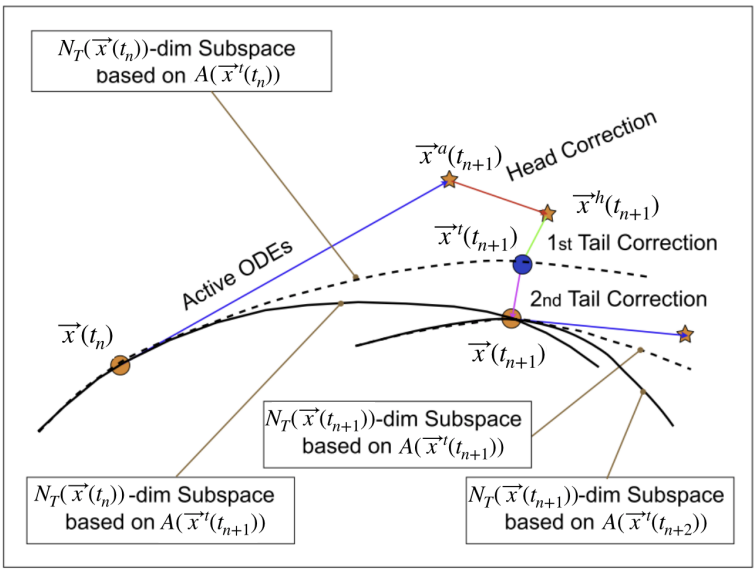


FIGURE 3.1: The G-Scheme step-by-step [49] starting from a given state $\vec{x}(t_n)$ on a subspace of dimension evaluated at t_n: orange stars denote intermediate new states before the application of head or tail corrections, the blue circle denotes the new state after head and tail corrections onto the subspace evaluated at t_n and where the basis vectors are subsequently updated to t_{n+1}, orange circle denotes the new state $\vec{x}(t_{n+1})$ and the location where the subspace dimension possibly changes. Note that in reality the orange circles are not exactly on the SIM; we're actually calculating the Approximate SIM (ASIM), identified by the eigenvectors (and not ideal basis vectors) of the Jacobian of the vector field. We do not show both the SIM and ASIM so as not to make the figure unduly complex.

It represents the scaled difference between the two trajectories $\vec{x}_1(t)$ and $\vec{x}_2(t)$ with respect to $\hat{\varepsilon} = \vec{x}_2(0) - \vec{x}_1(0)$. The dynamic of $\vec{v}(t)$ evolves approximatively as the following linear dynamic system shows:

$$\frac{d\vec{v}(t)}{dt} = J_{\vec{f}}(\vec{x}_1(t)) \cdot \vec{v}(t), \qquad \vec{v}(0) = 1 \tag{3.22}$$

where $J_{\vec{f}}(\vec{x}_1(t))$ is the Jacobian of the vector field $\vec{f}$ evaluated along the trajectory $\vec{x}_1(t)$ and 1 is a unit vector at $t = 0$ taken along any direction. By introducing the norm of $\vec{v}(t)$ and the unit vector $\tilde{u}(t) := \vec{v}(t)/v(t)$, the Eq. (3.22) is rewritten as:

$$\frac{dv(t)}{dt} = \left(\frac{\vec{v}(t) \cdot J_{\vec{f}}(\vec{x}_1(t)) \cdot \vec{v}(t)}{v(t)^2} \right) v(t) = (\tilde{u}(t) \cdot J_{\vec{f}}(\vec{x}_1(t)) \cdot \tilde{u}(t)) v(t) = \omega_{\tilde{u}}(t) v(t), \quad v(0) = 1 \tag{3.23}$$

The parameter $\omega_{\tilde{u}}(t)$ is the local stretching rate of the dynamics at time t, evaluated along the direction identified by $\tilde{u}(t)$: it represents the rate at which the perturbation is amplified (positive value of $\omega_{\tilde{u}}(t)$) or damped (negative value of $\omega_{\tilde{u}}(t)$) with time. By setting the unit vector $\tilde{\tau}$ spanning the vector field direction, the Tangential Stretching Rate (TSR) is introduced [1] and written as

$$\omega_{\tilde{\tau}}(t) := \tilde{\tau}(t) \cdot J_{\vec{f}}(\vec{x}_1(t)) \cdot \tilde{\tau}(t). \tag{3.24}$$

By combining the TSR definition with the eigen-decomposition of the Jacobian matrix of the vector field one has:

$$\omega_{\tilde{\tau}}(t) = \sum_{s=E+1}^{H-1} \left(\frac{g^s}{f}\right)^2 \lambda_s + \sum_{a=H}^{T} \left(\frac{g^a}{f}\right)^2 \lambda_a + \sum_{r=T+1}^{N} \left(\frac{g^r}{f}\right)^2 \lambda_r \tag{3.25}$$

where the labels s, a and r denote the slow, active and fast modes, respectively.

3.4 Approximate Inertial Manifolds

The model reduction obtained through the G-Scheme allows to solve a number of DEs typically much smaller than the dimension of the original problem, by integrating only the active DoFs. With the similar goal to represent the system dynamics in a subspace smaller than the entire state-space producing a considerable simplification in the study of the dynamics of the original system, an alternative approach is proposed by Akram and Raman [2]: it is based on the attempt to identify and build an Approximate Inertial Manifold (AIM) to study canonical flows. They can be described by PDEs of the form

$$\frac{d\vec{v}(t)}{dt} + \mathcal{A}\vec{v}(t) + \mathcal{F}(\vec{v}(t)) = 0 \qquad \vec{v}(t=0) = \vec{v}_0, \tag{3.26}$$

where $\mathcal{A}$ and $\mathcal{F}$ are a linear and nonlinear operator, respectively, and $\vec{v}_0$ is the initial condition. Inertial Manifolds (IM) are finite-dimensional, invariant manifolds and attract exponentially all solutions of Eq. (3.26). A projection operator $\mathcal{P}$ is introduced to partition the state-space into resolved $\vec{u}(t)$ and unresolved variables $\vec{w}(t)$

$$\vec{u}(t) = \mathcal{P}\vec{v}(t), \quad \vec{w}(t) = (I - \mathcal{P})\vec{v}(t) = \mathcal{Q}\vec{v}(t), \quad \vec{v}(t) = \vec{u}(t) + \vec{w}(t). \tag{3.27}$$

The projection operator $\mathcal{P}$ is defined by considering the first m eigenfunctions of the linear operator $\mathcal{A}$. Applying the projection operator to the discrete governing equation, the evolution equations for the resolved and unresolved fields are obtained, respectively:

$$\frac{d\vec{u}(t)}{dt} + \mathcal{A}\vec{u}(t) + \mathcal{P}\mathcal{F}(\vec{v}(t)) = 0, \quad \vec{u}(t=0) = \mathcal{P}\vec{v}_0, \tag{3.28}$$

$$\frac{d\vec{w}(t)}{dt} + \mathcal{A}\vec{w}(t) + \mathcal{Q}\mathcal{F}(\vec{v}(t)) = 0, \quad \vec{w}(t=0) = \mathcal{Q}\vec{v}_0. \tag{3.29}$$

The IM approximation is formulated to solve the resolved and unresolved fields: the dynamics of $\vec{w}(t)$ are directly determined by the dynamics of $\vec{u}(t)$, *i.e.* the unresolved $\vec{w}(t)$ state variables are slaved to the resolved $\vec{u}(t)$ through the algebraic constrain obtained by imposing $d\vec{w}(t)/dt = 0$ in Eq. (3.29), namely

$$\vec{w}(t) = -\mathcal{A}^{-1}\mathcal{Q}\mathcal{F}(\vec{u}(t) + \vec{w}(t)). \tag{3.30}$$

Equation (3.30) is solved iteratively. The reduced order model described by Eqs. (3.28)-(3.30) is so obtained.

3.4.1 Comparison with the G-Scheme

The AIM plays a similar role of the active subspace of the G-Scheme. Both the approaches have the main goal to simplify the system dynamics, by representing it in a smaller subspace. Some relevant features of the two approaches can be compared.

- The G-Scheme is able to find the active subspace by evaluating the eigensystem associated with the Jacobian matrix of the full nonlinear vector field. On the contrary, the AIM is found with respect to the sole linear part of the vector field;
- no a-priori criterion are available to identify the dimension of the AIM. This dimension is constant in time (quite so near to the steady state) and it is given by m. In the G-Scheme framework the dimension of the active subspace is time varying (far from steady state) and must be such that contribution of the fast subspace over the driving time scale is negligible;
- in the G-Scheme the eigenmodes are related to physical space, while in AIM approach they are related to Fourier space.

CHAPTER 4

Space-Time Adaptive Resolution

In this chapter the coupling between the non-parallel version of WAMR and the G-Scheme is proposed and introduced by the author. The basic coupling of the two algorithms is illustrated and described in section (4.1). During the time integration of PDEs describing a particular physical problem, WAMR can be used to represent the solutions at each time instant on a grid composed by a reduced number of points, that is built by deleting the points where the wavelet functions having an amplitude lower than a prescribed parameter are localized. The efficiency of the WAMR data compression is quantified through the wavelet compression ratio, defined in section (4.1.1). In the proposed coupled scheme, the DoFs generated by WAMR are analysed at each time instant by the G-Scheme with the goal to identify only a few number of DoFs - namely the active DoFs - to be integrated in time. The G-Scheme capability to select only a few number of active DoFs is quantified though a second index, introduced in section (4.1.1). The most important factors affecting the G-Scheme performance from the computational point of view are summarized in section (4.2): this part of the work is taken by the work of Valorani [31]. Section (4.3) is dedicated to the description of the main strategies implemented in this thesis to improve the G-Scheme performance. The coupling between WAMR and the G-Scheme, that is the core of the present thesis, is proposed also as a possible useful idea to reach this goal. Improvements of the performance are obtained through the reuse of the G-Scheme basis vectors. Moreover, further advantages can be obtained if the G-Scheme is used in classical Operator Splitting techniques. The implementation of the G-Scheme in a simple splitting technique is described in section (4.3.1).

4.1 Coupling of WAMR and the G-Scheme

To describe the coupling between WAMR and the G-Scheme, a nonlinear autonomous system of PDEs is considered

$$\frac{\partial \vec{y}}{\partial t} = \vec{f}(\vec{y}(x,t)), \qquad \vec{y}(t_0) = \vec{y}_{t_0}, \qquad \vec{y} \in \mathbb{R}^N \tag{4.1}$$

where t is the time, x is the space variable, $\vec{f}$ is a nonlinear vector field, $\vec{y}(x,t)$ is the state vector where the n_v dependent variables are stored and $\vec{y}(t_0) = \vec{y}_{t_0}$ is the initial condition. The basic coupling is represented in Fig. (4.1):

1. The wavelet analysis allows to obtain a compressed state vector built on an adaptive grid composed by N_T number of points ($\vec{y}_{N_{T_{new}}}(t_n)$), where N_T is given by the sum of the essential, neighbouring and non-essential points. Note that, as just specified in chapter (2), the non-essential points are temporarily used to locally complete the stencils to compute spatial derivatives. This means that they are not integrated in time. The new grid, generated by WAMR, is identified with the subscript $N_{T_{new}}$, while the old one is $N_{T_{old}}$;

2. the compressed state vector is analysed by the G-Scheme: the slow, fast, active and invariant subspaces are identified and only the active DoFs are integrated in time with an explicit Runge-Kutta method;

3. the solution at $t = t_{n+1}$, namely $\vec{y}_{N_{T_{new}}}(t_{n+1})$, is again analysed by WAMR to perform a new data compression;

4. the compression in space and the G-Scheme model reduction is repeated until a particular convergence criterion for the time integration is satisfied. For instance, if the system steady state is looked for, the loop can be broken by requiring that the RMS of the equations (4.1) RHS is lower than a prescribed value.

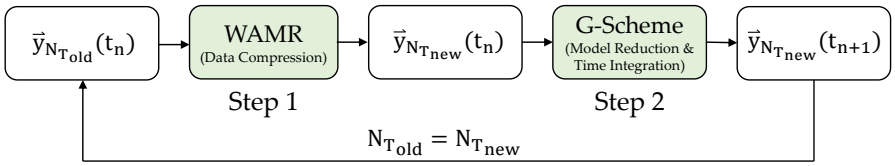


FIGURE 4.1: Basis coupling between WAMR and the G-Scheme framework.

4.1.1 The compression ratios

To quantify the capability of WAMR ant the G-Scheme to reduce the number of DoFs, some indices are introduced.

A useful tool to evaluate the efficiency of the wavelet compression is represented by the compression degree π_w, defined as

$$\pi_w = 1 - \eta_w = 1 - \frac{N_T}{N_{ref}}, \tag{4.2}$$

where η_w is the ratio between the total number of grid points of the adaptive mesh and a reference uniform grid having the same minimal spacing. Largest values of π_w correspond to largest data compression. In the same way, the G-Scheme efficiency can be evaluated by the index π_{gs}:

$$\pi_{gs} = 1 - \eta_{gs} = 1 - \frac{N_A}{N_{dof_w}}, \tag{4.3}$$

where η_{gs} represents the ratio between the number of DoFs integrated by the G-Scheme (N_A, the active DoFs) and generated by WAMR ($N_{dof_w} = N_T \times n_v$). Largest values of π_{gs} correspond to larger G-Scheme efficiency: a few number of active modes are generated and integrated. The combined overall compression efficiency is:

$$\pi_O = 1 - \eta_w \eta_{gs} = 1 - \eta_O = 1 - \frac{N_A}{N_{dof_{ref}}}, \tag{4.4}$$

where $N_{dof_{ref}} = N_{ref} \times n_v$ is the number of DoFs associated with the reference uniform grid.

4.2 The G-Scheme performance

The G-Scheme performance can be evaluated in terms of computational time (CPU time) required to perform the model reduction and integration. As explained by Valorani [31], the most important contributions in the model reduction process are:

- the time to evaluate the Jacobian matrix of the vector field CPU_{Jac};
- the time to diagonalize the Jacobian matrix CPU_{Diag};
- the time to calculate the basis vectors CPU_{Basis};
- the time associated with the tail and head evaluation, CPU_{Tail} and CPU_{Head}, respectively.

By adding the time to integrate the set of non-stiff DEs produced by the G-Scheme (the active equations), the total CPU time is given by the following relation:

$$\mathrm{CPU}_{tot} \approx \mathrm{CPU}_{Jac} + \mathrm{CPU}_{Diag} + \mathrm{CPU}_{Basis} + \mathrm{CPU}_{Head} + \mathrm{CPU}_{Tail} + \mathrm{CPU}_{Int} \tag{4.5}$$

The largest contribution is typically due to the basis vectors and Jacobian matrix evaluation. The workloads are proportional to the number of DoFs identified by N, given by the product between the number of dependent variables n_v and the number of total grid points N_T: in particular one has $\simeq N^{1.5-2.0}$ for the Jacobian matrix and $\simeq N^{2.5-3.0}$ for the basis vectors, Fig. (4.2).

The next section is dedicated to the description of several strategies studied and implemented by the author to try to reduce the workload associated with the Jacobian matrix and basis vectors evaluation.

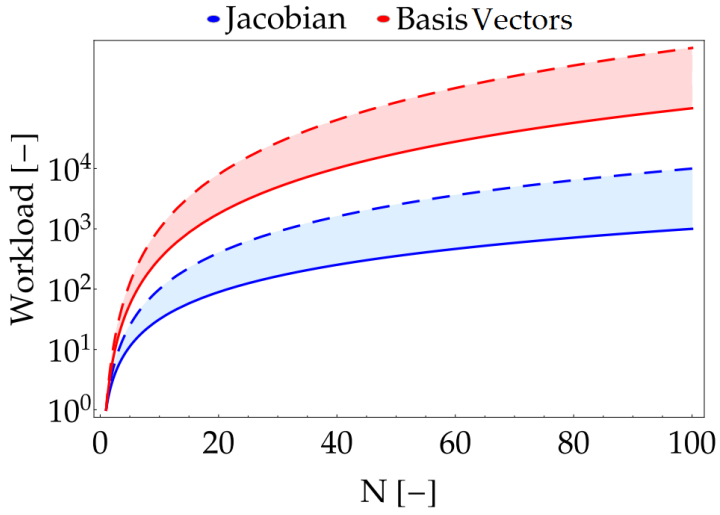


FIGURE 4.2: The workloads due to the Jacobian matrix and basis vectors evaluation are represented as functions of the problem dimension. The dashed lines are associated with the maximum exponents, while the continuous lines with the minimum exponents. The number of DoFs (N) is given by the product of the number of dependent variables (n_v) and the number of total grid points (N_T).

4.3 Improvement of the G-Scheme performance

The possible solutions to improve the G-Scheme performance can be summarized in the following points, summarized in Fig. (4.3):

- looking at Fig. (4.2), it is clear that the reduction of the number of total grid points allows to decrease the computational time associated with the Jacobian matrix and basis vectors evaluation. This goal can be easily carried out through the mesh refinement made by WAMR: the coupling between WAMR and the G-Scheme (following the scheme (4.1)) represents a basic solution allowing to improve the G-Scheme performance;

- another smart strategy is represented by the reuse of the basis vectors. As explained in section (3.2), the calculation of the basis vectors can be avoided if the effects of the system nonlinearities are not important;

- OS techniques can be used to build the Jacobian matrix *locally*. In this case it has dimensions equal to $n_v \times n_v$. The Jacobian matrix must be calculated at each grid point. Deep details of this strategy are explained in section (4.3.1);

- further improvements of the performance are obtained if WAMR and the reuse option are used at the same time or if the G-Scheme, used in OS techniques, is coupled with WAMR.

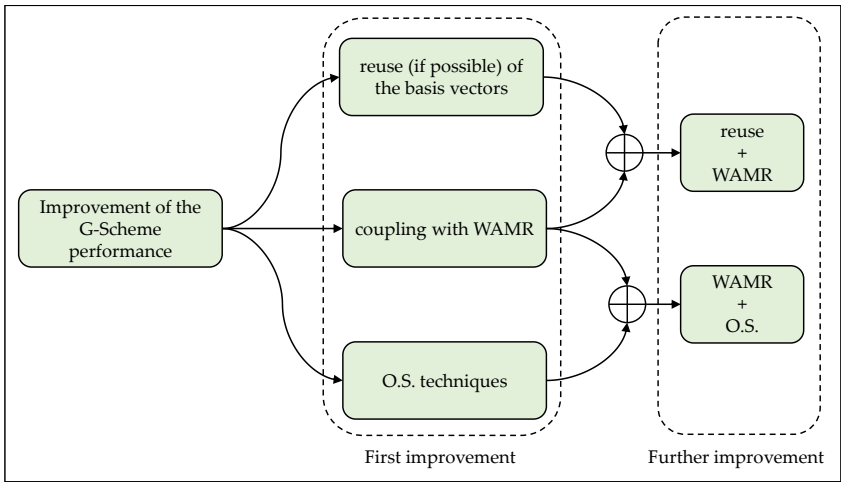


FIGURE 4.3: G-Scheme performance improvement.

4.3.1 Operator Splitting technique

A classical operator splitting technique shown by Sportisse [42] is employed to illustrate how the G-Scheme, coupled with WAMR, can be *locally* used. The idea is to decompose the system of PDEs (4.1) into simpler subproblems and treat them individually using specialized numerical algorithms. For this purpose, the system of PDEs (4.1) is rewritten as

$$\frac{\partial \vec{y}(x,t)}{\partial t} = L_1 \vec{y}(x,t) + L_2 \vec{y}(x,t), \tag{4.6}$$

where L_1 and L_2 are linear differential operators representing physical phenomena. In this context, L_1 is associated with the source term (chemistry) while L_2 with the diffusive term (diffusion). To describe how the splitting technique can be used with the G-Scheme, the scheme (4.1) is modified as shown in Fig. (4.4). The splitting technique is a first-order scheme, that consists in two steps, Step A and Step B, described in the following sections, respectively.

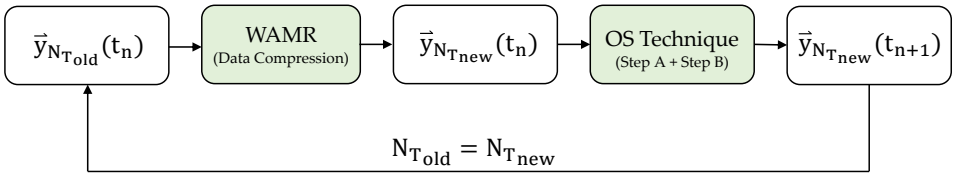


FIGURE 4.4: Coupling between WAMR and the OS technique. The OS technique consists in two parts, namely Step A and Step B.

Step A - Operator Splitting

Looking at Fig. (4.4), the output of WAMR is represented by the state vector

$$\vec{y}_{N_{T_{new}}}(t_n) = \{\vec{y}_1(t_n), \ldots, \vec{y}_{N_T}(t_n)\}, \tag{4.7}$$

where each component is a vector built on the i-th grid point, having dimensions $1 \times n_v$ and enclosing the n_v dependent variables of the problem,

$$\vec{y}_i(t_n) = \{y_i^1(t_n), \ldots, y_i^{n_v}(t_n)\}, \qquad i = 1, \ldots, N_T. \tag{4.8}$$

The structure of the vectors (4.8) is represented in Fig. (4.5).

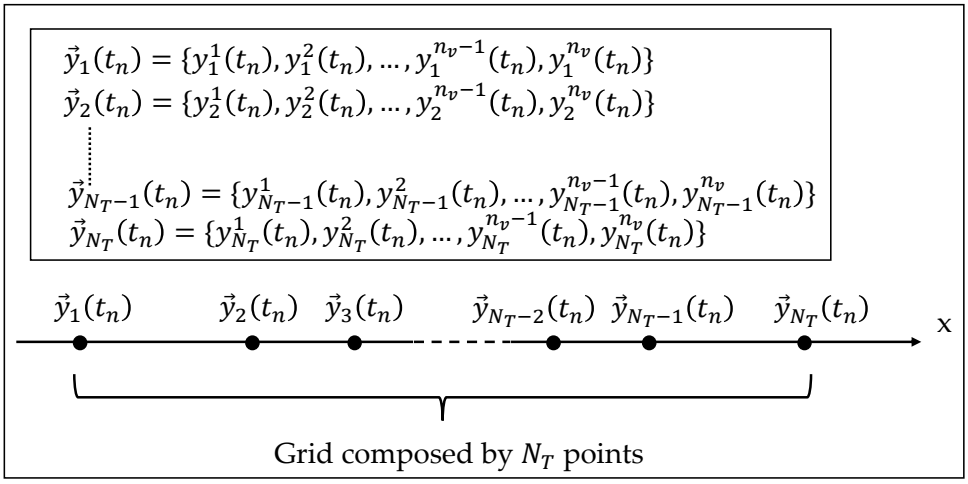


FIGURE 4.5: Structure of the vectors (4.8).

The Step A of the first order scheme is considered and it is described in Fig. (4.6). The chemistry is locally analysed and integrated with the G-Scheme: in this way the G-Scheme is able to build local Jacobian matrices - namely, at each grid point - having the minimum possible dimension, equal to $n_v \times n_v$. In this type of OS scheme the source term of the governing equations is firstly integrated between the time instants $t = t_n$ and $t = t_{n+1} = t_n + \Delta t$

$$\frac{\partial \vec{y}_i^*(t)}{\partial t} = L_1 \vec{y}_i^*(t) \quad \text{with} \quad \vec{y}_i^*(t_n) = \vec{y}_i(t_n) \quad \text{for} \quad t \in [t_n, t_{n+1}] \quad i = 1, \ldots, N_T, \tag{4.9}$$

where $\vec{y}_i^*(t)$ represents an intermediate state (between Step A and Step B, identified with the superscript *) evaluated at the i-th grid point. The solution at $t = t_{n+1}$ is

$$\vec{y}^*_{N_{T\,new}}(t_{n+1}) = \{\vec{y}_1^*(t_{n+1}), \ldots, \vec{y}_{N_T}^*(t_{n+1})\}. \tag{4.10}$$

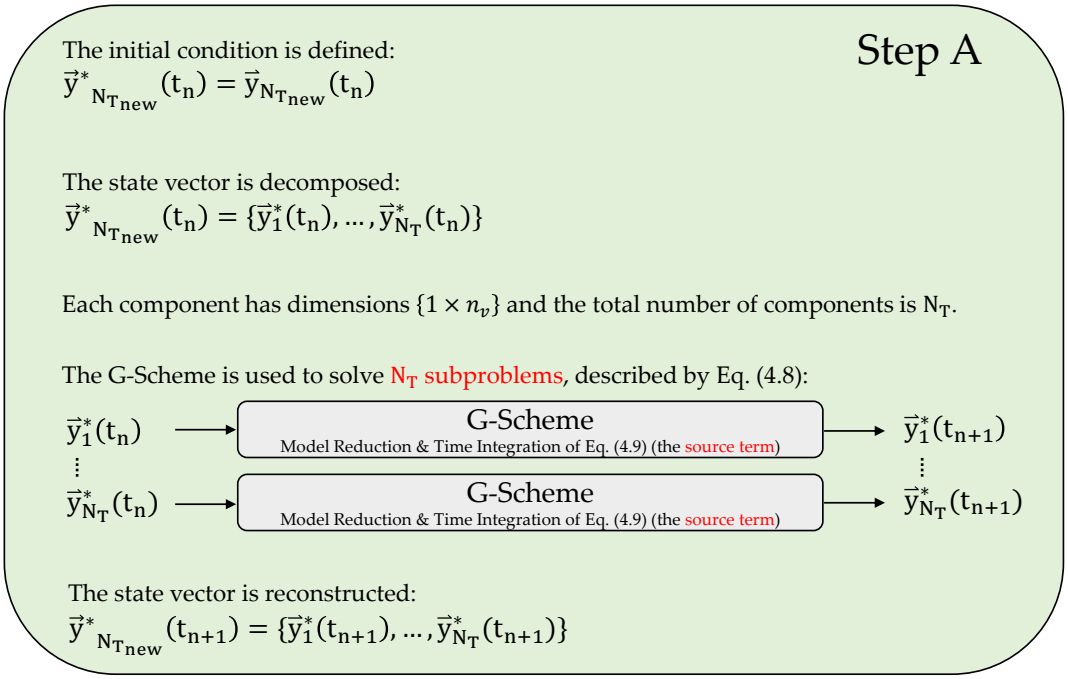


FIGURE 4.6: Operator Splitting - Step A.

Step B - Operator Splitting

The state vector solution of the Step A represents the initial condition for the integration of the diffusive term of the governing equations between the time instants $t = t_n$ and $t = t_{n+1} = t_n + \Delta t$:

$$\frac{\partial \vec{y}^{**}_{N_{T\,new}}(t)}{\partial t} = L_2 \vec{y}^{**}_{N_{T\,new}}(t) \quad \text{with} \quad \vec{y}^{**}_{N_{T\,new}}(t_n) = \vec{y}^{*}_{N_{T\,new}}(t_{n+1}) \quad \text{on} \quad t \in [t_n, t_{n+1}]. \tag{4.11}$$

After the time integration of Eq. (4.11) with a classical Runge-Kutta method characterized by an order compatible with the order of the operator splitting technique, the solution of the scheme is found,

$$\vec{y}^{**}_{N_{T\,new}}(t_{n+1}) = \{\vec{y}^{**}_1(t_{n+1}), \dots, \vec{y}^{**}_{N_T}(t_{n+1})\} = \vec{y}_{N_{T\,new}}(t_{n+1}). \tag{4.12}$$

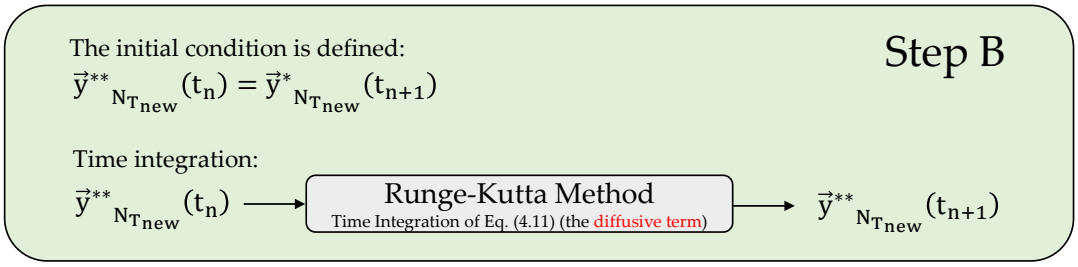


FIGURE 4.7: Operator Splitting - Step B.

Remarks

As explained by Sportisse [42], such scheme is a first-order scheme with respect to the splitting time step Δt. For instance, the local error for the A-B splitting described above is given by

$$le = (exp(B\Delta t))exp(A\Delta t)) - exp((A+B)\Delta t))\vec{y}_{N_{T_{new}}}(t_n) \tag{4.13}$$

The usual study of this error is performed by asymptotic expansion ald leads straightforwardly to

$$le = \frac{BA - AB}{2}\Delta t^2 \vec{y}_{N_{T_{new}}}(t_n) + O(\Delta t^3). \tag{4.14}$$

The global error is then a first-order error with respect to Δt. Note that, in order to improve the accuracy, the previous splitting scheme could be symmetrize. This would lead to a second-order scheme, proposed by Strang [43]. The use of second-order schemes is not the goal of the project.

CHAPTER 5

A Reaction-Diffusion Model

As a typical simple reaction-diffusion system exhibiting a rich dynamic structure, the Elezgaray-Arneodo model is considered. This model is able to mimic the sustained stationary and periodically oscillating "front structures" observed in an experiment conducted in the Couette flow reactor [10, 11]. It is described by two PDEs, where the parameters can be properly set to study different system behaviours.

The first goal of the chapter is to test the good implementation of WAMR. This is done by studying the steady state solutions of the system, obtained through a standard time integrator (DVODE): an error analysis is performed with the aim to check if the adaptive steady state solutions are at least accurate as prescribed by the wavelet threshold parameter. Different values of the wavelet order are also taken into account.

The governing equations parameters are set to study the periodic bursting behaviour of the system. The time evolution of the Arneodo variables is obtained over one limit cycle using adaptive and uniform grids, and the G-Scheme as time integrator. The main goal is to show that the coupling between WAMR and the G-Scheme (as proposed in Fig. (4.1)) is able to generate time varying solutions that well capture the reference ones. Furthermore, a large improvement of the performance evaluated in terms of computational time is expected using the solver WAMR/G-Scheme with respect to the cases where the G-Scheme is used on uniform grids. To decrease the computational time associated with the basis vectors evaluation, the reuse technique (section (3.2)) is also employed. The periodic bursting behaviour of the Arneodo model was also studied by Grenga [16] in pWAMR, with the aim to check if the solution obtained using the G-Scheme was sufficiently close to that one obtained with a standard Runge-Kutta-Fehlberg (RKF45) time integrator. The main differences between Grenga's approach and that one proposed here can be summarized in the following points: (i) in the present thesis the non parallel version of the adaptive algorithm based on the wavelet transform is tested and used, (ii) a detailed performance evaluation is proposed and (iii) the basis vectors reuse technique is employed to improve the performance and check if accurate solutions are obtained with respect to the reference ones.

The third part of the chapter is dedicated to a parametric study of the performance of the solver WAMR/G-Scheme, made by changing the relative wavelet threshold parameter and, correspondingly, the G-Scheme relative tolerances.

In the last part of the chapter the comparison between the AIM approach described in chapter (3) and suggested by Akram and Raman [2] is shown. The importance to have a time varying tail dimension, as proposed in the G-Scheme theory, is strongly stressed and proved.

5.1 Arneodo-Elezgaray reaction-diffusion model

The Arneodo-Elezgaray reaction-diffusion model is considered [10]. It is described by two coupled nonlinear PDEs (5.1) ($n_{eq} = 2$), where the reaction term is a two-variable Van der Pol-like system, while the diffusion term is given by a Fick law [11]. The dependent variables $u(x,t)$ and $v(x,t)$ (with $x \in [x_0, x_1] = [0,1]$ and $t \in [t_0, t_f) = [0, \infty)$) represent the concentrations of two chemical species with an isothermal explosive kinetics. Setting properly the equations parameters α, β and D, different system behaviours can be studied. In particular, the parameters α and β are taken constant for all cases and equal to $\alpha = \beta = 10^{-2}$, while the coefficient D must be changed to obtain the steady state and the periodic bursting behaviour of the system. In particular, it is equal to $D = 3.300 \times 10^{-2}$ and $D = 3.224 \times 10^{-2}$, respectively. The PDEs are solved imposing symmetric Dirichlet boundary conditions, namely $u(x_0,t) = u(x_1,t) = -2$ and $v(x_0,t) = v(x_1,t) = -4$.

$$\begin{cases} \frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + \beta^{-1}(v^2 - (u^2 + u^3)) \\ \frac{\partial v}{\partial t} = D\frac{\partial^2 v}{\partial x^2} - u + \alpha \end{cases} \tag{5.1}$$

5.2 Study of the steady state solutions using WAMR

The scheme represented in Fig. (4.1) is proposed to solve the governing equations (5.1). The steady state of the system is reached using DVODE as time integrator, with absolute and relative tolerances equal to $a_{tol} = 10^{-12}$ and $r_{tol} = 10^{-10}$, respectively. Four tests are performed using WAMR to generate adaptive grids, varying the relative value of the wavelet threshold parameter of one order of magnitude from $\varepsilon_r = 10^{-1}$ to $\varepsilon_r = 10^{-4}$, while the absolute value, ε_a, is equal to 10^{-8} in all cases. Fourth-order wavelets are used in all the tests where WAMR is employed.

The first step is represented by finding the adaptive steady state solutions, associated with the four values of the relative wavelet threshold parameters. Starting from the parabolic initial conditions represented in Fig. (5.3), the PDEs are integrated in time. The integration time step Δt is assigned knowing the minimum grid size (Δx_{min}) (*i.e.* the maximum resolution level) through a CFL-like condition, as Eq. (5.2) shows: in these tests the CFL is chosen equal to $K = 0.9$. The convergence to the steady state is checked looking at the root mean square (RMS) of the system RHS. Its limit value is set lower than the minimum relative wavelet threshold parameter by 4 orders of magnitude, namely 10^{-8}, as Eq. (5.3) shows. To check if the four adaptive steady state solutions are accurate in space as prescribed by the wavelet threshold parameters, four reference steady state solutions built on uniform grids are required. Each uniform grid is characterized by a resolution in space (Δx) given by the minimal grid spacing reached in the corresponding test where WAMR is used to perform adaptive mesh refinement (Δx_{min}). The procedure to build the reference and adaptive steady state solutions is summarized in Figs. (5.1): the blue block, representing a modified version of the scheme (4.1) (to which reference is made for the adopted symbols),

describes the basic coupling between WAMR and DVODE, while the red block is associated with the procedure to build the reference solutions. The four simulations where WAMR is adopted differ only in the value of the relative wavelet threshold parameter (ε_r), as indicated in Fig. (5.2): the figure stresses the fact that to know the spacing characterizing the uniform grids (Δx), the simulations where WAMR is adopted need to be done first.

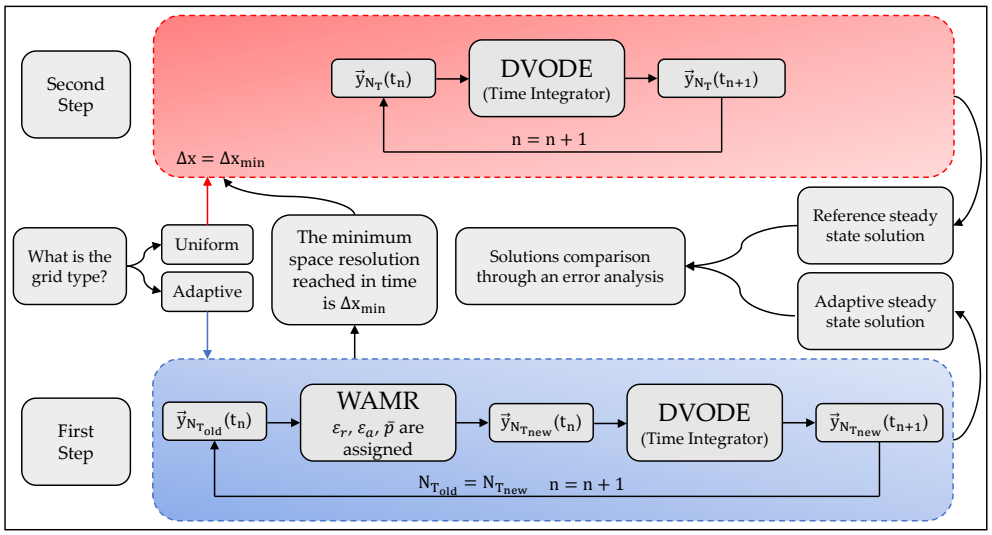


FIGURE 5.1: Procedure to build and compare the reference and adaptive steady state solutions. The scheme represented in the blue block is similar to the scheme (4.1), but now DVODE is used as time integrator. The scheme represented in the red block is a simplified version of the scheme where WAMR is used: the number of mesh points do not change. The minimum spacing reached in the simulations where WAMR is adopted are used to build the uniform grids for the reference tests.

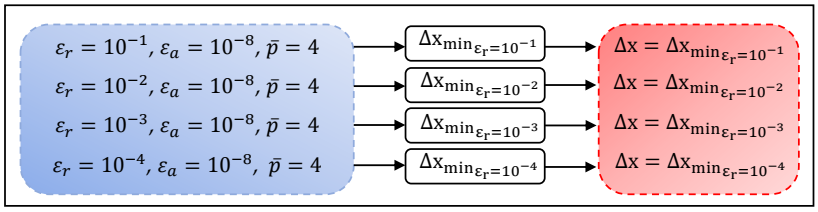


FIGURE 5.2: Δx assignment for the reference tests, for which uniform grids are used. The colours of the blocks correspond to those ones associated with Fig. (5.1).

Table (5.1) summarizes the results of the four WAMR/DVODE simulations in terms of total number of grid points (N_T), maximum resolution level (J) and minimum grid size (Δx_{min}) characterizing the steady state solutions: as expected, a decrease of the relative wavelet threshold parameter corresponds to an upward trend of the number of grid points and maximum resolution level. Indeed, higher resolution levels have to be added in order to obtain more accurate representations in space of the solution. Examples of reference steady state solutions are represented with continuous lines in Fig. (5.4), where also the initial conditions are shown with dashed lines:

these steady state solutions are obtained for the case of uniform grid characterized by a grid spacing equal to the minimum one reached in the WAMR/DVODE test where $\varepsilon_r = 10^{-3}$, namely $\Delta x = 2.041 \times 10^{-2}$, Tab. (5.1).

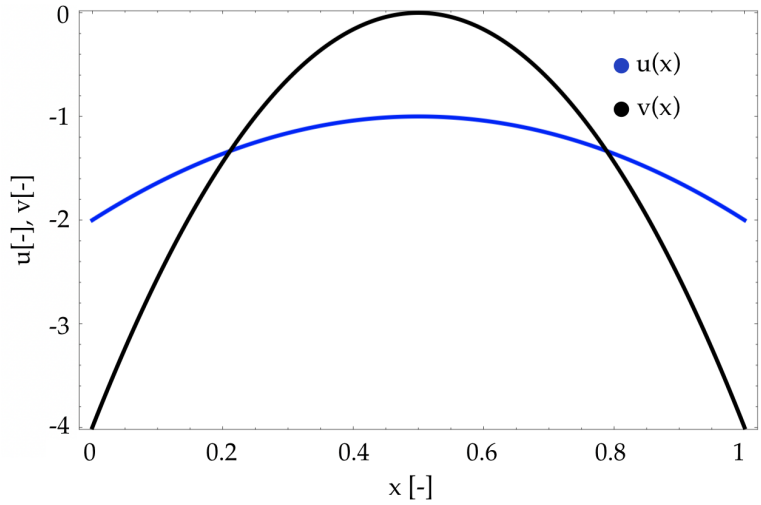


FIGURE 5.3: Arneodo model - parabolic initial conditions $u(x, t_0) = -4x^2 + 4x - 2$ and $v(x, t_0) = -16x^2 + 16x - 4$.

$$\mathrm{D} \times \frac{\Delta \mathrm{t}}{\Delta x_{min}^2} \leq K = 0.9 \;\rightarrow \Delta \mathrm{t} \leq K \times \frac{\Delta x_{min}^2}{\mathrm{D}} \tag{5.2}$$

$$\mathrm{RMS}_{\mathrm{lim}} = \sqrt{\frac{\sum_{i=1}^{n_{eq}} \mathrm{RHS}^2(i)}{n_{eq}}} = 10^{-8} \tag{5.3}$$

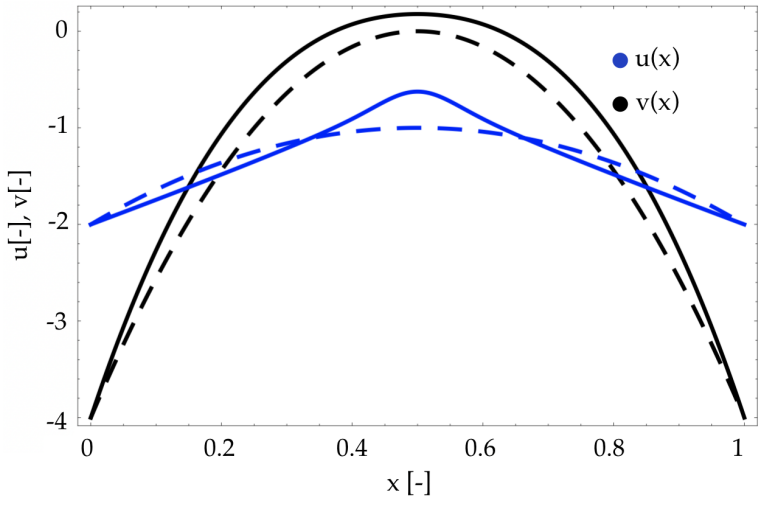


FIGURE 5.4: Arneodo model - steady state solutions. The continuous lines represent the reference steady state solutions: these solutions are obtained for the case of uniform grid characterized by a Δx associated with the WAMR/DVODE test where $\varepsilon_r = 10^{-3}$, namely $\Delta x = 2.041 \times 10^{-2}$, Tab. (5.1). The dashed lines are the parabolic initial conditions.

ε_r	J	N_T	Δx_{min}
10^{-1}	1	33	3.030×10^{-2}
10^{-2}	2	41	2.439×10^{-2}
10^{-3}	3	49	2.041×10^{-2}
10^{-4}	4	69	1.449×10^{-2}

TABLE 5.1: Relative wavelet threshold parameter, maximum resolution level, total number of grid points and minimum grid spacing with respect to the threshold parameters - the wavelet order is $\bar{p} = 4$.

To check if the maximum error introduced by the SWR of the steady state solutions is at most of the same order of magnitude as the threshold parameters, the L_∞-norm is chosen as error evaluation tool. It is evaluated as shown by Eqs. (5.4)-(5.5): the reference solutions are indicated with u_{ss}^{ref} and v_{ss}^{ref}, while the adaptive solutions are u_{ss}^{ε} and v_{ss}^{ε}. Note that the L_∞-norm is only evaluated at the essential points (indicated with the symbol x_{ess} in Eqs. (5.4)-(5.5)) given by the wavelet analysis, because of the neighbouring and non-essential points are not required to represent the solutions with the user prescribed accuracy in space.

$$\|u\|_\infty = \text{Max} \left| \frac{u_{ss}^{ref}(x_{ess}) - u_{ss}^{\varepsilon}(x_{ess})}{u_{ss}^{ref}(x_{ess})} \right| \tag{5.4}$$

$$\|v\|_\infty = \text{Max} \left| \frac{v_{ss}^{ref}(x_{ess}) - v_{ss}^{\varepsilon}(x_{ess})}{v_{ss}^{ref}(x_{ess})} \right| \tag{5.5}$$

The errors are expected to be of the same order of magnitude of the wavelet threshold parameters associated with the two variables u and v, namely ε_u and ε_v, respectively. They are calculated as shown in Eq. (2.67) and they are compared to the errors evaluated in terms of L_∞-norms in Tab. (5.2). Note that the evaluation of ε_u and ε_v requires the knowledge of $\bar{f}$, as indicated in Eq. (2.67): $\bar{f}$ is here represented by $\bar{u}$ and $\bar{v}$, calculated at the steady state condition. In particular, one has $\bar{u} = |\max(u(x))| = 6 \times 10^{-1}$ and $\bar{v} = |\max(v(x))| = 10^{-1}$.

ε_r	ε_a	ε_v	$\|v\|_\infty$	ε_u	$\|u\|_\infty$
10^{-1}	10^{-8}	$\approx 10^{-2}$	1×10^{-4}	$\approx 6 \times 10^{-2}$	1.5×10^{-3}
10^{-2}	10^{-8}	$\approx 10^{-3}$	4×10^{-5}	$\approx 6 \times 10^{-3}$	2×10^{-4}
10^{-3}	10^{-8}	$\approx 10^{-4}$	1.5×10^{-5}	$\approx 6 \times 10^{-4}$	1.5×10^{-4}
10^{-4}	10^{-8}	$\approx 10^{-5}$	1×10^{-5}	$\approx 6 \times 10^{-5}$	1×10^{-5}

TABLE 5.2: The first two columns represent the assigned relative and absolute wavelet threshold parameters, the third and fifth columns represent the wavelet threshold parameters evaluated as indicated by Eq. (2.67), while the fourth and last columns are the errors, calculated in terms of L_∞-norms.

The errors are lower than the estimations given by ε_u and ε_v. This is due to the presence of the neighbouring and non-essential grid points in the SWR of any function adapted by WAMR: these additional points make the steady state solution more accurate than expected. Figure (5.5) shows that the slope of log-log plots of errors versus the number of points is approximatively -4, as predicted by the theory: it can be proved that the following relation holds [55],

$$||f - f_\varepsilon^l||_\infty \leq CN^{-\bar{p}/d}, \tag{5.6}$$

where f_ε^l is the SWR of a given function f (in this problem it is u or v) at the scale level l, $\bar{p}$ is the wavelet order and d is the dimension of the problem (it is equal to 1 for the Arneodo model), while C is a constant depending on f, d, $\bar{p}$ and mildly on the threshold value ε [55]. The dashed lines of Fig. (5.5) represent the theoretical slope of the curves: the slope is simply given by the ratio between the wavelet order and the dimension of the problem.

The tests are repeated using sixth-order wavelets, and the results of the four WAMR/DVODE simulations are summarized in Tab. (5.3) in terms of number of grid points, maximum resolution level and minimum grid size characterizing the steady state solutions: a comparison with Tab. (5.1) confirms that by increasing the order of the Lagrange interpolating polynomials through the wavelet order, the maximum resolution levels are lower than the case for which $\bar{p} = 4$. This means that, for the same relative and absolute wavelet threshold parameters, a lower number of grid points is required in the case of $\bar{p} = 6$.

ε_r	J	N_T	Δx_{min}
10^{-1}	1	33	3.030×10^{-2}
10^{-2}	1	33	3.030×10^{-2}
10^{-3}	2	45	2.222×10^{-2}
10^{-4}	3	57	1.754×10^{-2}

TABLE 5.3: Relative wavelet threshold parameter, maximum resolution level, total number of grid points and minimum grid spacing with respect to the threshold parameters - the wavelet order is $\bar{p} = 6$.

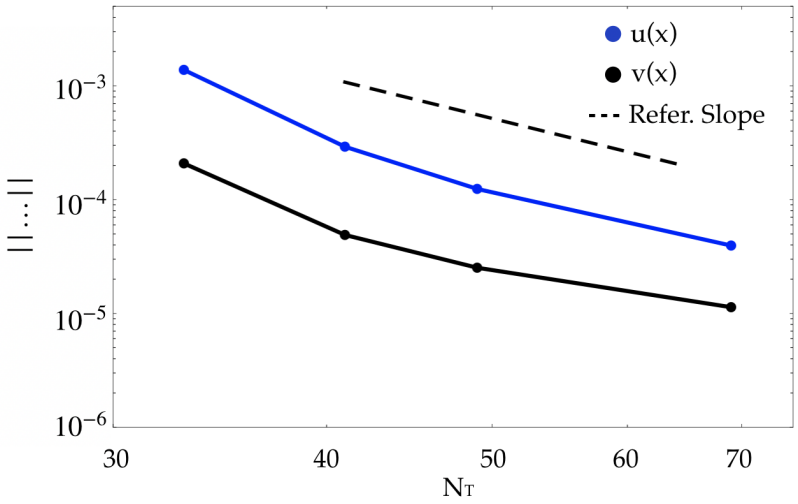
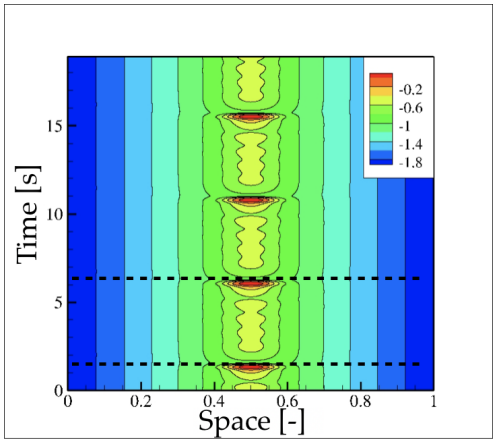


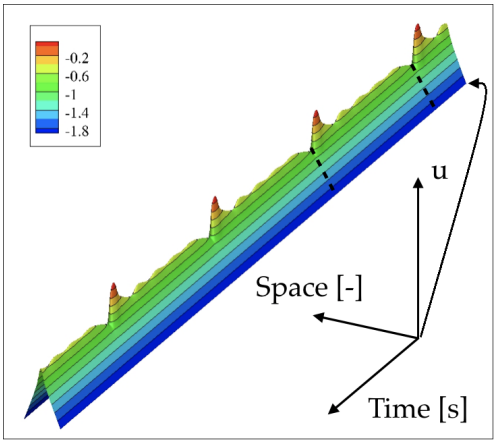
FIGURE 5.5: Arneodo model - L_∞ norm with respect to the total number of grid points; the dashed line represents the theoretical slope of the curves, equal to $-\bar{p}/d = -4$ - the wavelet order is $\bar{p} = 4$.

5.3 Periodic bursting behaviour

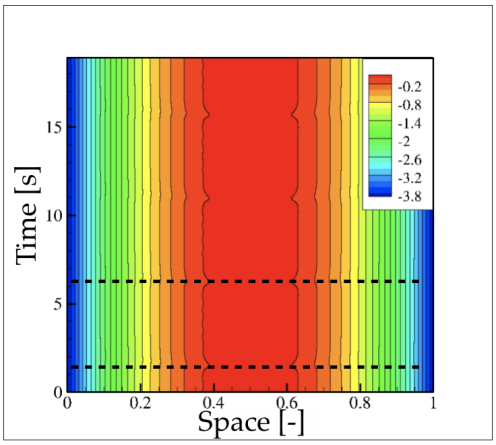
The periodic bursting behaviour of the Arneodo-Elezgaray model is obtained setting the parameters of Eqs. (5.1) as sec. (5.1) shows. With the sole purpose to display the limit cycles structure, the governing equations are integrated in time with DVODE (with absolute and relative tolerances equal to $a_{tol} = 10^{-12}$ and $r_{tol} = 10^{-10}$, respectively) on a uniform grid composed by 8193 points. The limit cycles structure is represented in Fig. (5.6), where the time variation of u and v is shown in two and three dimensions. A single limit cycle is described in a period equal to $T_{pb} = 4.73$ s. This time interval is considered for the next tests.



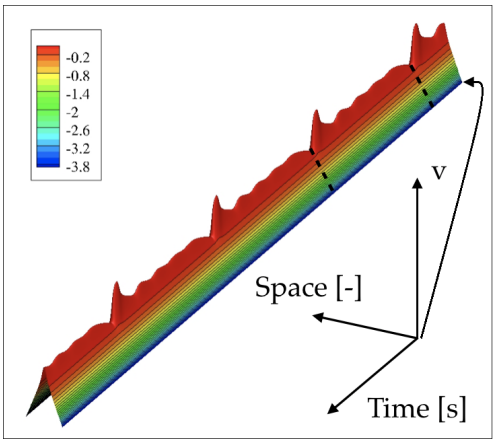
(a) Iso-contours of $u(x,t)$ - 2D.



(b) Iso-contours of $u(x,t)$ - 3D.



(c) Iso-contours of $v(x,t)$ - 2D.



(d) Iso-contours of $v(x,t)$ - 3D.

FIGURE 5.6: Limit cycles structure of the reaction-diffusion system (5.1) for the model parameters $\alpha = 10^{-2}$, $\beta = 10^{-2}$ and $D = 3.224 \times 10^{-2}$: (a) $u(x,t)$ variation in the time-space plan, (b) $u(x,t)$ variation in three-dimensional view, (c) $v(x,t)$ variation in the time-space plan, (d) $v(x,t)$ variation in three-dimensional view.

5.3.1 Time varying solutions comparison using WAMR and the G-Scheme with respect to reference solutions

Following the scheme (4.1), the coupling between WAMR and the G-Scheme is proposed to study the periodic bursting behaviour of the Arneodo model. The goal is twofold:

1. to check if the time varying solutions of the system (5.1) are sufficiently accurate in space and time with respect to reference time varying solutions;

2. to study the performance evaluated in terms of computational time as described by Eq. (4.5).

Five tests are made over one limit cycle ($T_{pb} = 4.73$ s) and their main features are summarized in Tab. (5.4): the time varying solutions are built on uniform or adaptive grids (produced by WAMR), using DVODE (for the reference test) or the G-Scheme as time integrators. Note that the reuse of the basis vectors is also tested when the G-Scheme is used, given that better performance is expected. The tests features and goals are also summarized in Fig. (5.7).

Test Name	Grid Type	Time Integrator	Reuse Basis
Test R - Ref.	Uniform	DVODE	-
Test A	Uniform	G-Scheme	Deactivated
Test B	Uniform	G-Scheme	Activated
Test C	Adaptive (WAMR)	G-Scheme	Deactivated
Test D	Adaptive (WAMR)	G-Scheme	Activated

TABLE 5.4: Tests structure: the simulations are made over one limit cycle described in a period equal to $T_{pb} = 4.73$ s, using uniform or adaptive grids (generated by WAMR) and the G-Scheme or DVODE as time integrators. Whenever the G-Scheme is adopted, the reuse of the basis vectors is activated (Test B and Test D) or not (Test A and Test C).

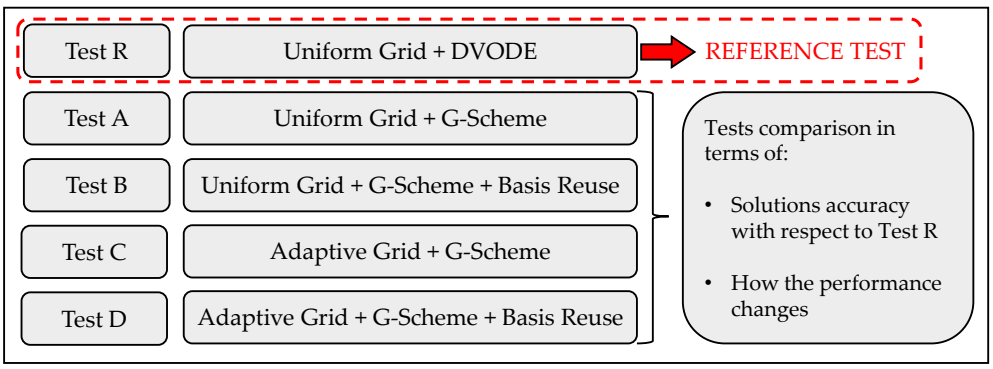


FIGURE 5.7: Tests structure and goals.

Simulations parameters setting

The uniform grids are built looking at the maximum resolution and the minimum grid size reached in the tests where WAMR is used. The maximum resolution level and the minimum grid size reached in the Test C and Test D are the same regardless of the reuse or not of the basis vectors. In particular, one has $J = 5$ and $\Delta x_{min} = 1.9553125 \times 10^{-3}$. Therefore, the

corresponding number of grid points of the uniform grids associated with Test R, Test A and Test B are $N_T = 513$. The DVODE absolute and relative tolerances are set $atol = 10^{-12}$ and $rtol = 10^{-10}$, respectively, while the G-Scheme absolute and relative tolerances are $atol_{head} = atol_{tail} = 10^{-8}$ and $rtol_{head} = rtol_{tail} = 10^{-3}$, respectively. The relative value of the wavelet threshold parameter is $\varepsilon_r = 10^{-3}$ (with a corresponding absolute value $\varepsilon_a = 10^{-8}$) and fourth order wavelet functions are adopted.

The reference solution - Test R

The reference solution is built. The time evolution of the dependent variables at $x = 0.5$ is shown in Fig. (5.8), while the phase plan is represented in Fig. (5.9). The red dot of Fig. (5.9) is the initial condition, while the red arrows show the direction to follow along the limit cycle.

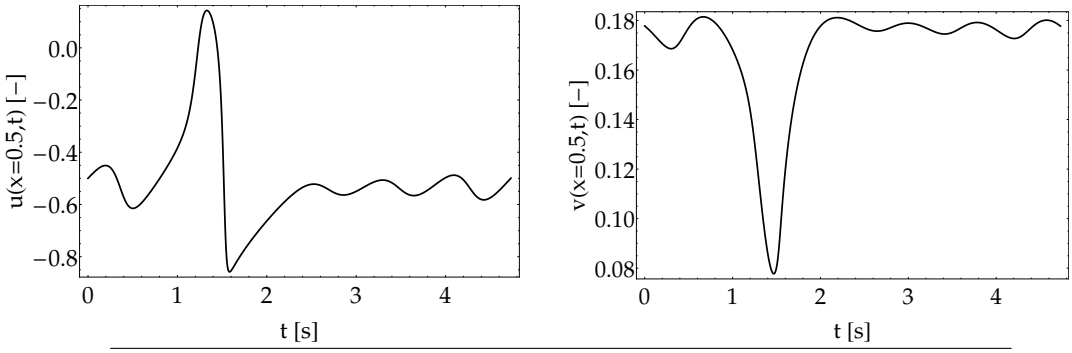


FIGURE 5.8: Time evolution of the dependent variables $u(x,t)$ and $v(x,t)$ at $x = 0.5$. The time integration is performed with DVODE.

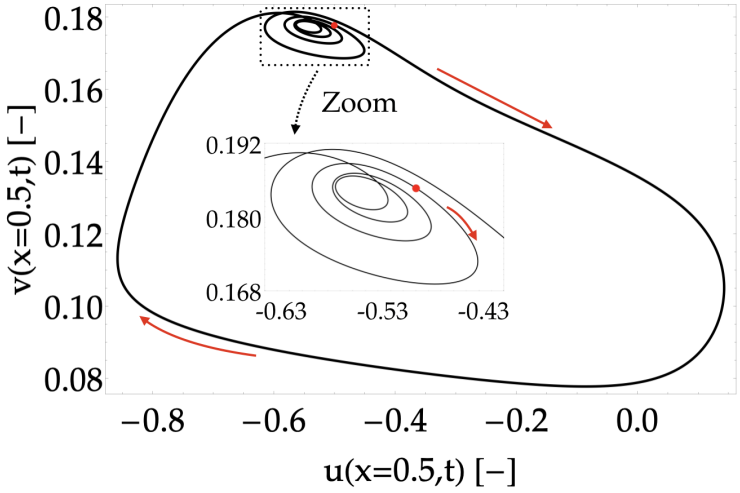


FIGURE 5.9: Arneodo model - periodic bursting behaviour - limit cycle at $x = 0.5$. The red dot is the initial condition.

Test A and Test B

Test A and Test B are considered. The time evolution of the two dependent variables $u(\bar{x},t)$ and $v(\bar{x},t)$ at $\bar{x} = 0.5$ is compared to the reference trends (Test R) in Fig. (5.10): the G-Scheme capability to produce excellent approximations of the reference solutions can be appreciated. The comparison is also proposed choosing as error evaluation tools the following ratios:

$$\mathrm{Log}_{10}\left|\frac{u^{ref}(\bar{x},t)-u^{\#}(\bar{x},t)}{u^{ref}(\bar{x},t)}\right|, \quad \mathrm{Log}_{10}\left|\frac{v^{ref}(\bar{x},t)-v^{\#}(\bar{x},t)}{v^{ref}(\bar{x},t)}\right|, \tag{5.7}$$

where $u^{ref}(\bar{x},t)$ and $v^{ref}(\bar{x},t)$ are the reference trends at $\bar{x} = 0.5$, while the superscript # refer to the other tests of Tab. (5.4). The results are shown in Fig. (5.11). As expected, the errors are typically lower if the reuse of the basis vectors were not considered (Test A). However, the advantage to reuse the basis becomes evident if the G-Scheme performance is evaluated. The results are summarized in Figs. (5.12)-(5.13). In both cases (Test A and Test B) the time associated with the basis vectors evaluation represents the largest contribution in the total computational time: it is 98.984 % for Test A and 95.829 % for Test B, while the percentage contribution of the other times is always below 3%. The basis evaluation time associated with Test A is 1397.523 s, while it is 145.284 s for the Test B and the ratio is approximatively equal to 10. This numerical result, coupled with the excellent accurate results of Fig. (5.10), underlines significantly the importance to reuse the basis vectors.

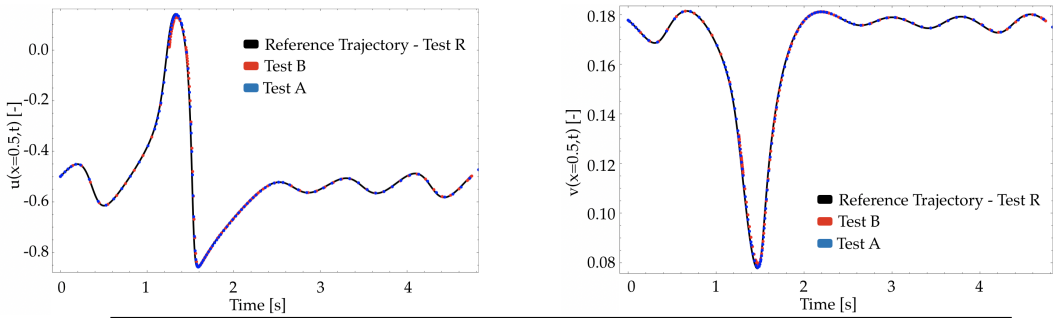


FIGURE 5.10: Time evolution of the dependent variables at $x = 0.5$. The solutions obtained with the G-Scheme - reusing or not the basis vectors - on uniform grids are compared to the reference solutions.

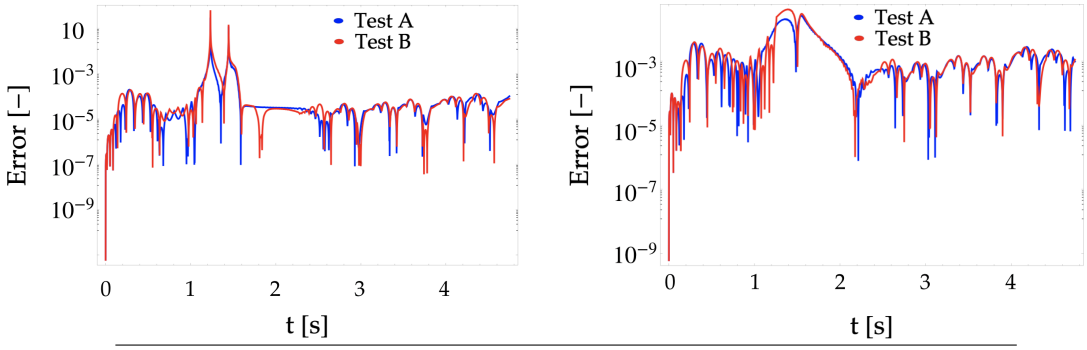


FIGURE 5.11: Time evolution of the errors defined in Eqs. (5.7).

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	9,837	1397,523	4,396	8,081E-03	6,917E-02	0

CPU Total Time [s]
1411,867

Percentage Reuse Basis Vectors
0

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	0,697	98,984	0,311	5,724E-04	4,899E-03	0

FIGURE 5.12: Performance evaluation for Test A in terms of computational times.

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
1,275E-02	0,978	145,284	5,182	1,168E-02	9,952E-02	0

CPU Total Time [s]
151,606

Percentage Reuse Basis Vectors
91,604

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
8,409E-03	0,645	95,829	3,418	7,703E-03	6,565E-02	0

FIGURE 5.13: Performance evaluation for Test B in terms of computational times.

Test C and Test D

As explained in section (4.2), the workloads associated with the basis and Jacobian matrix evaluation are strongly dependent on the number of DoFs, given by the product of the number of dependent variables ($n_v = 2$ for the Arneodo model) and the total number of grid points N_T. Therefore, it is expected to obtain better G-Scheme performance (namely lower workloads) if WAMR is used to make adaptive mesh refinement. This is done in Test C and Test D. Figure

(5.14) shows the variables time evolution for Test C and Test D, at $x = 0.5$. The excellent capability of the coupling between WAMR and the G-Scheme to produce solutions extremely near to the reference ones can be appreciated. The comparison with the reference solutions is also proposed through error evaluation tools proposed in Eq. (5.7), and the results are shown in Fig. (5.15). As expected, the errors are larger is the basis reuse is activated (Test D). The performance associated with Test C and Test D are represented in Figs. (5.16)-(5.17): one of the first things to note is the significant decrease of the total computational time, with respect to the cases of uniform grid (Test A and Test B): one has 6.142 s for Test C ($\simeq$ 230 times lower than Test A) and 3.571 s for Test D ($\simeq$ 42 times lower than Test B). The time requested to evaluated the basis vectors represents again the most important contribution (91.064% for Test C and 89.035% for Test D), while the percentages associated with other calculations are always below the limit value 7%.

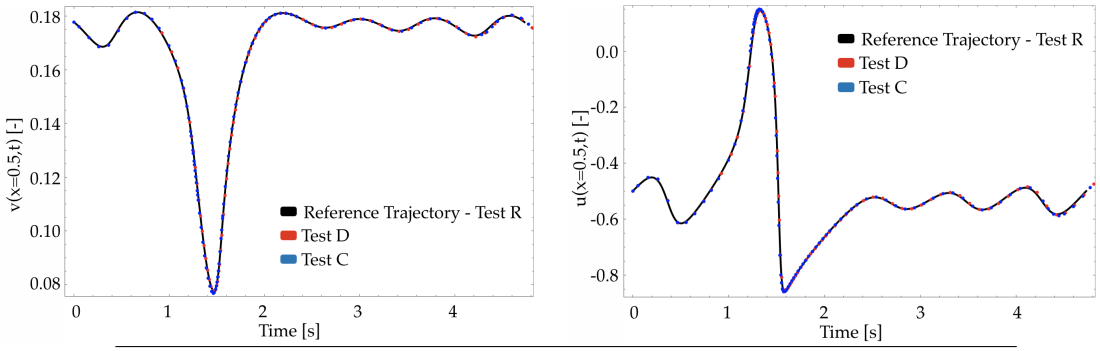


FIGURE 5.14: Time evolution of the dependent variables at $x = 0.5$. The solutions obtained with the G-Scheme - reusing or not the basis vectors - on adaptive grids (WAMR) are compared to the reference solutions.

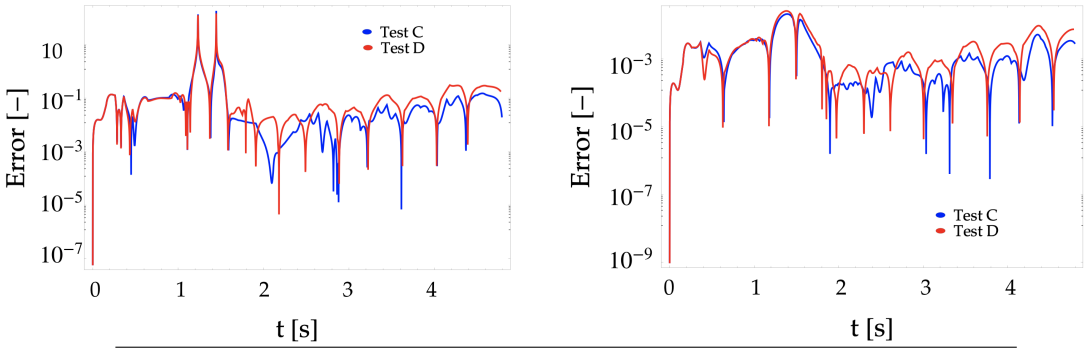


FIGURE 5.15: Time evolution of the errors defined in Eqs. (5.7).

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	0,347	5,593	0,101	9,679E-04	2,236E-02	7,715E-02

CPU Total Time [s]
6,142

Percentage Reuse Basis Vectors
0

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	5,648	91,064	1,652	1,576E-02	0,364	1,256

FIGURE 5.16: Performance evaluation for Test C in terms of computational times.

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
3,771E-03	0,216	3,179	7,479E-02	9,426E-04	1,931E-02	7,635E-02

CPU Total Time [s]
3,571

Percentage Reuse Basis Vectors
32,941

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0,106	6,059	89,035	2,095	2,640E-02	0,541	2,138

FIGURE 5.17: Performance evaluation for Test D in terms of computational times.

Tests comparison

Comparing Test B and Test D for which the reuse option is activated, the reuse percentage of the basis vectors is extremely different: it is 91.604% for Test B and 32.941% for Test D. This extreme difference can be explained considering that when WAMR is used (Test D) the grid dimension is modified in time: whenever a grid change occurs, the eigenproblem at the base of the G-Scheme needs to be mandatorily evaluated and the reuse of basis is not allowed. Figure (5.18) shows the

time evolution of the number of grid points for the Test B and Test D: the red points are associated with time instants where the basis vectors are effectively calculated.

As explained in section (3.2), the reuse strategy allows to have a temporal update of the basis vectors only when rotations of the basis are significant, namely when the effects of nonlinearities are relevant. Looking at the straight line associated with Test B of Fig. (5.18), the system nonlinearities seem to be particularly included in the time interval $t \in (1.0, 2.5)$ s. Note that if the system is characterized by a linear behaviour, the basis vectors shapes at different time instants do not change significantly. Figure (5.19) shows the eigenvalues time evolution in terms of order of magnitude for Test B. The active, slow and fast subspaces are clearly indicated, while the dashed orange line is associated with the time evolution of a selected eigenvalue, namely the sixth eigenvalue of the Jacobian matrix (the eigenvalues are ordered as shown in Eq. (3.1.1)). The Arneodo model is characterized by 4 null eigenvalues corresponding to the conserved modes of the G-Scheme: they are obtained because of the imposition of the Dirichlet boundary conditions. Figure (5.19) shows also eleven time instants whose values are explicitly indicated in Tab. (5.5). These are the selected time instants where the right eigenvectors of the Jacobian matrix are explicitly calculated for the sixth eigenvalue. It is expected to have a slight variation in time of the modes shapes if the selected eigenvalue has a quite constant trend in time. Figure (5.20) shows the modes shapes at the different time instants of Tab. (5.5). The expectations are clearly confirmed: the modes shapes do not change significantly when the system behaviour is quite linear, namely when the time is $t \geq t_9$, while for $t \in [t_1, t_8]$ the effects of nonlinearities are relevant. These effects are extremely important for $t \in (1.0, 2.5)$ s. Figure (5.21) shows the time evolution of u and v at $x = 0.5$, while Fig. (5.22) represents the limit cycle: the zone where the system has a nonlinear/linear behaviours are indicated with the red/blue colours, respectively.

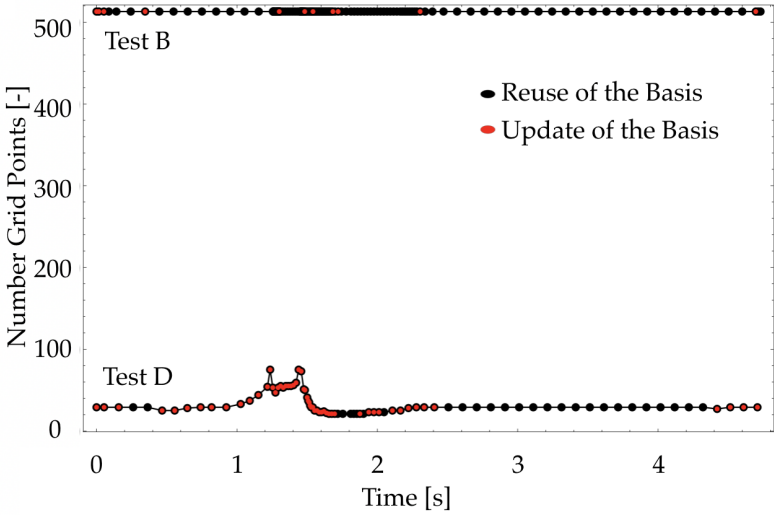


FIGURE 5.18: Time evolution of the number of grid points for Test B and Test D. The red dots are associated with updates of the basis vectors.

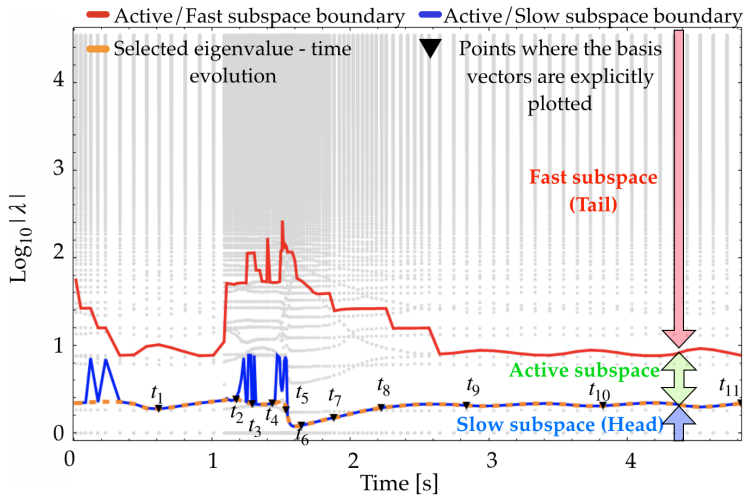


FIGURE 5.19: Eigenvalue time evolution in terms of order of magnitude. The red/blue lines are associated with the boundary between the fast/active and slow/active subspaces, respectively. The dashed orange line correspond to the time evolution of the sixth eigenvalue of the Jacobian matrix, according to the proposed eigenvalues arrangement of Eq. (3.1.1). Eleven time instants are indicated: at these points the right eigenvectors of the Jacobian matrix are plotted for the selected eigenvalue.

t_1 [s]	t_2 [s]	t_3 [s]	t_4 [s]	t_5 [s]	t_6 [s]	t_7 [s]	t_8 [s]	t_9 [s]	t_{10} [s]	t_{11} [s]
0.527	1.159	1.289	1.419	1.530	1.631	1.850	2.190	2.745	3.734	4.723

TABLE 5.5: Time instants where the eigenvectors of the Jacobian matrix are evaluated for the selected eigenvalue of Fig. (5.19).

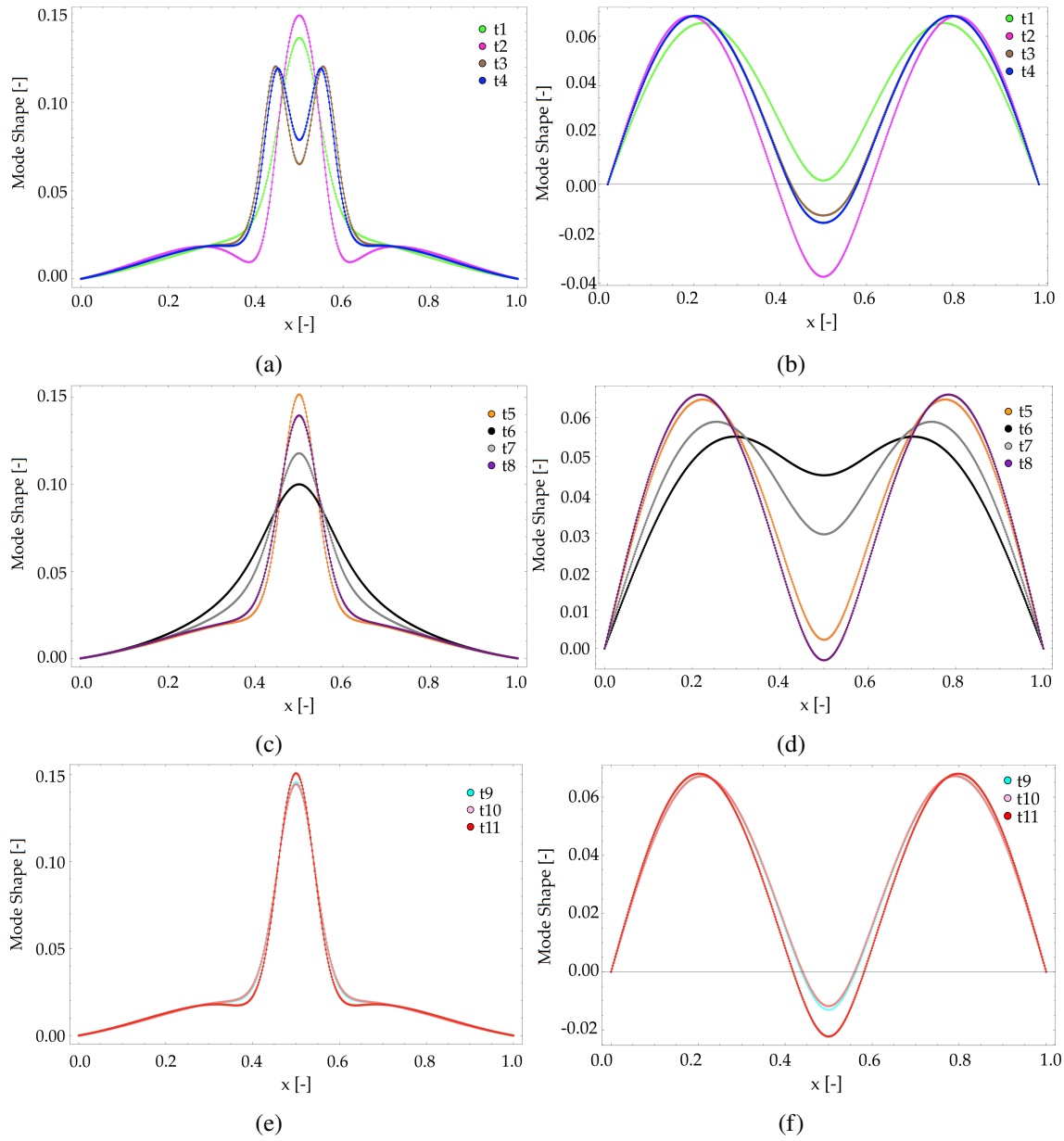
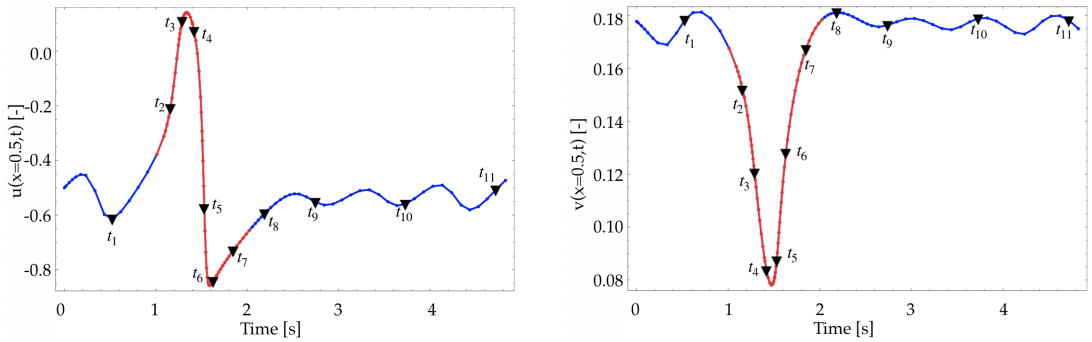


FIGURE 5.20: Modes shapes associated with the selected eigenvalue of Fig. (5.19). Different time instants are considered, whose values are shown in Tab. (5.5). The system behaviour is quite linear for the time instants equal to t_9, t_{10} and t_{11}, for which the modes shapes is quite unchanged.



(a) (b)

FIGURE 5.21: Time evolution of u (a) and v (b) at $x = 0.5$ for Test B. The red/blue lines are associated with the nonlinear/linear behaviour of the system, respectively.

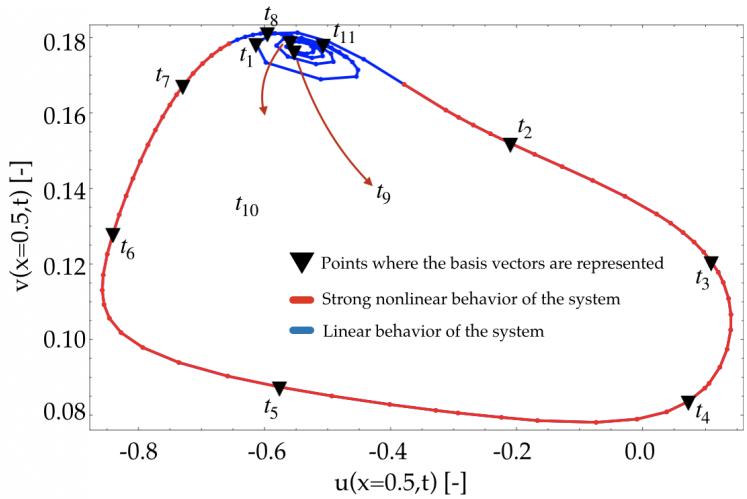


FIGURE 5.22: Limit cycle for Test B. The red/blue lines are associated with the nonlinear/linear behaviour of the system, respectively.

Figure (5.23) summarizes the results in terms of performance, for all the five tests. Two important things to note are that

- the total computational time increase if the G-Scheme is used on uniform grid without the reuse of the basis vectors;
- the best performance is obtained with the reference solver, namely 0.330 s.

Figure (5.24) shows the order of magnitude of the ratio between the total computational times associated with Test A, Test B, Test C and Test D, and the total computational time of the reference test, Test R. From the bar chart it is clear that the coupling between WAMR and the G-Scheme with the basis vectors reuse allows to extremely reduce the gap between the computational times: for Test D the gap is approximatively equal to 1 order of magnitude, while for Test A it is 3.63.

Figure (5.25) summarizes the results in terms of performance considering only the heaviest contribution, given by the basis vectors evaluation. It is clear that:

- regardless of the grid type (uniform or adaptive), the reuse of the basis vectors allows always to obtain better performance in terms of total computational time;
- when WAMR is used (Test C and Test D) the number of DoFs drops and the computation times are reduced with respect to the cases of uniform grids;
- the best performance are obtained using WAMR and the G-Scheme with the reuse of the basis vectors (Test D).

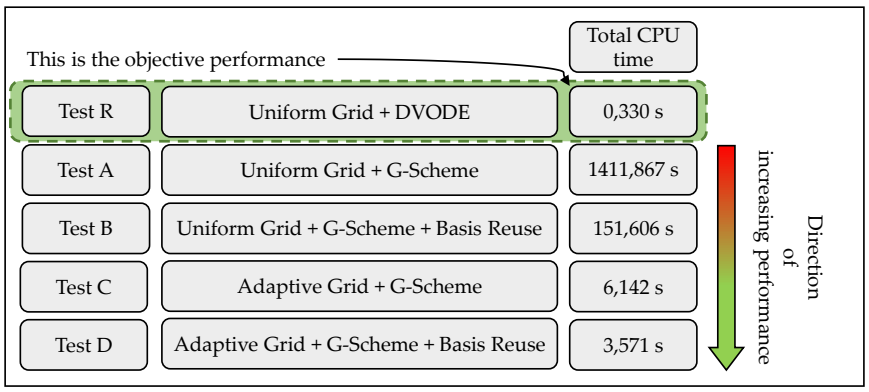


FIGURE 5.23: Total computational time variation from Test R to Test D. The objective performance is represented by the CPU time associated with Test R.

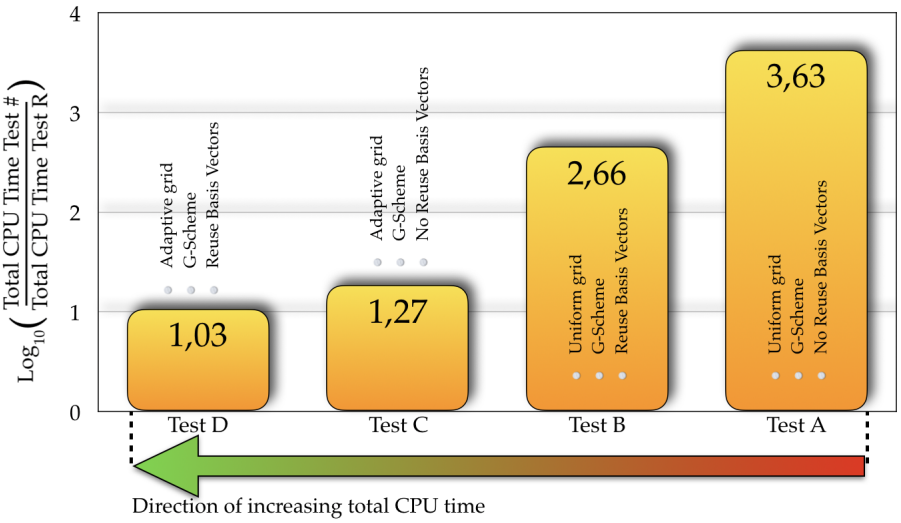


FIGURE 5.24: Order of magnitude of the ratio between the total computational time associated with Test A, Test B, Test C and Test D, and the total computational time associated with the reference test, namely Test R. Better performance are obtained moving from Test A to Test D.

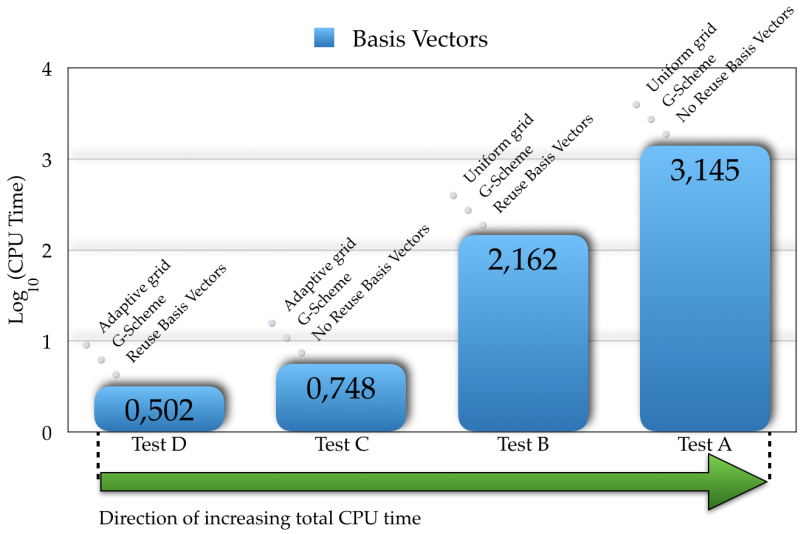


FIGURE 5.25: Performance comparison in terms of computational times. Note that only the most important contribution to the total computational time is considered. Better performance (lower computational times) are obtained using adaptive grids, the G-Scheme and the reuse of the basis vectors.

5.3.2 Parametric study with respect to the relative tolerances of WAMR and the G-Scheme

In the previous section it has been shown that excellent performance and accurate solutions are obtained for Test D, where WAMR is coupled with the G-Scheme and the reuse of the basis vectors is activated. The same test is now repeated with the aim to study the performance of the adaptive scheme (4.1) varying the relative wavelet threshold parameter of one order of magnitude from $\varepsilon_r = 10^{-3}$ to $\varepsilon_r = 10^{-6}$, as Fig. (5.26) shows. All the simulations are again made over one limit cycle described in a period equal to $T_{pb} = 4.73$ s, using WAMR, the G-Scheme and the reuse of the basis vectors. The relative and absolute tolerances of the G-Scheme are simultaneously modified with respect to the wavelet threshold parameters, as Tab. (5.6) shows. Note that in all tests the absolute tolerances for WAMR and the G-Scheme are the same and equal to 10^{-8}. What is expected is to observe worse performance if the relative tolerances (ε_r, $rtol_{head}$, $rtol_{tail}$) are lower because of the requirement of more accurate solutions. The logic of the tests and the expectations are summarized in Fig. (5.27): the choice of lower relative tolerances will mainly bring to larger dimension of the eigenproblem at the base of the G-Scheme. Focus is given on this aspect because, as just explained (section (5.3.1)), it is expected that the performance of the scheme (5.27) are mainly affected by the basis vectors evaluation: it will be shown that the workload associated with the mesh refinement performed by WAMR, the computation and the diagonalization of the Jacobian matrix, the integration of the PDEs (5.1), the evaluation of tail and head play minor roles.

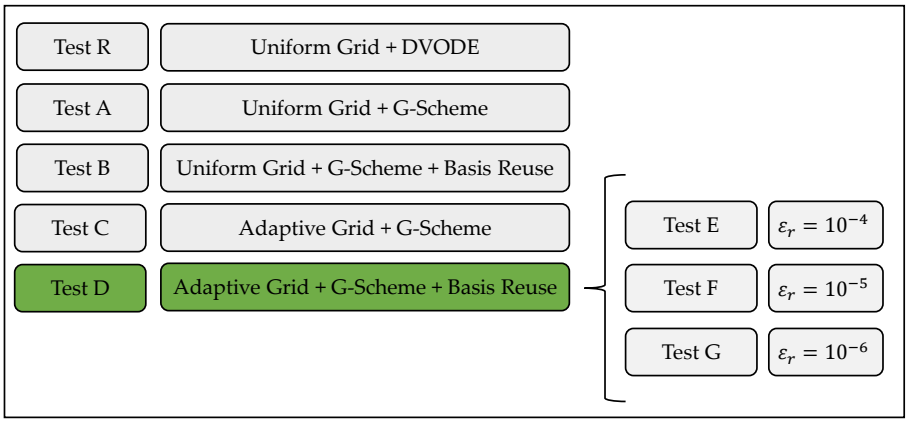


FIGURE 5.26: Tests structure: starting from the excellent results obtained for Test D, new simulations are performed with the aim to check how the performance and the solutions accuracy change with respect to the relative WAMR and G-Scheme tolerances.

Test Name	ε_r	ε_a	$rtol_{head}$	$rtol_{tail}$	$atol_{head}$	$atol_{tail}$
Test D	10^{-3}	10^{-8}	10^{-3}	10^{-3}	10^{-8}	10^{-8}
Test E	10^{-4}	10^{-8}	10^{-4}	10^{-4}	10^{-8}	10^{-8}
Test F	10^{-5}	10^{-8}	10^{-5}	10^{-5}	10^{-8}	10^{-8}
Test G	10^{-6}	10^{-8}	10^{-6}	10^{-6}	10^{-8}	10^{-8}

TABLE 5.6: Tests structure: the wavelet threshold parameters are simultaneously modified with respect to the G-Scheme relative tolerances. The G-Scheme and WAMR absolute tolerances are taken constant.

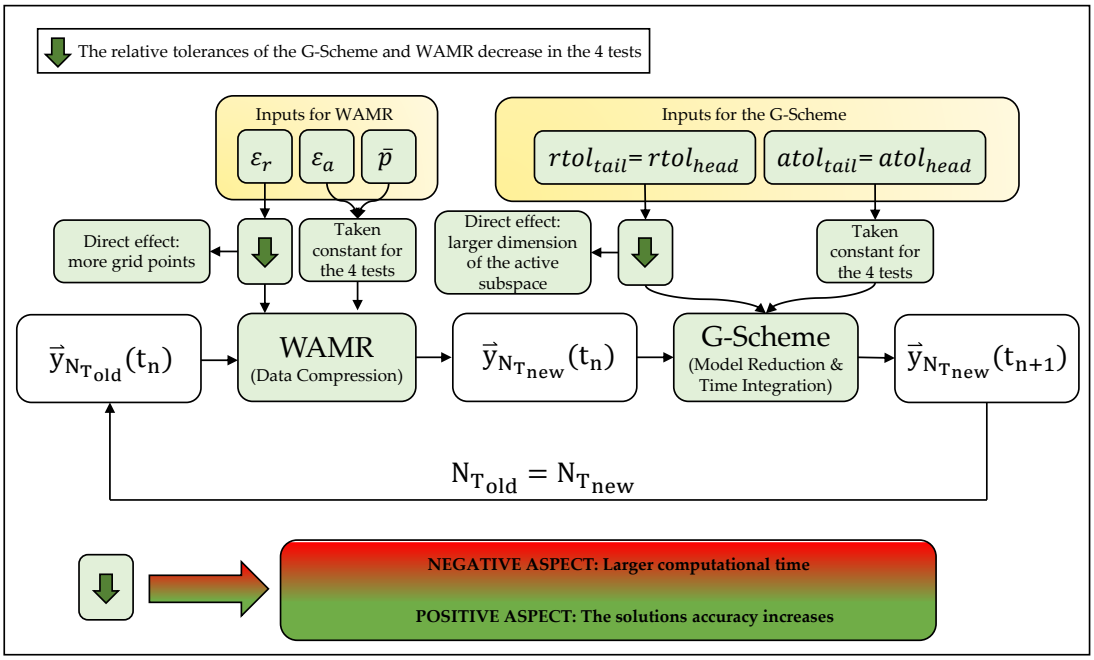


FIGURE 5.27: Decreasing the G-Scheme and WAMR relative tolerances it is expected to obtain better solutions in terms of accuracy with respect to the reference test, but also worse performance because of the larger dimension of the eigenproblem (mainly). This scheme is a modified version of the scheme (4.1) at which reference is made for the meaning of the unknown symbols.

To check how the solution accuracy varies moving from $\varepsilon_r = 10^{-3}$ ($rtol_{tail} = rtol_{head} = 10^{-3}$) to $\varepsilon_r = 10^{-6}$ ($rtol_{tail} = rtol_{head} = 10^{-6}$), the chosen error evaluation tool is simply given by:

$$\begin{aligned} &|u_{\varepsilon_r=10^{-3}}(\bar{x},t) - u_{\varepsilon_r=10^{-4}}(\bar{x},t)| \quad |v_{\varepsilon_r=10^{-3}}(\bar{x},t) - v_{\varepsilon_r=10^{-4}}(\bar{x},t)| \\ &|u_{\varepsilon_r=10^{-4}}(\bar{x},t) - u_{\varepsilon_r=10^{-5}}(\bar{x},t)| \quad |v_{\varepsilon_r=10^{-4}}(\bar{x},t) - v_{\varepsilon_r=10^{-5}}(\bar{x},t)| \\ &|u_{\varepsilon_r=10^{-5}}(\bar{x},t) - u_{\varepsilon_r=10^{-6}}(\bar{x},t)| \quad |v_{\varepsilon_r=10^{-5}}(\bar{x},t) - v_{\varepsilon_r=10^{-6}}(\bar{x},t)|, \end{aligned} \tag{5.8}$$

where $\bar{x} = 0.5$. To evaluate the errors as indicated in (5.8), the Arneodo variables need to be built as continuous functions with respect to time: this is simply done through an interpolation procedure. The results are summarised in Fig. (5.28) for both the Arneodo variables: it is clear that the error becomes lower if the G-Scheme and WAMR relative tolerances are decreased.

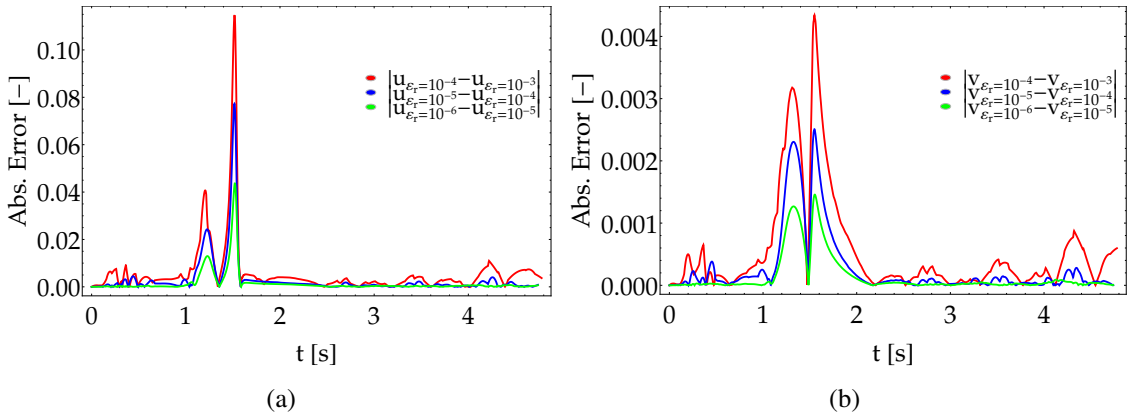


FIGURE 5.28: Time evolution of the absolute error of $u(\bar{x},t)$ and $v(\bar{x},t)$, with $\bar{x} = 0.5$, for different relative tolerances of the G-Scheme and WAMR.

Figure (5.1) shows the variation of the number of DoFs generated by WAMR in the four cases. More accurate time varying solutions are produced decreasing the tolerances from Test D to Test G as indicated in Tab. (5.6): the maximum number of generated DoFs increases from 298 (Test D) to 1906 (Test G).

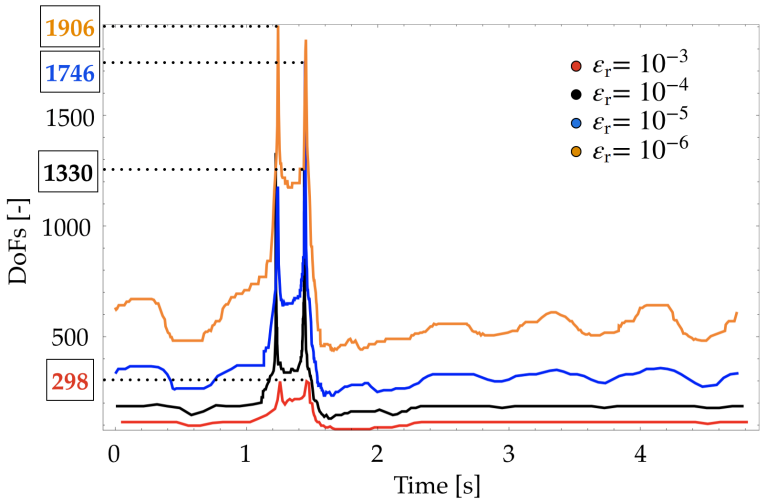


FIGURE 5.29: Number of DoFs generated by WAMR for the four tests of Tab. (5.6). The maximum values are 298 (Test D), 1330 (Test E), 1746 (Test F) and 1906 (Test G).

Figure (5.30) shows the time evolution of the number of DoFs produced by WAMR (red line), integrated by the G-Scheme (blue line) and associated with reference uniform grids for the all tests of Tab. (5.6). The reference uniform grids are characterized by a spacing equal to minimum grid size (maximum resolution level) reached in the corresponding WAMR/G-Scheme simulation, indicated in Tab. (5.7). One thing that clearly stands out is that the G-Scheme is able to perform a strong model reduction in time: only a few number of DoFs generated by WAMR are typically integrated. The tools to evaluate the efficiency of the wavelet and the G-Scheme compression are represented by the compression degrees introduced in section (4.1.1). Figure (5.31) shows the time

evolution of the compression degrees π_w, π_{gs} and π_O for ε_r going from $\varepsilon_r = 10^{-3}$ to $\varepsilon_r = 10^{-6}$. The G-Scheme compression degree π_{gs} is typically above the percentage 90% for all cases, with the exception of some time instants where the system behaviour is strongly nonlinear: in this case the efficiency drops and reaches the minimum value. The G-Scheme efficiency is particularly low around $t \approx 1.7$ s for Test F and Test G ($\pi_{gs} \approx 10\%$ for Test F and $\pi_{gs} \approx 20\%$ for Test G): this means that, at this time instants, the number of active equations is extremely near to the number of DoFs generated by WAMR. This is confirmed looking again at Fig. (5.30): at $t \approx 1.7$ s the blue curves of Figs. (5.30c)-(5.30d) (representing the time evolution of the dimension of the active subspace) are quite near to the red curves (representing the number of DoFs generated through the adaptive mesh refinement).

The performance are again studied through the evaluation of all the computational times associated with the solver WAMR/G-Scheme. Looking at Tab. (5.8) and the pie chart (5.32), it is clear that regardless of the relative tolerance, the most negative contribution for the workload is always represented by the time associated with the basis vectors evaluation. The orders of magnitude of this important contribution are also shown in the bar chart of Fig. (5.33): it is clear that the calculation of the basis vectors becomes extremely expensive decreasing the relative tolerances of WAMR and the G-Scheme (one has that the order of magnitude goes from 0.5 for Test D to ≈ 3.2 for Test G). This result shows that the workload is extremely worse if the requirement of more accurate solutions is imposed.

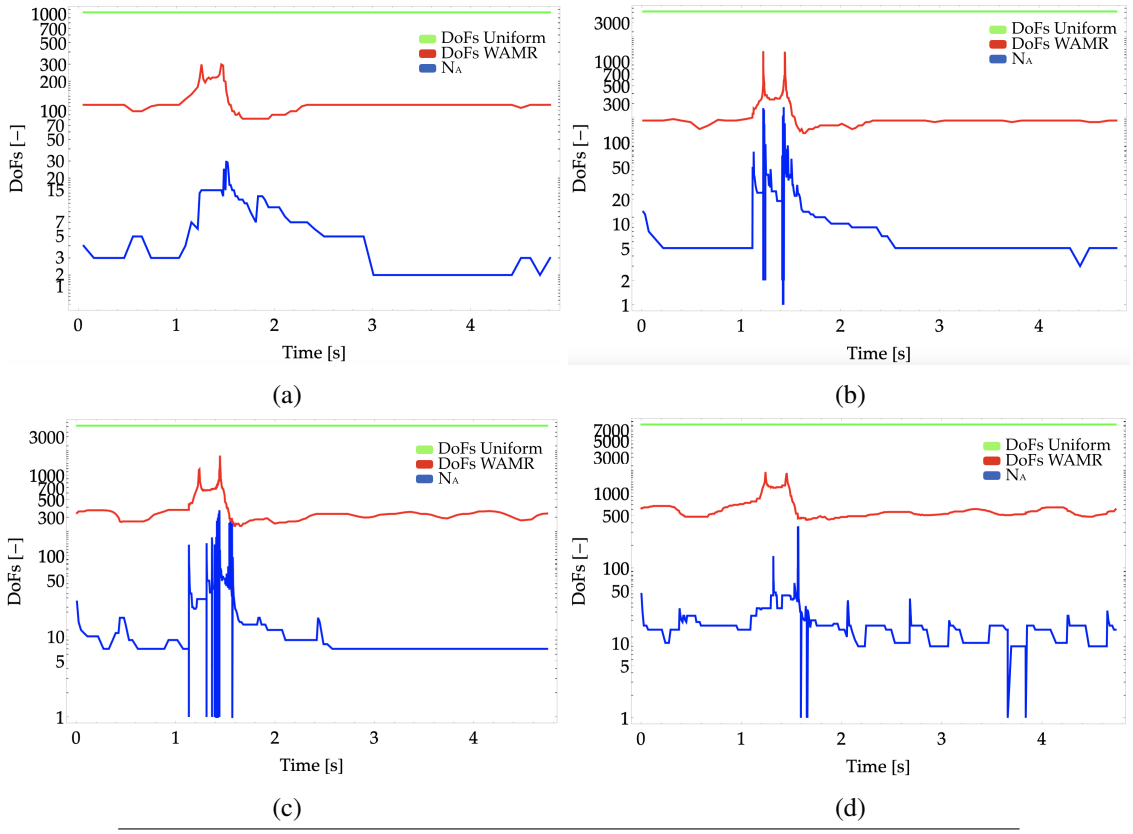


FIGURE 5.30: Time evolution of the number of DoFs for the tests of Tab. (5.6): (a) Test D - (b) Test E - (c) Test F - (d) Test G.

Test Name	Max. Resol. Lev. (J)	Min. Grid Size (Δx_{min})
Test D	5	1.9553125×10^{-3}
Test E	7	4.8828125×10^{-4}
Test F	7	4.8828125×10^{-4}
Test G	8	2.4414063×10^{-4}

TABLE 5.7: Maximum resolution level and associated minimum grid size for the tests of Tab. (5.6). The reference uniform grids are characterized by a spacing equal to the minimum grid size.

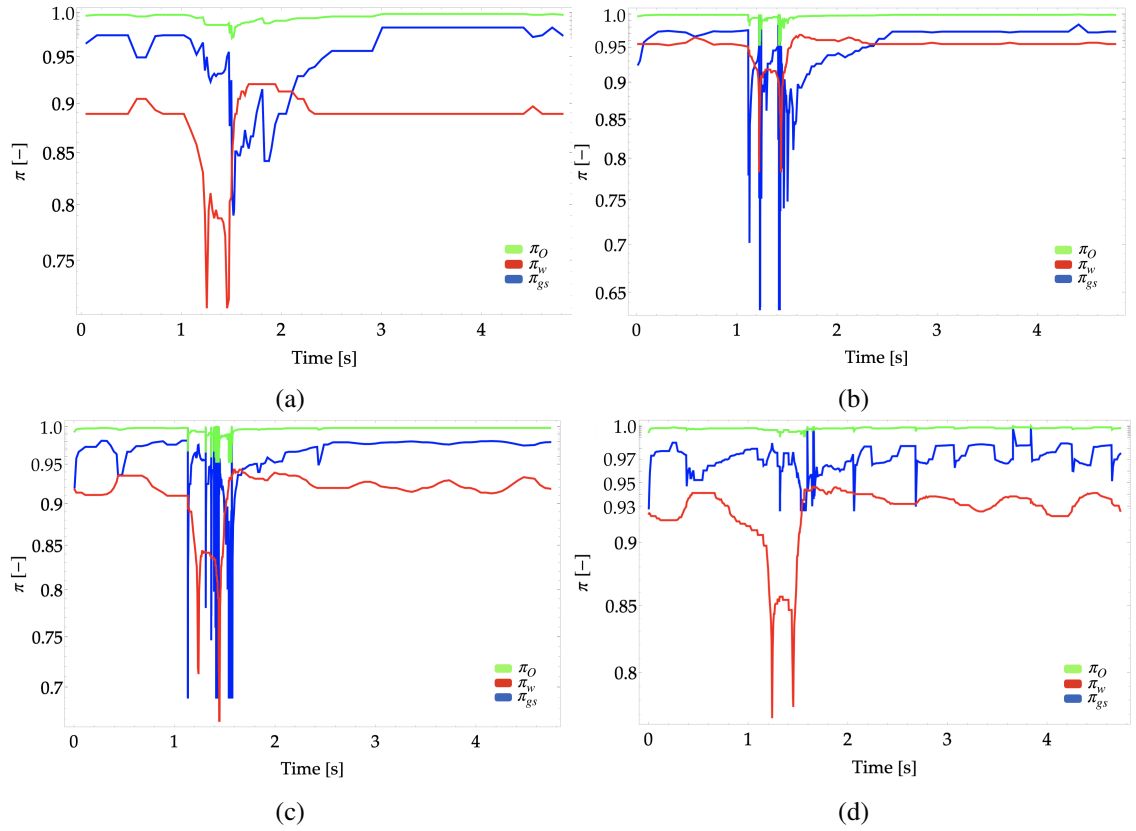


FIGURE 5.31: Time evolution of the compression degrees for the tests of Tab. (5.6): (a) Test D - (b) Test E - (c) Test F - (d) Test G.

ε_r [-]	Time WAMR [s]	Time Basis [s]	Time Other [s]	Total Time [s]
10^{-3}	0.076	3.179	0.316	3.571
10^{-4}	1.298	287.454	5.481	294.233
10^{-5}	5.578	1235.058	20.477	1261.113
10^{-6}	7.676	1878.047	33.406	1919.129

TABLE 5.8: Performance in terms of computational time as function of the relative wavelet threshold parameter. *Time Other* is given by the sum of the time to evaluate the Jacobian matrix, to diagonalize the Jacobian matrix, to integrate the PDEs (5.1), to find the tail and the head.

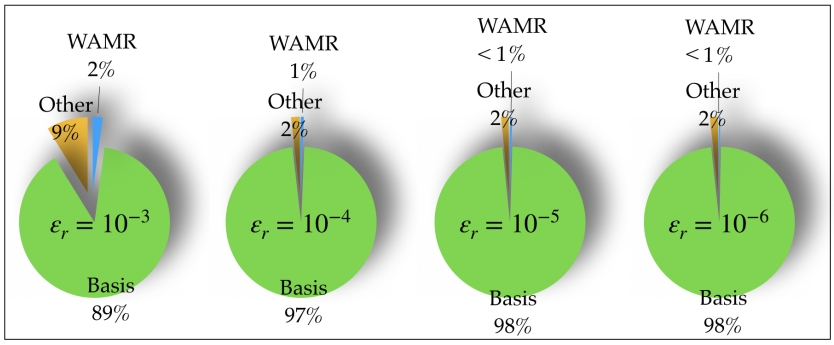


FIGURE 5.32: Percentages of the different contributions in the total computational time. The exact values are indicated in Tab. (5.8).

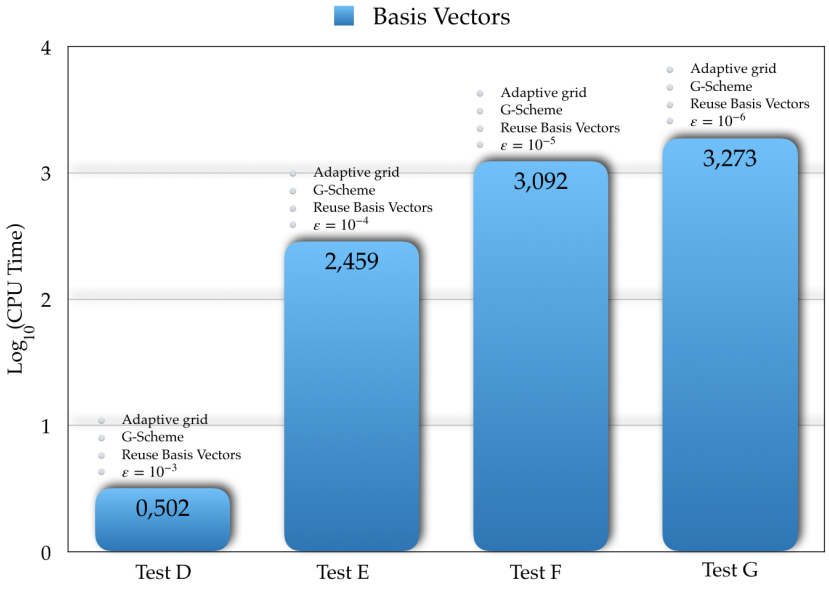


FIGURE 5.33: Order of magnitude of the computational time associated with the basis vectors evaluation. The basis vectors evaluation becomes more expensive when the relative tolerances become lower.

5.3.3 Parametric study with respect to the tail dimension - motivation

As explained in section (3.4) the AIM approach is an alternative attempt with respect to the G-Scheme to represent the system dynamics in a subspace smaller than the entire state-space, with the goal to generate a considerable simplification in the study of the dynamics of the original systems. However, as underlined in section (3.4.1), there are strong differences between the two strategies:

- following the Akram and Raman [2] approach, the state vector $\vec{v}$ of Eq. (3.26) is split into resolved $\vec{u}$ and unresolved $\vec{v}$ components: a projection operator $\mathcal{P}$ is defined to obtain $\vec{u}$ (Eq. (3.27)), while the complement of $\mathcal{P}$ (namely $\mathcal{Q} = \mathcal{I} - \mathcal{P}$) defines $\vec{w}$ (Eq. (3.27)). There is a lack of an a-priori criterion to identify the dimension of $\mathcal{P}$: it is taken constant in time and it is given by the first eigenfunctions of the linear operator $\mathcal{A}$ (Eq. (3.26)). In this sense, the resolved dynamics of the flow lie in this m-dimensional space, where $m \ll n_g$ [2] with n_g equal to the product between the number of dependent variables and the number of grid points;

- in the G-Scheme approach the dynamics is decomposed into active, slow, fast and, when applicable, invariant subspaces [49]. The G-Scheme introduces locally a curvilinear frame of reference, defined by a set of orthonormal basis vectors with corresponding coordinates, attached to this decomposition [49]. The evolution of the curvilinear coordinates associated with the active subspace is described by non-stiff DEs, whereas that associated with the slow and fast subspaces is accounted for by applying algebraic corrections derived from asymptotics of the original problem [49]. The dimension of the lower-dimensional manifold - the active subspace - is time varying and must be such that the contribution of the fast subspace over the driving time scale is negligible. This means that, at the base of the G-Scheme approach, there is a rigorous criterion to choose the dimension of the subspace in which the system dynamics are represented.

With the sole purpose to reproduce a similar idea at the base of Akram and Raman approach [2], the periodic bursting behaviour of the Arneodo model is again studied using the G-Scheme with forced constant fast subspace dimensions. Several tests are made with different constant dimensions of the fast subspace. The results are compared in terms of solution accuracy and performance to those ones obtained for a reference test, where the dimension of the fast subspace is not forced to be constant in time, but it is produced by the rigorous criterion at the base of the G-Scheme. The chosen reference test could be Test D or Test C (illustrated in section (5.3.1)), where adaptive grids are generated by WAMR and the reuse of the basis vectors is/is not activated, respectively. The good capability of Test C and Test D to generate accurate solutions with respect to those ones generated by DVODE and uniform grid (Test A) has been shown in section (5.3.1) (Figs. (5.14)-(5.15)). With the purpose to avoid errors introduced by the reuse of the basis vectors, Test C is chosen to be the reference test for the solutions comparisons.

Time varying and constant tail dimension

To understand the importance to have a time varying fast subspace dimension, the tests showed in Tab. (5.9) are considered: the reference case is Test H = Test C, where the fast subspace dimension is modified in time as prescribed by the G-Scheme, while it is constant in the other cases. Note that only cases where WAMR is used are considered: this choice lies on the just known capability to generate accurate solutions and better performance with respect to the cases where uniform grids are used. The tolerances are set equal to $\varepsilon_r = 10^{-3}$ and $\varepsilon_a = 10^{-8}$ for WAMR, $rtol_{head} = rtol_{tail} = 10^{-3}$ and $atol_{head} = atol_{tail} = 10^{-8}$ for the G-Scheme.

Test Name	Grid Type	Time Integrator	Tail Dimension	Reuse Basis Vectors
Test H = Test C	Adaptive (WAMR)	G-Scheme	Time Varying	Deactivated
Test I	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated
Test J	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated
Test K	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated
Test L	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated
Test M	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated
Test N	Adaptive (WAMR)	G-Scheme	To be Defined	Deactivated

TABLE 5.9: Tests structure: the reference case is Test H = Test C, where the fast subspace dimension varies in time. In the other cases, the fast subspace dimension is taken constant.

The reference test (Test H) is considered: the time evolution of the eigenvalues order of magnitude is shown in Fig. (5.34), where the red and blue lines are associated with the boundary between the active/fast subspace and active/slow subspace, respectively. The lack of DoFs and of the corresponding eigenvalues due to adaptive mesh refinement performed by WAMR can be clearly appreciated looking at Zone A and Zone B: this means that no grey points (namely eigenvalues) are here detected.

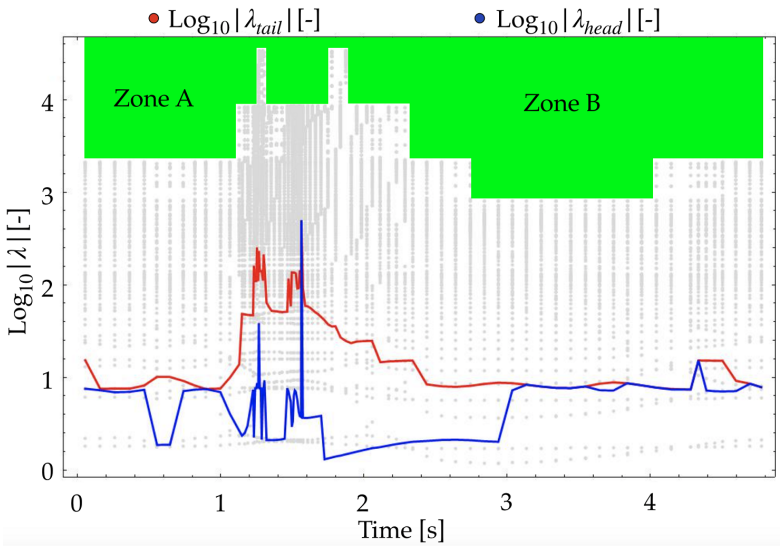


FIGURE 5.34: Time evolution of the eigenvalues order of magnitude of the Jacobian matrix of the Arneodo model (5.1). The red and blue lines represent the boundary between the active/fast subspace and active/slow subspace, respectively.

Figure (5.35) shows the time evolution of the tail dimension (black line) for the reference test (Test H): it is evaluated to achieve the largest degree of reduction satisfying a user-prescribed accuracy tolerance. The six dashed lines are associated with constant tail dimensions chosen to perform the simulations from Test I to Test N, whose numerical values are shown in Tab. (5.10). They are above the mathematical average (red line), approximatively equal to 19. By fixing the time instant, if the value of the chosen tail dimension is above that one indicated by the black curve, the requirement of the largest degree of reduction is not satisfied: the degree of reduction satisfies a tolerance lower than that one prescribed by the user. This means that it is expected to obtain better solutions when the tail dimension rises from 23 (Test I) to 73 (Test N). In this sense, Test N represent the most conservative case, where the tail dimension is constantly larger than that one prescribed by the G-Scheme (black line), with the exception of a sole time instant ($t \approx 1.58$ s) where the tail dimension is exactly equal to that one predicted by the G-Scheme: this point is identified by the intersection of the black curve with the blue dashed curve.

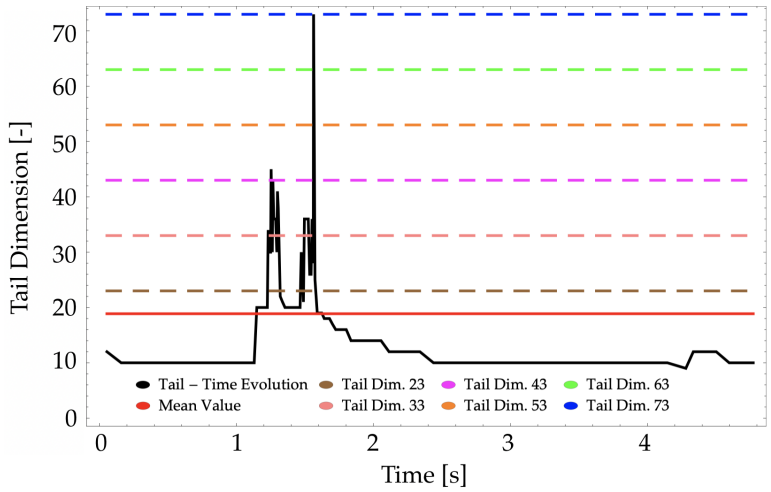


FIGURE 5.35: Time evolution of the tail dimension. The case of time varying tail dimension (Test H) is represented by the black curve, while the red line is associated with its mathematical average. The dashed lines are related to the six imposed values of the tail dimension.

Test Name	Grid Type	Time Integrator	Tail Dimension	Reuse Basis Vectors
Test H = Test C	Adaptive (WAMR)	G-Scheme	Time Varying	Deactivated
Test I	Adaptive (WAMR)	G-Scheme	23	Deactivated
Test J	Adaptive (WAMR)	G-Scheme	33	Deactivated
Test K	Adaptive (WAMR)	G-Scheme	43	Deactivated
Test L	Adaptive (WAMR)	G-Scheme	53	Deactivated
Test M	Adaptive (WAMR)	G-Scheme	63	Deactivated
Test N	Adaptive (WAMR)	G-Scheme	73	Deactivated

TABLE 5.10: Tests structure: the reference case is Test H = Test C, where the fast subspace dimension varies in time. In the other cases, the fast subspace dimension is taken constant.

The time evolution of the dependent variables at $x = 0.5$ is represented in Fig. (5.36) for all the cases listed in Tab. (5.10), while Fig. (5.37) shows the limit cycles: solutions getting closer and closer to the reference one (Test H - black curves of Figs. (5.36)-(5.37)) are obtained increasing the tail dimension from 23 to 73. The results are confirmed looking at Fig. (5.38), where the errors calculated as defined in Eqs. (5.7) are shown. Note that in Fig. (5.37) the limit cycle associated with the tail dimension $T = 23$ is not represented, because of the strong deviation from the reference limit cycle.

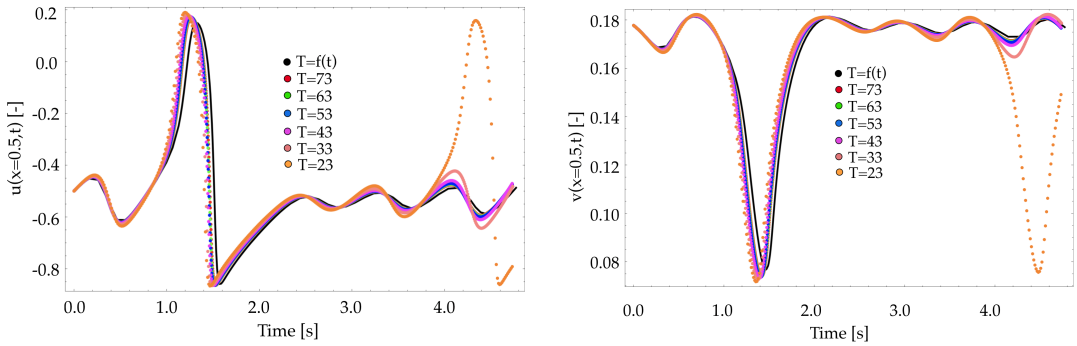


FIGURE 5.36: Time evolution of the dependent variables at $x = 0.5$ for all the tests listed in Tab. (5.10).

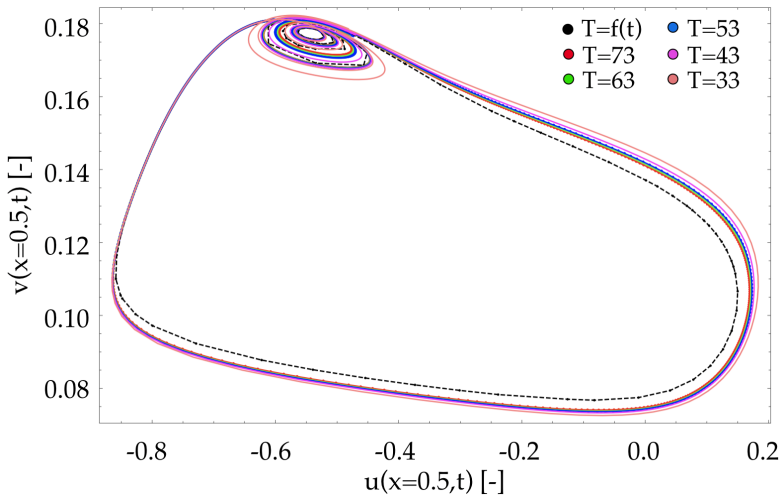


FIGURE 5.37: Limit cycles for all the tests listed in Tab. (5.10). Note that the case of $T = 23$ is not shown because of the strong deviation from the reference cycle (black dashed line), as can be appreciated looking at Fig. (5.36).

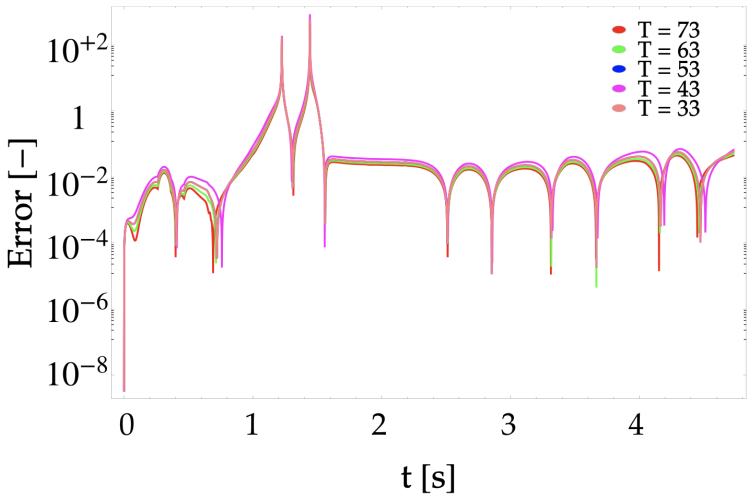


FIGURE 5.38: Time evolution of the errors defined in Eqs. (5.7).

The time variation of the eigenvalues order of magnitude associated with the tail are shown in Fig. (5.39) for all tests listed in Tab. (5.10): when the fast subspace dimension rises from 23 to 73, the eigenvalues order of magnitude associated with the boundary between the fast/active subspace increase. This means that also the integration time steps calculated by the G-Scheme algorithm ($\Delta t_s \approx 1/|\lambda_T|$) rise. The results shown in Fig. (5.40) confirm the expectations: the number of integration time steps required in the different tests fall gradually from 2739 (Test N) to 356 (Test I). The minimum number of integration time steps is obtained with a time varying tail dimension ($T = 101$). The advantage to have a time varying tail dimension as prescribed by the G-Scheme algorithm is also confirmed looking at the performance evaluated as shown by Eq. (4.5), Fig. (5.41): the most important contribution, represented by the time to evaluate the basis vectors, drops from 87.684 s (Test N) to 16.261 s (Test I). However, the best performance are again obtained for Test H: the computational time to calculate the basis is 5.593 s. The sum of the time to evaluate the Jacobian matrix, to diagonalize the Jacobian matrix, to calculate the tail and the head and to integrate the PDEs (5.1), that is called *Other* in Fig. (5.41), represents a minimal contribution. However, also this contribution decreases when the integration time steps increase (from 14.754 s for Test N to 16.261 s for Test I). Its minimum value (that is 0.472) is again obtained for the reference test (Test C = Test H).

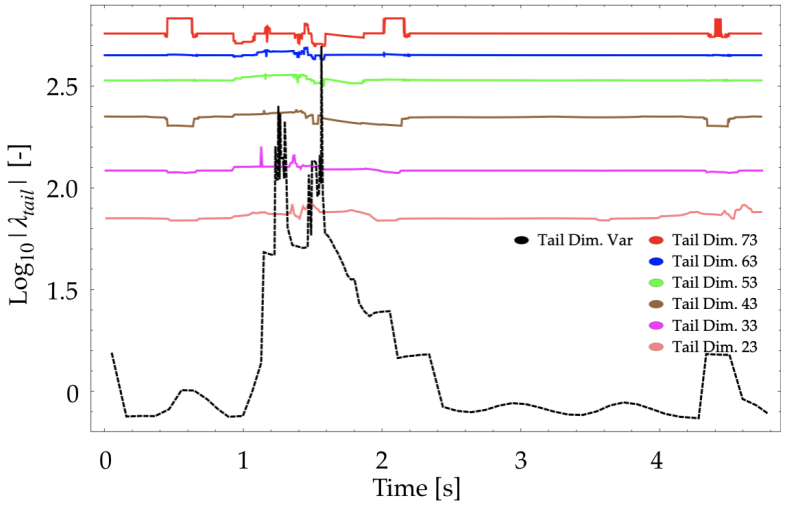


FIGURE 5.39: Time evolution of the eigenvalues order of magnitude associated with the tail for all the tests listed in Tab. (5.10).

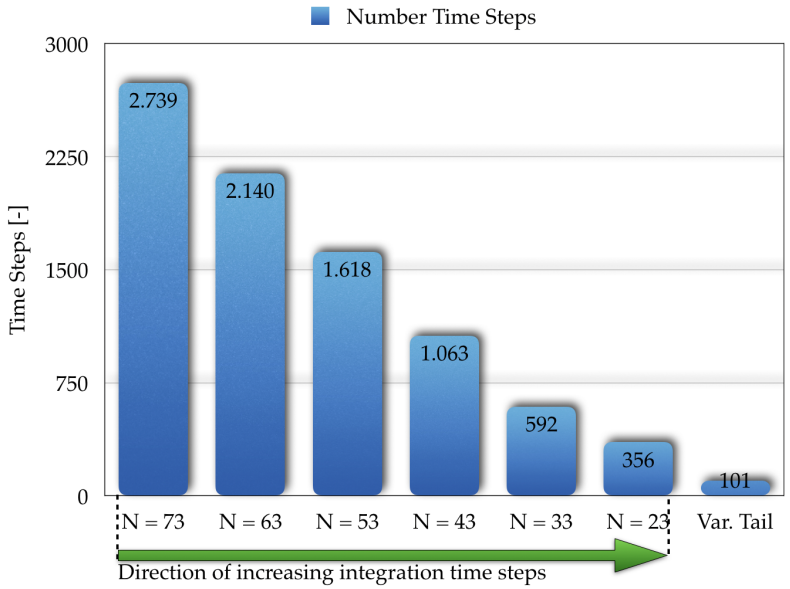


FIGURE 5.40: Integration time steps evaluated by the G-Scheme for all the tests listed in Tab. (5.10).

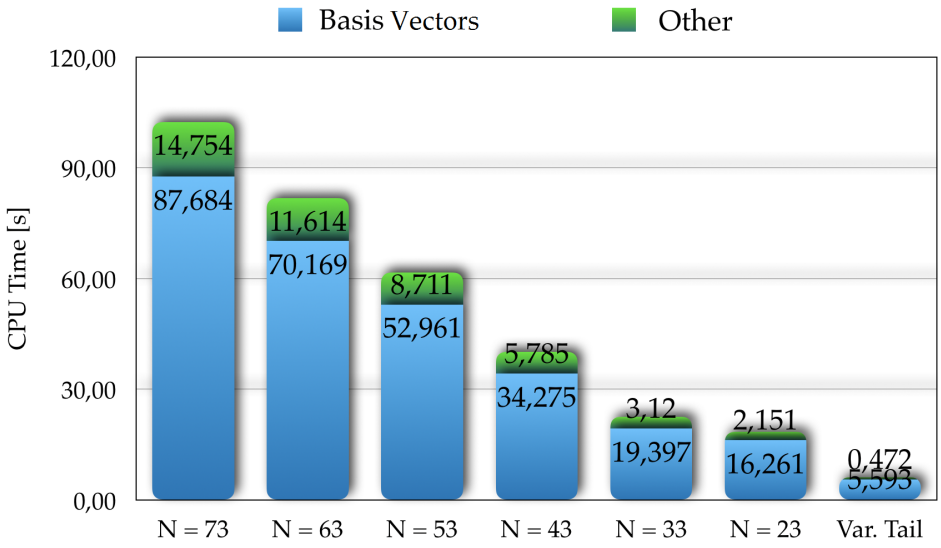


FIGURE 5.41: Workload associated with the tests listed in Tab. (5.10). The sum of the time to evaluate the Jacobian matrix, to diagonalize the Jacobian matrix, to calculate the tail and the head and to integrate the PDEs (5.1) is called *Other*.

CHAPTER 6

The Unsteady Flamelets Model

As a more complex reaction-diffusion system, the unsteady flamelet model for non-premixed combustion is studied. Section (6.1) is dedicated to a brief description of this model and the governing equations are presented. In section (6.2) the comparison between the time evolution of adaptive solutions and reference solutions is presented. Finally, in section (6.3) the performance of the coupling between WAMR and the G-Scheme are analysed, using the G-Scheme in the OS technique described in section (4.3.1).

6.1 The unsteady flamelet model for non-premixed combustion

In non-premixed combustion the fuel and the oxidizer are non-mixed before they enter the combustion chamber. Looking at the model represented in Fig. (6.1), the fuel and oxidizer are injected from the left hand side and right hand side, respectively. They diffuse towards the reaction zone, where they burn and generate heat. In this region the temperature is maximum and diffuses away from the flame front towards the oxidizer and fuel streams.

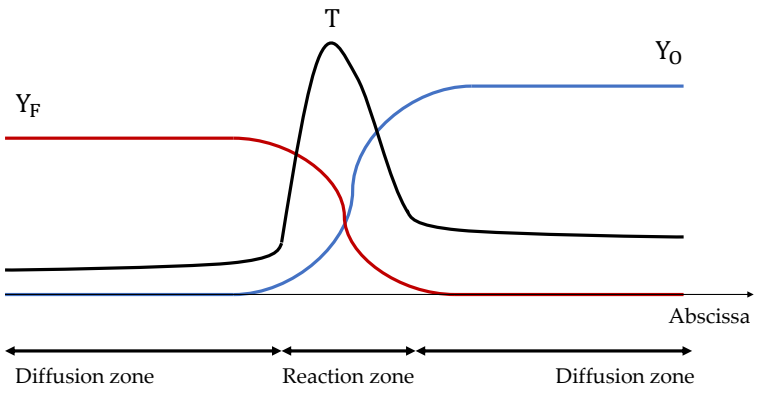


FIGURE 6.1: Flame structure: the black curve represents the temperature variation along the abscissa, while the red and blue curves the fuel and oxidizer mass fractions, respectively. The temperature peak is reached in the reaction zone.

The unsteady flamelet equations taken by Pitsch and Peters [38] are considered for the adiabatic case, at constant pressure and unity Lewis number,

$$\frac{\partial Y_i}{\partial t} = \frac{\chi}{2}\frac{\partial^2 Y_i}{\partial z^2} + \frac{\dot{\omega}_i}{\rho} \qquad i = 1, \ldots, N_s \tag{6.1}$$

$$\frac{\partial T}{\partial t} = \frac{\chi}{2}\frac{1}{c_p}\frac{\partial^2 h}{\partial z^2} - \frac{\chi}{2}\frac{1}{c_p}\sum_{k=1}^{N_s} h_k \frac{\partial^2 Y_k}{\partial z^2} - \frac{1}{\rho c_p}\dot{\omega}_T \tag{6.2}$$

where t is the time, Y_i represents the mass fractions of the i-th chemical species, χ is the scalar dissipation rate, z is the mixture fraction variable, $\dot{\omega}_i$ is the production rate for the i-th species due to chemical reactions, ρ is the density, T is the temperature, c_p is the specific heat at constant pressure, h is the enthalpy of the mixture, N_s is the total number of mixture components, h_k is the enthalpy of the k-th species, and $\dot{\omega}_T = \sum_{k=1}^{N_s} h_k \dot{\omega}_k$ is the rate of enthalpy production due to the variation in the composition of the mixture. The scalar dissipation rate is expressed as a function of the mixture fraction as shown by Girimaji [13], $\chi = \chi^0_{max}\exp(-2\mathrm{erfc}^{-1}(2z))^2$.

6.2 Comparison between adaptive and reference time varying solutions

The flamelet equations (6.1)-(6.2) are integrated for pressure $p = 20$ atm and scalar dissipation rate $\chi^0_{max} = 200\ \mathrm{s}^{-1}$. The kinetic mechanism for CO, CH_2O and CH_3OH combustion [28] is considered, taking into account 12 chemical species, namely CO_2, H_2O_2, HO_2, O_2, HCO, CO, H_2O, OH, O, H_2, H and N_2. The thermodynamic properties of the fluid are evaluated with the ideal gas Equation of State (EoS). The initial conditions are represented in Fig. (6.2) in terms of mass fractions with respect to the mixture fraction. The initial temperature is taken constant along the mixture fraction and equal to 1200 K.

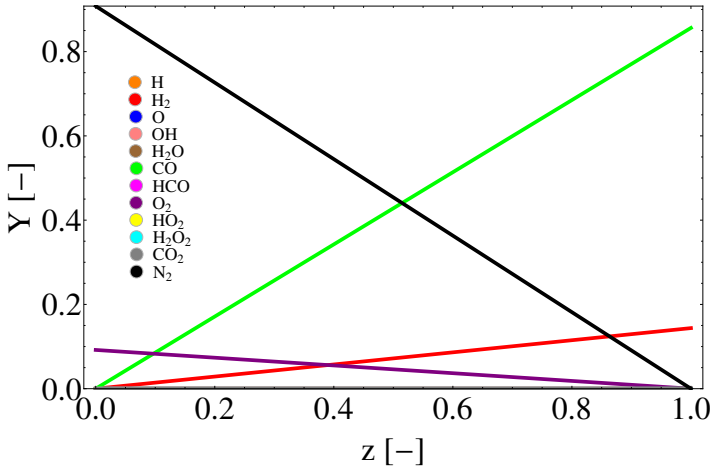


FIGURE 6.2: Initial conditions in terms of mass fractions with respect to the mixture fraction.

6.2.1 Tests structure and parameters setting

To describe the structure of the tests, Fig. (6.3) is considered. It represents an extended version of the basic scheme (4.1), to which reference is made for the meaning of the symbols. The adaptive solutions are built using the coupling between WAMR and the G-Scheme. Fourth order wavelet functions are adopted, fixing the relative and absolute wavelet threshold parameters equal to $\varepsilon_r = 10^{-3}$ and $\varepsilon_a = 10^{-8}$, respectively, while the relative and absolute tolerances of the G-Scheme are set equal to $r_{tol_{tail}} = r_{tol_{head}} = 10^{-3}$ and $a_{tol_{tail}} = a_{tol_{head}} = 10^{-8}$, respectively.

The reference solution is built on a uniform grid characterized by a grid spacing corresponding to the finest resolution required in the WAMR/G-Scheme simulation. The time integration is carried out through DVODE and the integration time steps are assigned with the aim to compare the time evolution of the temperature and concentrations along the mixture fraction variable with respect to the results of the simulation where WAMR and the G-Scheme are employed. In this sense, the integration time step trend is assigned from the simulation where WAMR/G-Scheme are used. The steady state is supposed to be reached when the RMS of the RHS of equations (6.1)-(6.2) becomes lower than a fixed prescribed minimum value (equal to 10^{-7}). The absolute and relative tolerances of DVODE are $a_{tol} = 10^{-12}$ and $r_{tol} = 10^{-10}$, respectively.

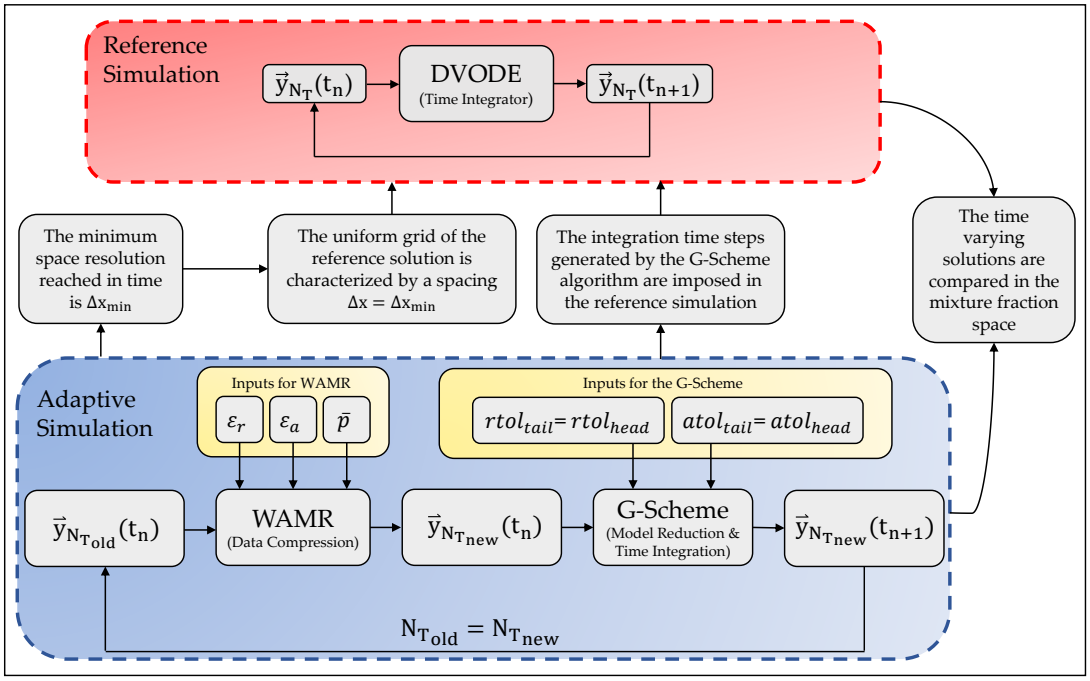


FIGURE 6.3: Procedure to build and compare the reference and adaptive steady state solutions. The scheme represented in the blue block is an extension of the scheme (4.1), to which reference is made for the unknown symbols. The scheme represented in the red block is a simplified version of the scheme where WAMR is used: the number of mesh points do not change. The minimum spacing reached in the simulations where WAMR is adopted is used to build the uniform grid for the reference tests.

6.2.2 The model validation

The steady state is reached in a time interval equal to $t_{ss} \approx 1.5 \times 10^{-3}$ s. The comparison between the time evolution of the reference and adaptive solutions is represented in Fig. (6.4) in terms of temperature with respect to the mixture fraction at five time instants, while in Fig. (6.5) the mass fraction of the molecular oxygen (as example of relevant concentration) is traced with respect to the mixture fraction: the solid lines show the reference transient solutions, while the markers are associated with the simulation where WAMR and the G-Scheme are employed. The comparison is not proposed for $t > t_5$, because significant time variations of the temperature and concentrations are not visibly detected for $t_5 < t < t_{ss}$: in this time interval the system is quite near to the steady state. The excellent level of accuracy produced by the G-Scheme and WAMR can be clearly appreciated: the time evolution of the reference solutions for the temperature and concentration of O_2 are well captured by the adaptive scheme.

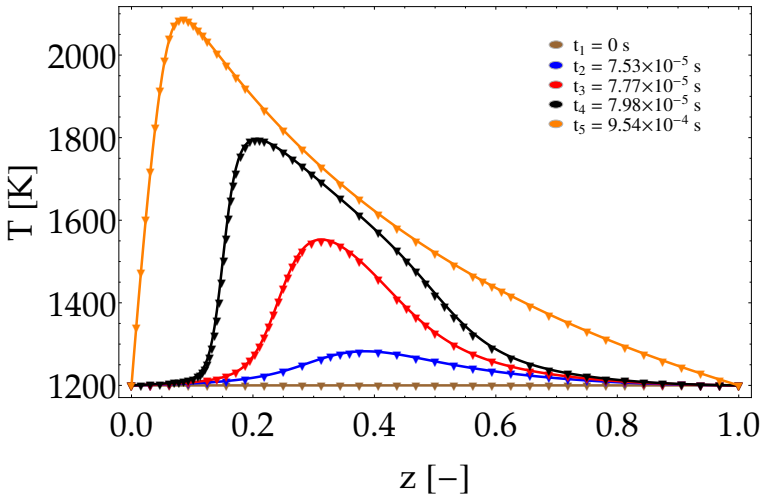


FIGURE 6.4: Solutions comparison in terms of temperature with respect to z for five values of t - in the WAMR/G-Scheme simulation the relative wavelet threshold parameter is $\varepsilon_r = 10^{-3}$; the dashed lines represent the reference curves, while the triangles are associated with the results of the WAMR/G-Scheme simulation.

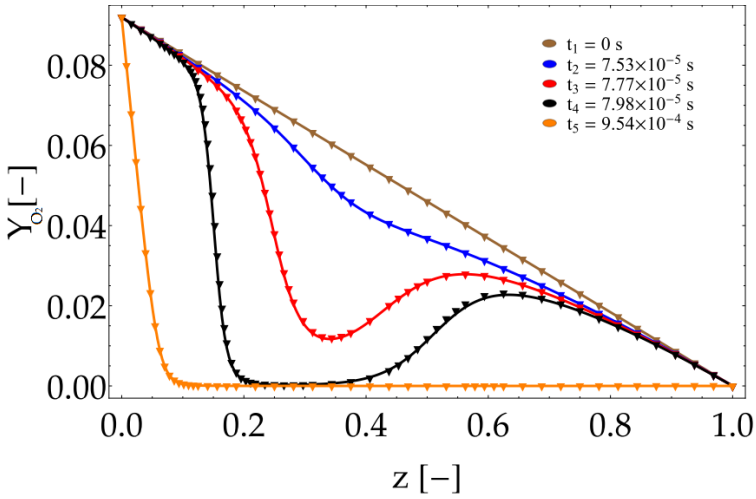


FIGURE 6.5: Solutions comparison in terms of Y_{O_2} with respect to z for five values of t - in the WAMR/G-Scheme simulation the relative wavelet threshold parameter is $\varepsilon_r = 10^{-3}$; the dashed lines represent the reference curves, while the triangles are associated with the results of the WAMR/G-Scheme simulation.

6.2.3 Time evolution of the number of DoFs and compression degrees

Figure (6.6) shows the time evolution of the number of DoFs produced by WAMR (green line), integrated by the G-Scheme (blue line) and associated with the reference uniform grid (red line). The number of DoFs generated by WAMR is significantly lower than that one associated with the reference uniform mesh: the maximum value reached during the transient of the system is 845 for WAMR at $t \simeq 8.2 \times 10^{-4}$ s, because here higher resolution levels are required to well capture the gradients of the variables. At the same time instant, the number of active equations is also maximum and equal to $N_A = 671$. For $t \geq 10^{-4}$ s significant variations of the number of DoFs generated by WAMR are not observed and the trend is quite flat (DoFs WAMR = 661): this result confirms that for $t \geq 10^{-4}$ s the system is quite near to the steady state and strong variations of the solutions are not longer detected by WAMR.

Figure (6.7) shows the time evolution of the compression ratios π_w, π_{gs} and π_O with respect to time. They are evaluated as shown in Eqs. (4.2)-(4.3)-(4.4), respectively. Looking at the Fig. (6.7), one thing that clearly stands out is that the WAMR compression degree remains fairly constant ($\pi_w \simeq 82\%$) during the approach to the steady state (from $t \simeq 8.2 \times 10^{-4}$ s to t_{ss} s). On the contrary, in the same time interval, the G-Scheme compression increases significantly from the minimum $\pi_{gs} \simeq 20\%$ (at $t \simeq 8.2 \times 10^{-4}$ s) to the maximum $\pi_{gs} \simeq 100\%$ (at $t \simeq 1.5 \times 10^{-3}$ s): this means that the steady state is approached with only one active mode. Finally, it can be observed that the overall efficiency of the adaptive scheme is always up to the minimum value $\pi_{O_{min}} = 80\%$ detected at $t \simeq 8.2 \times 10^{-4}$ s.

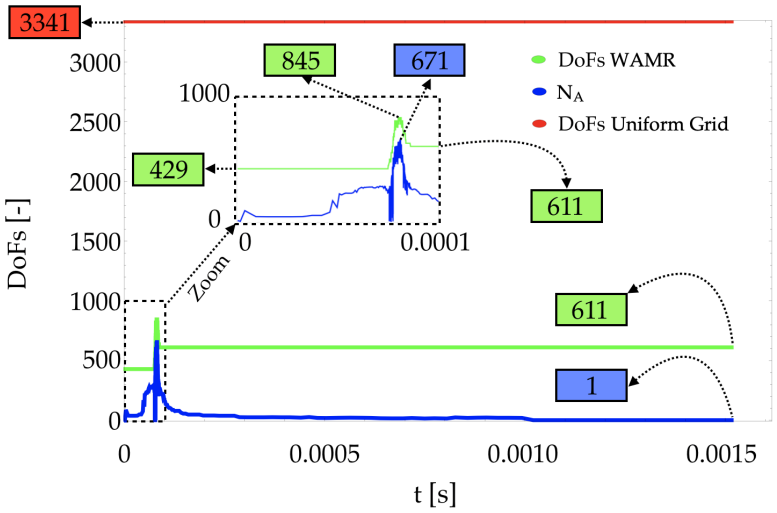


FIGURE 6.6: Number of DoFs with respect to time generated by WAMR (green line), integrated by the G-Scheme (blue line) and associated with the reference uniform grid (red line).

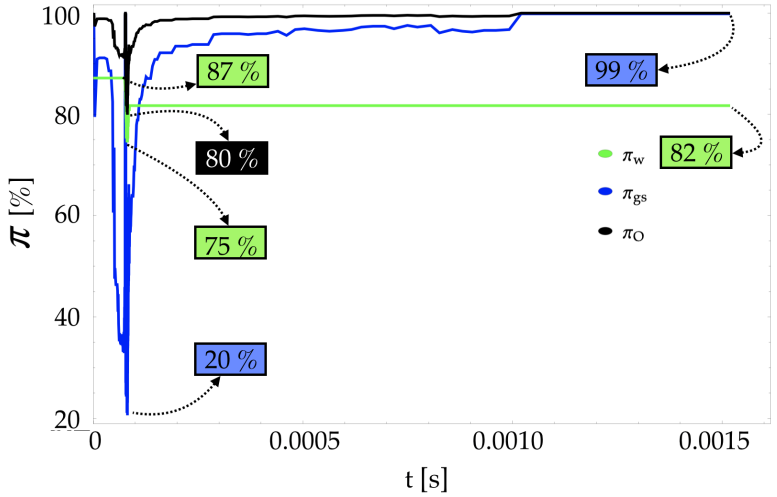


FIGURE 6.7: Time evolution of the compression ratios - the compression degrees associated with WAMR (π_w) and the G-Scheme (π_{gs}) are represented with the green and blue lines, respectively, while the the total compression ratio (π_O) is identified with the black curve.

6.2.4 Time scales evolution

Figure (6.8) represents the time evolution of the Jacobian matrix eigenvalues, in terms of orders of magnitude. The active DoFs integrated by the G-Scheme are included between the red and blue lines. The TSR curve, represented with a green continuous line shows that the most energetic scale is not always included in the active subspace. The dimension of the active subspace is strongly time varying until the steady state is approached. As expected, at this condition the number of active equations is exactly equal to $N_A = 1$. Figure (6.9) shows the modes shapes along the mixture fraction associated with the selected eigenvalue in Fig. (6.8) (yellow dot at the steady state), belonging to the active subspace and detected at the steady state.

Looking at the Fig. (6.7), the WAMR efficiency is maximum and equal to $\pi_w \simeq 87\%$ in the time interval $t \in [0, 10^{-4}]$ s: a few number of DoFs (approximatively constant and equal to $\simeq 429$, Fig. (6.6)) are required to describe the time evolution of the variables along z. This means that, in this time interval, the coupling between WAMR and the G-Scheme results in a lack of DoFs and the corresponding eigenvalues, as Fig. (6.8) shows: no grey points are detected in the upper part of the plot. The time evolution of the integration time step produced by the G-Scheme is shown in Fig. (6.10): it is represented by the reciprocal of the eigenvalues associated with the red line (the tail) of Fig. (6.8). The minimum integration time steps is detected at $t \approx 10^{-4}$ s, where the tail dimension is maximum and the G-Scheme efficiency is minimum ($\pi_{gs} \approx 20\%$).

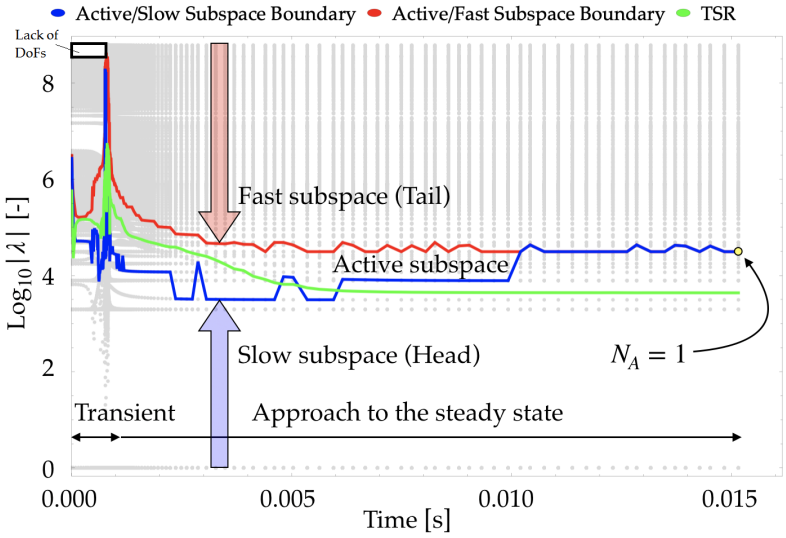


FIGURE 6.8: Time evolution of the Jacobian matrix eigenvalues in terms of orders of magnitude - the active subspace is included between the blue and red lines.

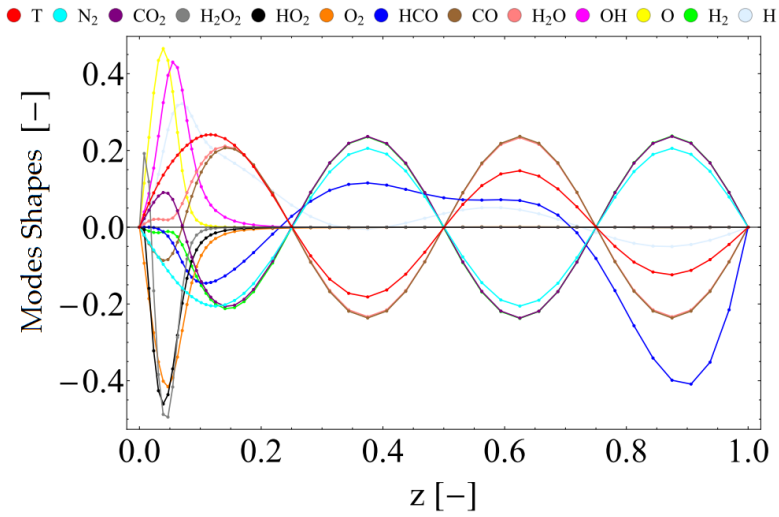


FIGURE 6.9: Modes shapes associated with the eigenvalue detected at the steady state (yellow point of Fig. (6.8)).

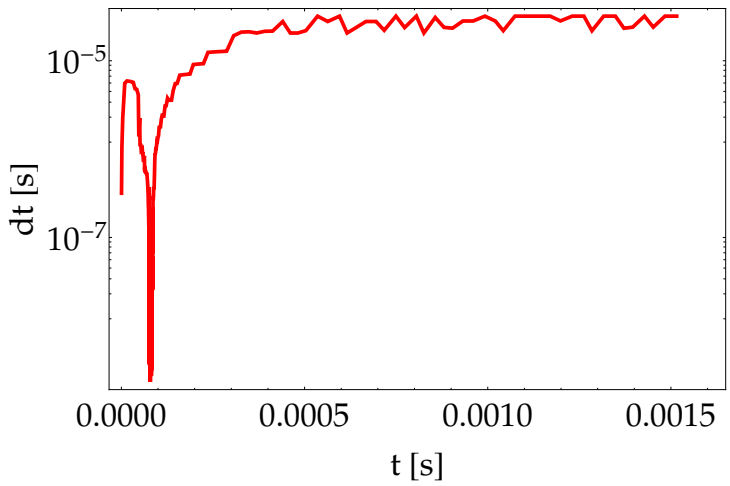


FIGURE 6.10: Time evolution of the integration time steps generated by the G-Scheme.

6.3 Performance evaluation

Tests are made with the aim to evaluate the performance of the coupled scheme (4.1) for the flamelet problem. The tests structure is shown in Tab. (6.1) and Tab. (6.2): the performance are firstly calculated without and with the basis vectors reuse technique, Test A and Test B, respectively. Therefore, the OS technique described in section (4.3.1) is also employed. The parameters setting for WAMR and the G-Scheme are shown in Tab. (6.3).

Whenever the OS technique is used, the integration time step (Δt) needs to be imposed. It is assigned knowing the minimum grid size (Δx_{min}) (*i.e.* the maximum reached resolution level) through a CFL-like condition, as made for the Arneodo model with the procedure described in Eq. (5.2): the parameter K is now assigned looking at the last column of Tab. (6.2), while D is represented by the coefficient of the diffusive term of the flamelet equations (6.1)-(6.2).

The performance for Test A and Test B are summarized in Figs. (6.11)-(6.12), respectively. As expected, the basis vectors evaluation is less expensive if the G-Scheme basis vectors are reused: the total computational time goes from $\approx$ 40000 s to $\approx$ 36000 s. The strong drop of the workload due to the use of the splitting technique by the scheme (4.3.1) can be clearly appreciated looking at Fig. (6.14): the total CPU time drops approximatively of 1.5 order of magnitude with respect to the Test A and Test B.

Test Name	Grid Type	Time Integrator	Reuse CSP Basis
Test A	Adaptive (WAMR)	G-Scheme	Deactivated
Test B	Adaptive (WAMR)	G-Scheme	Activated

TABLE 6.1: Tests structure: Test A and Test B are made using adaptive grids generated by WAMR and the G-Scheme, activating or not the reuse of the G-Scheme basis vectors.

Test Name	Grid Type	Time Integrator	CFL
Test C	Adaptive (WAMR)	G-Scheme with O.S.	0.35
Test D	Adaptive (WAMR)	G-Scheme with O.S.	0.40
Test E	Adaptive (WAMR)	G-Scheme with O.S.	0.45
Test F	Adaptive (WAMR)	G-Scheme with O.S.	0.50

TABLE 6.2: Tests structure: the OS technique described in section (4.3.1) is employed to build local Jacobian matrices. It is expected to obtain a strong reduction of the computational cost associated with the use of the G-Scheme.

ε_r	ε_a	$rtol_{head}$	$rtol_{tail}$	$atol_{head}$	$atol_{tail}$
10^{-3}	10^{-8}	10^{-3}	10^{-3}	10^{-8}	10^{-8}

TABLE 6.3: Parameters setting for the tests listed in Tab. (6.1) and Tab. (6.2).

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	2668,257	37180,012	67,597	8,991	41,930	51,485

CPU Total Time [s]
40018,272

Percentage Reuse Basis Vectors
0

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
0	6,668	92,908	0,169	2,247E-02	0,105	0,129

FIGURE 6.11: Performance evaluation for Test A in terms of computational times.

CPU Time [s]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
2,996	2709,616	32945,289	68,221	13,025	33,335	99,136

CPU Total Time [s]
35871,618

Percentage Reuse Basis Vectors
30,885

Percentage Values [-]						
Time Diag. Jac.	Time Jac.	Time Find Basis	Time Tail	Tail Head	Time PDEs	Time WAMR
8,352E-03	7,554	91,842	0,190	0,036	0,093	0,276

FIGURE 6.12: Performance evaluation for Test B in terms of computational times.

CPU Total Time [s]			
Test C	Test D	Test E	Test F
1098,374	1192,979	1027,050	976,527

FIGURE 6.13: Total computational time associated with Test C, Test D, Test E and Test F.

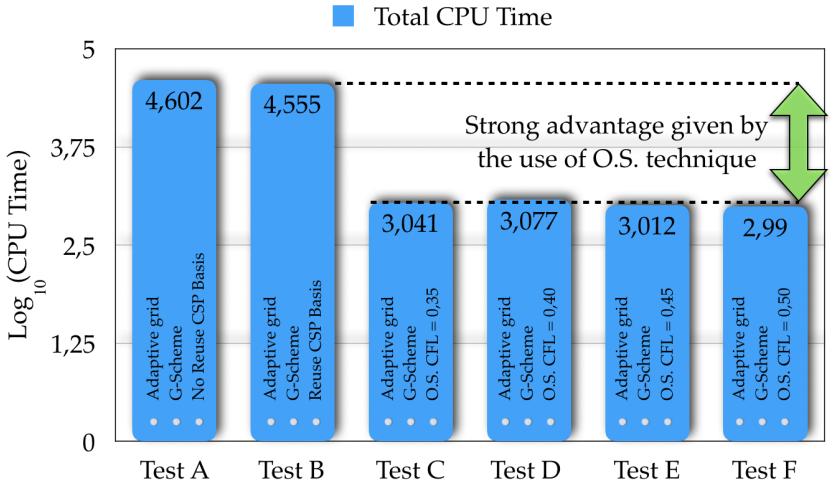


FIGURE 6.14: Performance comparison in terms of total computational times (orders of magnitude). Better performance are obtained using the G-Scheme to integrate the source term of the flamelet equations with the OS technique described in section (4.3.1).

CHAPTER 7

Unsteady Flamelets: Supercritical Thermodynamic Conditions

The requirement of high performance in liquid rockets and diesel engine combustion can be achieved by increasing the pressure in the combustion chamber. This could lead to the injection of oxidizer and/or fuel at pressure and temperature exceeding the critical point. Beyond this point the fluid is in supercritical thermodynamic conditions. In this context a study of the unsteady flamelets for non-premixed combustion at high pressure conditions is proposed. Their behaviour have been fully investigated by Kim et al. [18, 19] using a direct approach, where the solution is given in the mixture fraction space. By following this approach, they performed steady state analysis for kerosene/LOx rocket combustion [18] and hydrogen/liquid oxygen [19] in supercritical states. Unsteady effects of non-premixed methane/oxygen flame structures at supercritical pressures have been instead explored by Lapenna et al. [26], choosing an arbitrary non-uniform grid suitably compressed towards the oxidizer side. The main purpose of the work is to overcome this limitation, by using WAMR to build a dynamically adaptive grid able to reach higher resolution to capture the steep gradients characterizing the solution.

Using the WAMR algorithm to determine the dynamically adaptive grid for the flamelet problem, the computational cost is expected to be largely reduced: because in trans-critical and near-critical conditions the thermodynamic and transport properties are characterized by abrupt variations, steep gradients are expected to be well captured by the algorithm, with a relative reduced number of grid points and a consequent large reduction of the computational cost. Therefore, accurate chemical kinetic mechanisms involving a large number of chemical species and reactions can be used to solve flamelet problems. Furthermore, more complex and expensive EoS can be adopted to better describe the fluid behaviour in non-ideal conditions. These features allow the generation of flamelet databases for high-pressure combustion devices which usually operate under supercritical conditions.

The first part of the work is dedicated to the description of the main features of the real gas EoS used in this context. The strategy to calculate the parameters involved in the EoS is described for pure chemical species and multi-components mixture. Through the knowledge of all the parameters characterizing the EoS, the thermodynamic properties are evaluated and validated for specific chemical species. The theory described in section (7.1) and section (7.2) is taken by

[18]. Finally, the thermodynamic properties are used to solve particular flamelet problems under supercritical conditions, using WAMR to generate adaptive grids. The last part of the chapter is dedicated to suggestions for a possible future work, the generation of multi-dimensional tabulated flamelets using WAMR. This work will allow to obtain a multi-dimensional table with the defined accuracy.

7.1 Real gas Equation of State

Figure (7.1) represents a phase diagram of a generic substance: single phase regions are separated by lines where phase transitions occur. The three phases (gas, liquid, and solid) coexist in thermodynamic equilibrium at the triple point, while for temperature and pressure values larger than the critical temperature and pressure (T_c and p_c, respectively) the supercritical state is reached.

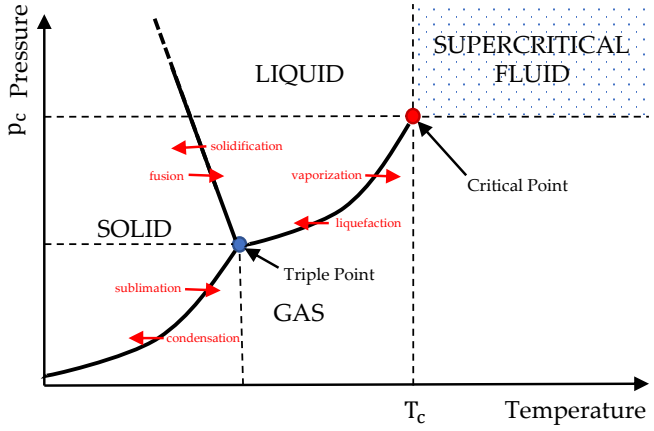


FIGURE 7.1: Phase diagram - the triple and critical points are identified with the blue and red points, respectively. The fluid is in supercritical thermodynamic conditions if $T \geq T_c$ and $p \geq p_c$.

In this work the three-parameters Redlich-Kwong/Peng-Robinson (RK-PR) [5] EoS is used to evaluate the thermodynamic properties of the fluid in supercritical conditions: this choice allows to have an accurate representation of the real fluid properties in an efficient way [18]. It takes the form

$$p = \frac{\rho R_u T}{M_w - b\rho} - \frac{a\alpha(T)\rho^2}{(M_w + \delta_1 b\rho)(M_w + \delta_2 b\rho)}. \tag{7.1}$$

where p is the pressure, ρ is the density, T is the temperature, R_u is the universal gas constant and M_w is the molecular weight. Through the parameters a and b the effects of the attractive and repulsive van der Waals forces are considered in the model, respectively; these parameters are functions of the thermodynamic variables at critical conditions and of the third parameter δ_1. They can be found also in the two-parameter cubic SRK (Soave-Redlick-Kwong) EoS and PR (Peng-Robinson) EoS, where δ_1 is instead an assigned costant. The additional DoF of RK-PR EoS introduced by [5], δ_1, (note that $\delta_2 = f(\delta_1)$) leads to a better description of the fluid behaviour in terms of density, with respect to the two-parameter cubic SRK EoS and PR EoS usually used.

Finally, $\alpha(T)$ is a temperature correction factor taking into account the polarity of the species. It depends on the reduced temperature (defined as T/T_c) and the acentric factor ω.

7.1.1 Parameters evaluation for pure chemical species

To evaluate the parameters a, b, δ_1 and δ_2 of Eq. (7.1), the critical properties of the individual chemical species and their acentric and compressibility factor Z_c must be evaluated. In particular, for each chemical species, the critical properties and the acentric factor are assigned and taken by the National Institute of Standard and Technology (NIST) database [12]. For instance, for the CH_4 and O_2 one has:

	CH_4	O_2
T_c [K]	190.564	154.581
p_c [bar]	45.9920	50.4300
ρ_c [kg/m^3]	162.660	463.100
ω [-]	0.01142	0.02220

TABLE 7.1: Critical properties and acentric factor for CH_4 and O_2.

where ρ_c is the critical density. Knowing also the molecular weight M_w of the chemical species, the compressibility factor is calculated:

$$Z_c = \frac{p_c \times M_w}{\rho_c \times R_u \times T_c}, \tag{7.2}$$

Considering again the chemical species CH_4 and O_2 for which the molecular weight are $M_{w,CH_4} \simeq 16\text{g/mol}$ and $M_{w,O_2} \simeq 32\text{g/mol}$, respectively, one has:

$$Z_{c,O_2} = 0.271, \quad Z_{c,CH_4} = 0.286 \tag{7.3}$$

All the parameters involved in the RK-PR EoS can be evaluated (more details can be found in Cismondi et al. [5]) following the procedure illustrated in Fig. (7.2).

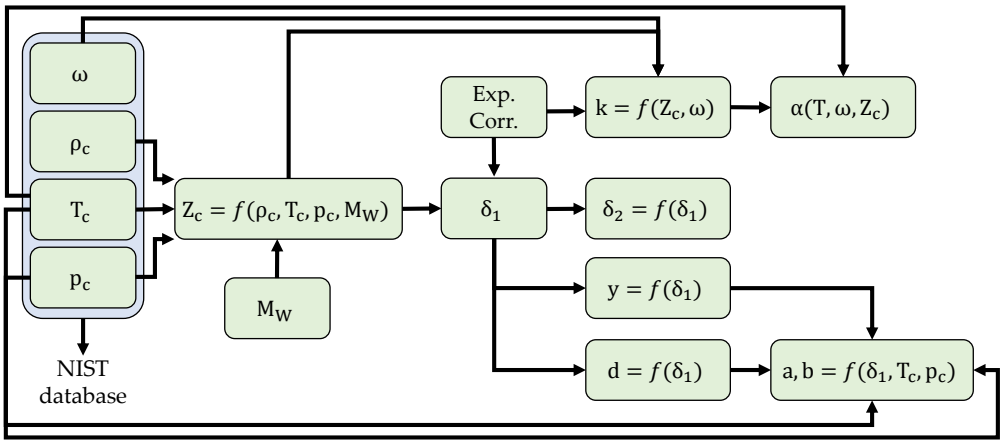


FIGURE 7.2: Flow diagram - RK-PR EoS parameters evaluation for pure chemical species.

The procedure shown in Fig. (7.2) is here briefly described (see Cismondi et al. [5] for details).

$$\delta_1 = d_1 + d_2(d_3 - 1.168Z_c)^{d_4} + d_5(d_3 - 1.168Z_c)^{d_6}, \tag{7.4}$$

where

d_1	d_2	d_3	d_4	d_5	d_6
0.428363	18.496215	0.338426	0.660000	789.723105	2.512392

$$\delta_2 = \frac{1-\delta_1}{1+\delta_1}, \tag{7.5}$$

$$a = \frac{3y^2 + 3yd + d^2 + d - 1}{(3y + d - 1)^2}\left(\frac{R_u^2 T_c^2}{p_c}\right), \tag{7.6}$$

$$b = \frac{1}{3y + d - 1}\left(\frac{R_u T_c}{p_c}\right), \tag{7.7}$$

$$\alpha(T) = \left(\frac{3}{2 + T/T_c}\right)^k, \tag{7.8}$$

$$d = \frac{1+\delta_1^2}{1+\delta_1}, \tag{7.9}$$

$$y = 1 + \left(2(1+\delta_1)\right)^{\frac{1}{3}} + \left(\frac{4}{1+\delta_1}\right)^{\frac{1}{3}}, \tag{7.10}$$

$$k = (1.168Z_c A_1 + A_0)\omega^2 + (1.168Z_c B_1 + B_0)\omega + (1.168Z_c C_1 + C_0), \tag{7.11}$$

where

A_1	A_0	B_1	B_0	C_1	C_0
-2.44070	0.00170	7.45130	1.96810	12.5040	-2.72380

The results are shown for the chemical species CH_4 and O_2 in Tab. (7.2).

	CH_4	O_2
δ_1	0.915475	1.95992
δ_2	0.0441273	-0.324306
a	0.232625	0.146396
b	2.99×10^{-5}	2.06×10^{-5}
$\alpha(T)$	$5.243\left(\frac{1}{2+5.245\times10^{-3}T}\right)^{1.50825}$	$4.318\left(\frac{1}{2+6.469\times10^{-3}T}\right)^{1.33147}$
d	0.959603	1.63561
y	3.8429	3.91458
k	1.50825	1.33147

TABLE 7.2: Evaluation of the parameters involved in the RK-PR EoS for the chemical species CH_4 and O_2.

7.1.2 Extension to multi-component mixture

For a multi-components mixture the parameters involved in the RK-PR EoS (7.1) are defined using conventional mixing rules

$$a\alpha = \sum_{i=1}^{N}\sum_{j=1}^{N} X_i X_j a_{ij}\alpha_{ij}, \tag{7.12}$$

$$b = \sum_{i=1}^{N} X_i b_j, \tag{7.13}$$

$$\delta_1 = \sum_{i=1}^{N} X_i \delta_{1,i}, \qquad \delta_2 = \sum_{i=1}^{N} X_i \delta_{2,i}, \tag{7.14}$$

where X_i is the species molar fraction, while $\delta_{1,i}$, $\delta_{2,i}$ and b_j are calculated for each chemical species as illustrated in Eqs. (7.4)-(7.5)-(7.7), respectively. The first and second derivatives with respect to temperature of the term $a\alpha$ are directly used for the derivation of the thermodynamic properties, as will be explained in section (7.2),

$$\frac{\partial a\alpha}{\partial T} = \sum_{i=1}^{N}\sum_{j=1}^{N} X_i X_j \frac{\partial a_{ij}\alpha_{ij}}{\partial T} = \sum_{i=1}^{N}\sum_{j=1}^{N} X_i X_j a_{ij} \frac{\partial \alpha_{ij}}{\partial T}, \tag{7.15}$$

$$\frac{\partial^2 a\alpha}{\partial T^2} = \sum_{i=1}^{N}\sum_{j=1}^{N} X_i X_j \frac{\partial^2 a_{ij}\alpha_{ij}}{\partial T^2} = \sum_{i=1}^{N}\sum_{j=1}^{N} X_i X_j a_{ij} \frac{\partial^2 \alpha_{ij}}{\partial T^2}. \tag{7.16}$$

The parameter a_{ij} of Eqs. (7.15)-(7.16) is calculated as shown in Eq. (7.6), where the critical pressure is now replaced with the pseudo-critical parameter $p_{c,ij}$ (introduced by Miller et al. [33])

$$p_{c,ij} = Z_{c,ij}\frac{R_u T_{c,ij}}{v_{c,ij}} \quad \text{with} \quad v_{c,ij} = \frac{1}{8}(v_{c,i}^{1/3} + v_{c,j}^{1/3})^3, \tag{7.17}$$

where $v_{c,j}$ represents the critical specific volume of pure chemical species, given by the ratio between the molecular weight and the critical density. The pseudo-critical parameters $T_{c,ij}$, $Z_{c,ij}$ and ω_{ij} (again introduced by Miller et al. [33]) are defined as

$$T_{c,ij} = \sqrt{T_{c,i}T_{c,j}}, \quad Z_{c,ij} = \frac{1}{2}(Z_{c,i} + Z_{c,j}), \quad \omega_{ij} = \frac{1}{2}(\omega_i + \omega_j). \tag{7.18}$$

The first and second derivatives of α_{ij} with respect to temperature are calculated taking the first and second derivatives of $\alpha(T)$ from Eq. (7.8), where T_c, Z_c and ω are replaced with $T_{c,ij}$, $Z_{c,ij}$ and ω_{ij}, respectively.

7.2 Thermodynamic properties evaluation

The mixture density ρ is calculated solving the cubic EoS (7.1), given the thermodynamic state (T, p) and all the mixture parameters. By differentiating Eq. (7.1) with respect to T, ρ and ρ_i, the following differential expressions are obtained

$$\left(\frac{\partial p}{\partial T}\right)_{\rho_j} = \frac{\rho R_u}{M_w - b\rho} - \left(\frac{\partial a\alpha}{\partial T}\right)\frac{\rho^2}{(M_w + \delta_1 b\rho)(M_w + \delta_2 b\rho)}, \tag{7.19}$$

$$\left(\frac{\partial p}{\partial \rho}\right)_{T,Y_j} = \frac{M_w R_u T}{(M_w - b\rho)^2} - \frac{a\alpha\rho M_w(2M_w + (\delta_1 + \delta_2)b\rho)}{(M_w + \delta_1 b\rho)^2(M_w + \delta_2 b\rho)^2}, \tag{7.20}$$

$$\begin{aligned}\left(\frac{\partial p}{\partial \rho_i}\right)_{T,\rho_{j\neq i}} = {} & \frac{M_w R_u T}{M_{w,i}(M_w - b\rho)^2}(M_w + \rho(b_i - b)) - \frac{2\rho M_w \sum_j X_j a_{ij}\alpha_{ij}}{M_{w,i}(M_w + \delta_1 b\rho)(M_w + \delta_2 b\rho)} + \\ & + \frac{a\alpha\rho^2 M_w b_i}{M_{w,i}(M_w + \delta_1 b\rho)^2(M_w + \delta_2 b\rho)^2}((\delta_1 + \delta_2)M_w + 2\delta_1\delta_2 b\rho).\end{aligned} \tag{7.21}$$

Equations (7.19)-(7.20)-(7.21) are repeatedly used in the derivation of all the thermodynamic properties involved in the solution of the flamelet problem described by Eqs. (6.1)-(6.2). In particular, focus is given on the derivation of the quantities directly involved in Eqs. (6.1)-(6.2), namely the partial-mass enthalpy of the species h_i, the specific enthalpy of the mixture h and the constant-pressure specific heat c_p,

$$h_i = \tilde{e}_i - \frac{\left(\sum_j Y_j \tilde{e}_j - e - \frac{p}{\rho}\right)}{\left(\frac{\partial p}{\partial \rho}\right)_{T,Y_j}}\left(\frac{\partial p}{\partial \rho_i}\right)_{T,\rho_{j\neq i}}, \tag{7.22}$$

$$h = \sum_{i=1}^{N} Y_i h_i, \tag{7.23}$$

$$c_p = c_v + \frac{T}{\rho^2} \frac{\left(\frac{\partial p}{\partial T}\right)^2_{\rho_j}}{\left(\frac{\partial p}{\partial \rho}\right)_{T,Y_j}}, \tag{7.24}$$

where the partial-density internal energy of the species $\tilde{e}_i$, the mixture internal energy e and the heat capacity at constant-volume c_v are:

$$\begin{aligned} \tilde{e}_i = \left(\frac{\partial \rho e}{\partial \rho_i}\right)_{T,\rho_{j\neq i}} = e_{i,0} + \frac{2\sum_j X_j \left[T\left(\frac{\partial a_{ij}\alpha_{ij}}{\partial T}\right) - a_{ij}\alpha_{ij}\right]}{(\delta_1-\delta_2)bM_{w,i}} \ln\left(\frac{M_w+\delta_1 b\rho}{M_w+\delta_2 b\rho}\right) + \\ + \frac{b_i \left[T\left(\frac{\partial a\alpha}{\partial T}\right) - a\alpha\right]}{(\delta_1-\delta_2)bM_{w,i}} \left[\frac{(\delta_1-\delta_2)\rho M_w}{(M_w+\delta_1 b\rho)(M_w+\delta_2 b\rho)} - \frac{1}{b}\ln\left(\frac{M_w+\delta_1 b\rho}{M_w+\delta_2 b\rho}\right)\right], \end{aligned} \tag{7.25}$$

$$e(T,\rho) = e_0(T) + \frac{1}{(\delta_1-\delta_2)bM_w}\left[T\left(\frac{\partial a\alpha}{\partial T}\right) - a\alpha\right]\ln\left(\frac{M_w+\delta_1 b\rho}{M_w+\delta_2 b\rho}\right), \tag{7.26}$$

$$c_v = \left(\frac{\partial e}{\partial T}\right)_{\rho,Y_j} = c_{v,0} + \frac{T}{(\delta_1-\delta_2)bM_w}\left(\frac{\partial^2 a\alpha}{\partial T^2}\right)\ln\left(\frac{M_w+\delta_1 b\rho}{M_w+\delta_2 b\rho}\right). \tag{7.27}$$

7.3 Validation of the thermodynamic properties

The thermodynamic properties based on the RK-PR EoS are compared to the data taken from the National Institute of Standard and Technology (NIST) database [12], considering a pure methane compound, whose critical properties are indicated in Tab. (7.1). Figures (7.3)-(7.4) show the values of ρ and c_p for different values of the pressure, in the temperature range $100 - 400$ K: for $T \geq T_c$ the fluid is in supercritical conditions. The continuous lines, representing the EoS results, are in good agreement with the reference data, represented by the dots, even if slight errors are present in the zone where there is a transition from subcritical to supercritical conditions. These inaccuracies are attributed to the limitations of the RK-PR EoS, although it provides a more accurate modelling of the density in comparison with the two-parameter EoS. Comparison data for the two-parameter EoS are provided in the study of Kim et al. [18] for saturated hydrocarbons with varying carbon number: the SRK EoS seems to well predict the density for species having a higher critical compressibility factor, instead the PR EoS works better for the larger hydrocarbons characterized by lower critical compressibility factors.

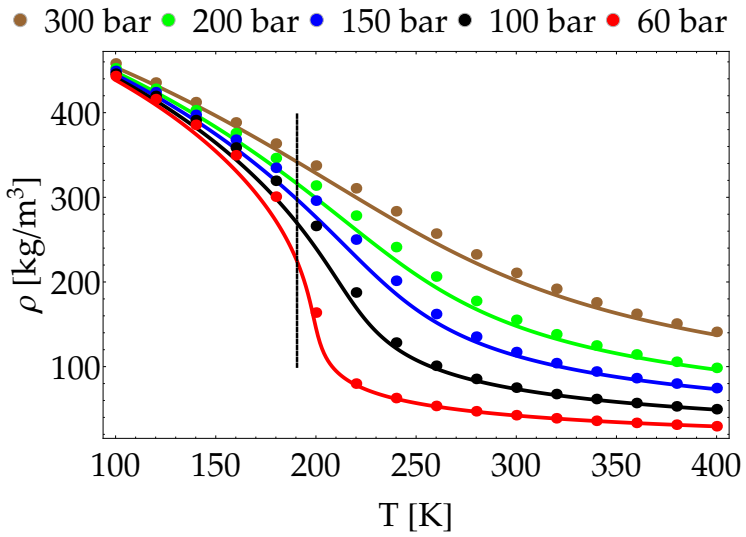


FIGURE 7.3: Density at different pressure and temperature values; the dots represent the NIST data, while the continuous lines are the values from RK-PR EoS; the vertical black line represents the critical temperature for methane, $T_{c,CH_4} = 190.564$ K.

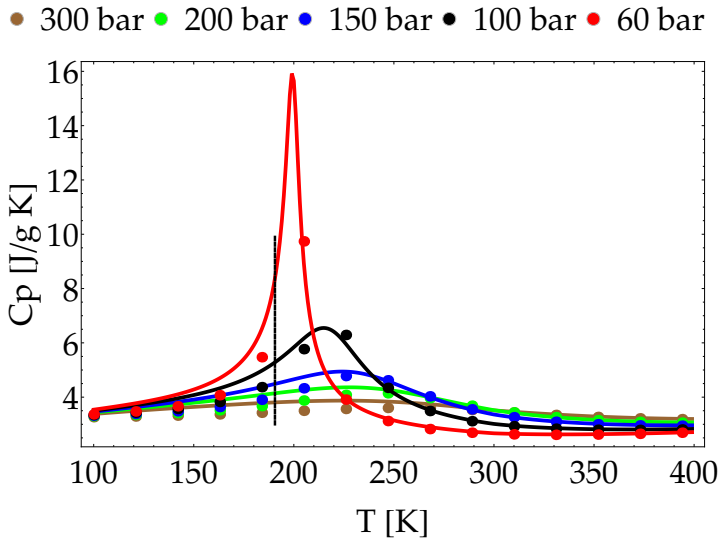


FIGURE 7.4: c_p at different pressure and temperature values; the dots represent the NIST data, while the lines are the values from RK-PR EoS; the vertical black line represents the methane critical temperature for methane, $T_{c,CH_4} = 190.564$ K.

7.4 Results and discussion

The flamelet equations (6.1)-(6.2) are solved with DVODE ($atol = 10^{-12}$ and $rtol = 10^{-10}$) considering the CH_4/O_2 RAM accelerator kinetic mechanism (RAMEC, Petersen et al. [37]), taking into account 38 chemical species and 190 reactions. The equations are initially integrated for $p = 60$ bar and $\chi^0_{max} = 200$ s^{-1}, choosing $\varepsilon_r = 10^{-3}$ and sixth-order wavelet functions. The critical pressures and temperatures are $p_{c,CH_4} = 45.992$ bar and $T_{c,CH_4} = 190.564$ K for the

methane, $p_{c,O_2} = 50.430$ bar and $T_{c,O_2} = 154.581$ K for the oxygen [12]. The oxygen and methane are injected at temperatures equal to $T^{in}_{O_2} = 120$ K and $T^{in}_{CH_4} = 1346$ K, so they are in liquid-like state and supercritical state, respectively.

The time integration of the flamelet equations is performed through DVODE and the steady state is supposed to be reached when the RMS of the equations RHS becomes lower than a fixed prescribed minimum value. The steady state solution is represented in Fig. (7.5) in terms of temperature: the total number of grid points, at the chosen space accuracy, is $N_T = 80$.

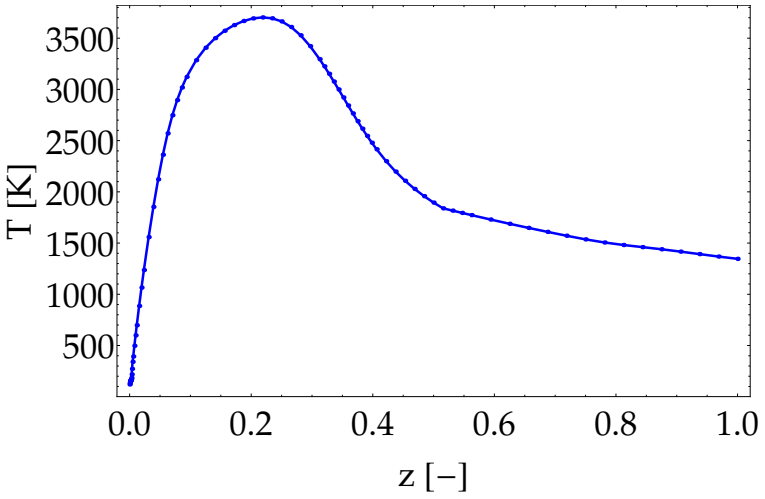


FIGURE 7.5: Steady state solution - temperature for a wavelet threshold parameter $\varepsilon_r = 10^{-3}$.

The wavelet compression degree π_w (Eq. (4.2)) is considered as tool to show the efficiency of the wavelet compression. In the current test, the compression ratio is equal to $\pi_w \simeq 98\%$. A large grid points concentration is present in the vicinity of $z = 0$, where the maximum reached resolution level is $J = 8$: here strong variations of the c_p are present, as shown in Figs. (7.6)-(7.7). This result highlights the excellent capability of the wavelet method to well capture strong variations of the variables, adding grid points at higher resolution levels to ensure the achievement of the prescribed accuracy.

The computational cost associated with the evaluation of the thermodynamic properties in near-critical and supercritical state according with RK-PR EoS represents approximatively the 95 % of the requested time to compute the RHS of Eqs. (6.1)-(6.2). The result can be justified by the highly expensive combination rules adopted to evaluate the thermodynamic properties through the RK-PR EoS for a multi-component mixture. Because of the RHS has to be evaluated in each point of the grid, the use of an adaptive grid allows to reduce substantially the total computational time needed to reach the steady state.

Table (7.3) summarizes the results of different simulations, for $p = 60$ bar, $\chi^0_{max} = 200$ s^{-1} and ε_r varying from 10^{-2} to 10^{-4}: a decrease of the threshold parameters corresponds to an upward trend of the number of grid points N_T, J and π_w. Indeed, higher resolution levels have to be added in order to obtain more accurate representations of the solution. In particular, Fig. (7.8) shows how the solution representation changes in the three cases looking at the temperature in the vicinity of $z = 0$, where the maximum resolution level is typically reached because of the extreme

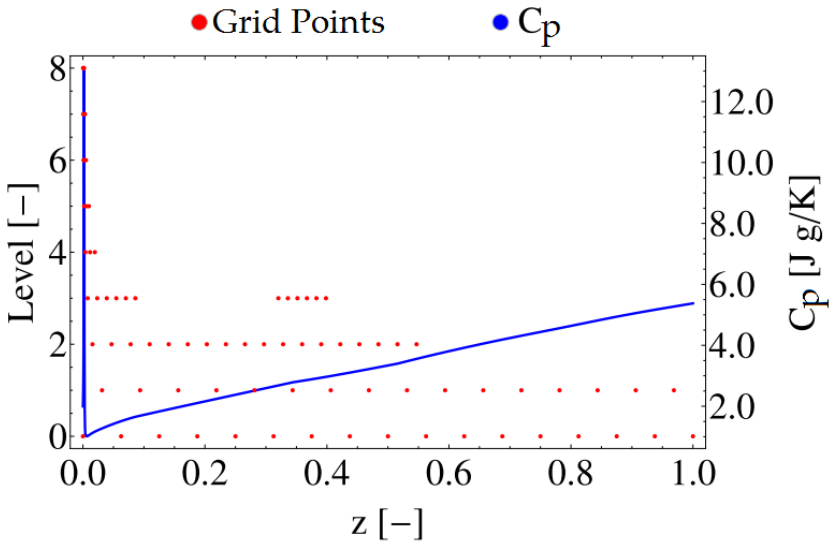


FIGURE 7.6: c_p and grid points at different resolution levels; the wavelet threshold parameter is $\varepsilon_r = 10^{-3}$.

variations of the c_p. Figure (7.8) shows clearly that using $\varepsilon_r = 10^{-2}$ the wavelet-based algorithm will capture only the temperature trend in the range $z = 0 - 5 \times 10^{-3}$; conversely, in the cases with $\varepsilon_r = 10^{-3}$ and $\varepsilon_r = 10^{-4}$ WAMR automatically refines the grid, the triangular symbols represent the added points, capturing also the relatively small temperature variations.

ε_r	J	N_T	π_w	Δz_{min}
10^{-2}	2	35	$\simeq 54\%$	3.125×10^{-2}
10^{-3}	8	80	$\simeq 98\%$	2.441×10^{-4}
10^{-4}	11	114	$\simeq 99\%$	3.052×10^{-5}

TABLE 7.3: Maximum resolution level, number of grid points and compression ratio with respect to the relative wavelet threshold parameters, with $p = 60$ bar and $\chi^0_{max} = 200\ s^{-1}$.

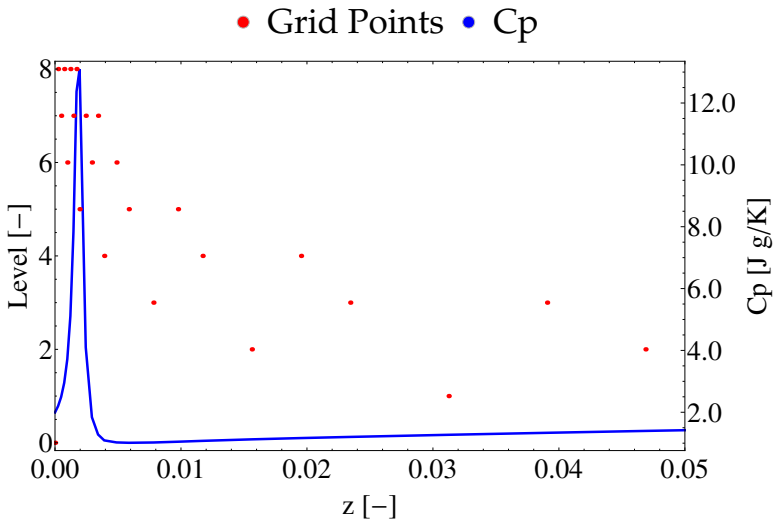


FIGURE 7.7: c_p and grid points at different resolution levels for $z = 0 - 0.05$; the wavelet threshold parameter is $\varepsilon_r = 10^{-3}$.

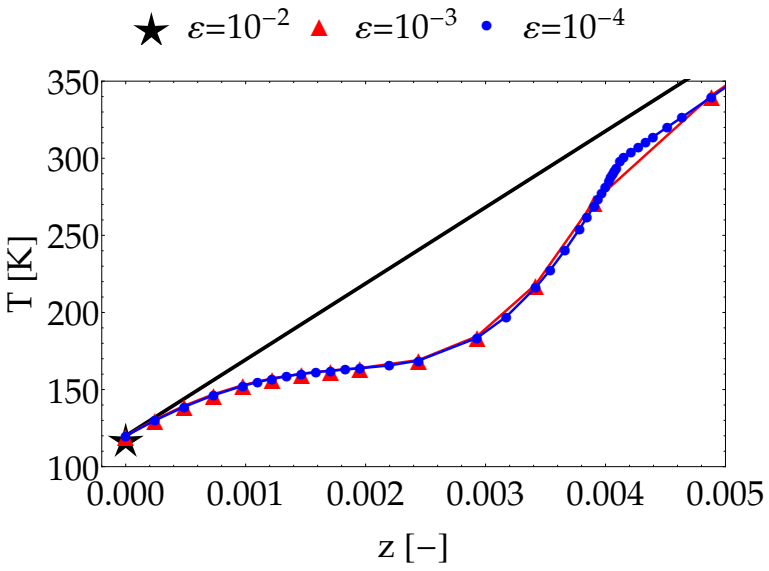


FIGURE 7.8: Temperature with respect to the mixture fraction for three threshold parameters, with $p = 60$ bar, $\chi^0_{max} = 200$ s^{-1}.

The pressure effects on steady state solutions have been investigated in the range $p = 60 - 160$ bar, with $\chi^0_{max} = 100$ s^{-1}. Sixth order wavelet functions and a threshold equal to $\varepsilon_r = 10^{-3}$ are again used for the wavelet representation. By increasing the pressure, an upward trend for maximum temperature and CO concentration can be observed in Figs. (7.9)-(7.10). Table (7.4) shows the variation of the maximum resolution with respect to pressure: higher values of the compression ratio are reached for $p = 60$ bar. This is due to a localized-high temperature variation in the vicinity of $z = 0$, that the wavelet-based method is able to accurately capture: the largest number of grid points is concentrated near $z = 0$.

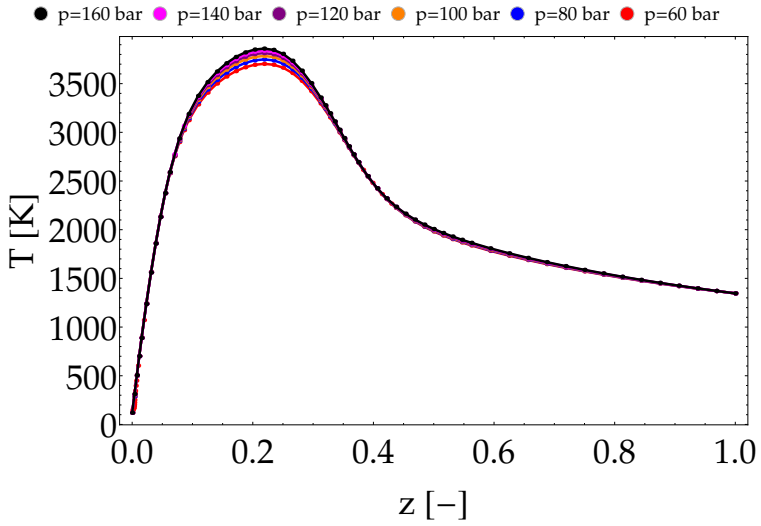


FIGURE 7.9: Temperature with respect to the mixture fraction in the pressure range $p = 60 - 160$ bar, with $\chi^0_{max} = 200\ \text{s}^{-1}$; the threshold parameter is $\varepsilon_r = 10^{-3}$ and the wavelet order is 6.

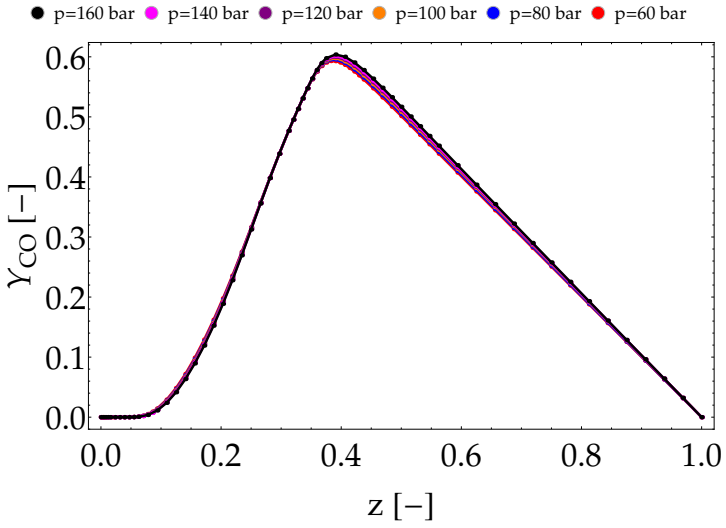


FIGURE 7.10: Y_{CO} with respect to the mixture fraction in the pressure range $p = 60 - 160$ bar, with $\chi^0_{max} = 200\ \text{s}^{-1}$; the threshold parameter is $\varepsilon_r = 10^{-3}$ and the wavelet order is 6.

p (bar)	J	N_T	π_w
60	8	80	$\simeq 98\%$
80	4	63	$\simeq 68\%$
100	4	63	$\simeq 68\%$
120	4	63	$\simeq 68\%$
140	4	63	$\simeq 68\%$
160	4	61	$\simeq 68\%$

TABLE 7.4: Maximum resolution level, number of grid points and compression ratio with respect to pressure.

7.5 Suggestions for future work

The results obtained in section (7.4) represent the preliminary step to build the multi-dimensional tabulated flamelets. This section of the chapter has the aim to show a possible strategy to build the multi-dimensional tabulated flamelets using WAMR. The so obtained library will play a pivotal role in the numerical simulation of turbulent diffusive flames in high pressure combustion devices.

Different strategies can be implemented. On one hand, the tabulated flamelets can be contructed by exploiting the just known capability of the wavelet-based collocation method to work in a multi-dimensional context, making use of higher-dimensional bases functions. Mathematical details of how WAMR works in a multi-d context can be found in the work of Grenga [16], with practical applications. On the other hand, a simple procedure can be implemented, by repeatedly using WAMR in the one-dimensional context. The next section is dedicated to the detailed explanation of the latter strategy.

7.5.1 Multi-dimensional tabulated flamelets

The main idea is to generate the table in the $p - \chi^0_{max}$ plan, where each point identifies fixed values of pressure and scalar dissipation rate through which simulations can be performed. A simulation consists in the time integration of the flamelet equations (6.1)-(6.2) with DVODE. Possible theoretical guidelines to build the tabulated flamelets are here summarized. The basic idea to build the table is to take constant values of pressure or scalar dissipation rate and to perform an adaptive refinement along the scalar dissipation rate or pressure direction, respectively. To explain the procedure, in this context the pressure value is fixed. Therefore, the adaptive refinement through the wavelet interpolation will be performed along the scalar dissipation rate direction.

The base grid

The first step for the table construction is the identification of a base grid in the $p - \chi^0_{max}$ plan, composed only of essential points. An example is shown in Fig. (7.11a). It has been obtained through preliminary simulations for the range $p = 40 - 180$ bar and $\chi^0_{max} = 200 - 4000\ \mathrm{s}^{-1}$. Each blue point is associated with a steady state solution (represented by the temperature and mass fractions trends along the mixture fraction) obtained for a test performed with assigned values of

pressure and scalar dissipation rate. As an example, in Fig. (7.12) four simulations performed at four points of the base grid are considered, namely:

- Simulation #1 at $p_1 = p^* = 40$ bar and $\chi^0_{max} = \chi^0_{max_1} = 2575$ s^{-1},
- Simulation #2 at $p_2 = p^* = 40$ bar and $\chi^0_{max} = \chi^0_{max_2} = 2100$ s^{-1},
- Simulation #4 at $p_4 = p^* = 40$ bar and $\chi^0_{max} = \chi^0_{max_4} = 1625$ s^{-1},
- Simulation #5 at $p_5 = p^* = 40$ bar and $\chi^0_{max} = \chi^0_{max_5} = 1150$ s^{-1},

where only the pressure is constant. The associated steady state solutions are:

- Simulation #1: $T^{ss}_1(z, p^*, \chi^0_{max_1})$ and $Y^{ss}_{1,i}(z, p^*, \chi^0_{max_1}) \quad i = 1, \dots, N_s \quad z = z_1, \dots, z_{N_{T_1}}$,
- Simulation #2: $T^{ss}_2(z, p^*, \chi^0_{max_2})$ and $Y^{ss}_{2,i}(z, p^*, \chi^0_{max_2}) \quad i = 1, \dots, N_s \quad z = z_1, \dots, z_{N_{T_2}}$,
- Simulation #4: $T^{ss}_4(z, p^*, \chi^0_{max_4})$ and $Y^{ss}_{4,i}(z, p^*, \chi^0_{max_4}) \quad i = 1, \dots, N_s \quad z = z_1, \dots, z_{N_{T_4}}$,
- Simulation #5: $T^{ss}_5(z, p^*, \chi^0_{max_5})$ and $Y^{ss}_{5,i}(z, p^*, \chi^0_{max_5}) \quad i = 1, \dots, N_s \quad z = z_1, \dots, z_{N_{T_5}}$.

Note that the steady state solutions are built on adaptive mixture fraction grids, characterized by different number of points.

The neighbouring points

Figure (7.11b) represents the neighbouring points (red stars) in the $p - \chi^0_{max}$ plan associated with the finer resolution level, that is added in a dyadic way, namely in the middle of two existing points of the base grid. Looking at Fig. (7.12), the neighbouring point represented by the red star is considered:

- Simulation #3 at $p_3 = p^* = 40$ bar and $\chi^0_{max} = \chi^0_{max_3} = 1862.5$ s^{-1},

and the steady state solution is:

- Simulation #3: $T^{ss}_3(z, p^*, \chi^0_{max_3})$ and $Y^{ss}_{3,i}(z, p^*, \chi^0_{max_3}) \quad i = 1, \dots, N_s \quad z = z_1, \dots, z_{N_{T_3}}$.

The wavelet interpolation

The steady state solution on the sparse grid #3 can be approximated with the wavelet interpolation. The extension of the interpolation stencil depends on the chosen wavelet order. If fourth order wavelet functions were considered, the interpolation stencil would be composed by 4 points. This means that, to evaluate the approximated function at a fixed mixture fraction value $z = \bar{z}$ belonging to the sparse grid #3 composed of N_{T_3} grid points, namely $\widetilde{f_3}(\bar{z}, p^*, \chi^0_{max_3})$ (where f plays the role of the temperature and/or mass fractions in the flamelet problem), four functions values are needed at the coarser resolution level, namely $f_1(\bar{z}, p^*, \chi^0_{max_1})$, $f_2(\bar{z}, p^*, \chi^0_{max_2})$, $f_4(\bar{z}, p^*, \chi^0_{max_4})$ and $f_5(\bar{z}, p^*, \chi^0_{max_5})$. If some functions values were not available for $z = \bar{z}$ because of the lack of this grid point on the corresponding mixture fraction grid, they should be found and reconstructed

through the wavelet interpolation performed along the local mixture fraction coordinate. The approximated temperature and mass fraction at $z = \bar{z}$ are $\widetilde{T}_3^{ss}(\bar{z}, p^*, \chi^0_{max_3})$ and $\widetilde{Y}^{ss}_{3,i}(\bar{z}, p^*, \chi^0_{max_3})$ with $i = 1, \ldots, N_s$. Repeating the interpolation procedure for all the mixture fraction values, namely $z = z_1, \ldots, z_{N_{T_3}}$, the approximated temperature and mass fractions trends along z are found:

- Simulation #3: $\widetilde{T}_3^{ss}(z, p^*, \chi^0_{max_3})$ and $\widetilde{Y}^{ss}_{3,i}(z, p^*, \chi^0_{max_3})$, $i = 1, \ldots, N_s$, $z = z_1, \ldots, z_{N_{T_3}}$.

To evaluate the error introduced by the wavelet interpolation, the infinity norm can be chosen as evaluation tool:

$$\left\| \frac{T^{ss}_3(z, p^*, \chi^0_{max_3}) - \widetilde{T}_3^{ss}(z, p^*, \chi^0_{max_3})}{T^{ss}_3(z, p^*, \chi^0_{max_3})} \right\|_\infty \leq \varepsilon_{T_3} \quad z = z_1, \ldots, z_{N_{T_3}}, \tag{7.28}$$

$$\left\| \frac{Y^{ss}_{3,i}(z, p^*, \chi^0_{max_3}) - \widetilde{Y}^{ss}_{3,i}(z, p^*, \chi^0_{max_3})}{Y^{ss}_{3,i}(z, p^*, \chi^0_{max_3})} \right\|_\infty \leq \varepsilon_{Y_{3,i}} \quad i = 1, \ldots, N_s \quad z = z_1, \ldots, z_{N_{T_3}}, \tag{7.29}$$

where, ε_{T_3} and $\varepsilon_{Y_{3,i}}$ are scaled wavelet threshold parameters (Eq. (2.67)). To promote the neighbouring point of Fig. (7.12) to new essential point, at least one of the $N_s + 1$ inequalities (7.28)-(7.29) must be unsatisfied, otherwise the point becomes a non-essential point.

The theoretical considerations described for the sole neighbouring point of Fig. (7.12) are valid for all the neighbouring points represented in Fig. (7.11b).
In the vicinity of the new essential points, new neighbouring points are added and new resolution levels are introduced. The wavelet adaptive refinement is repeated until no essential points are found at the current resolution level.

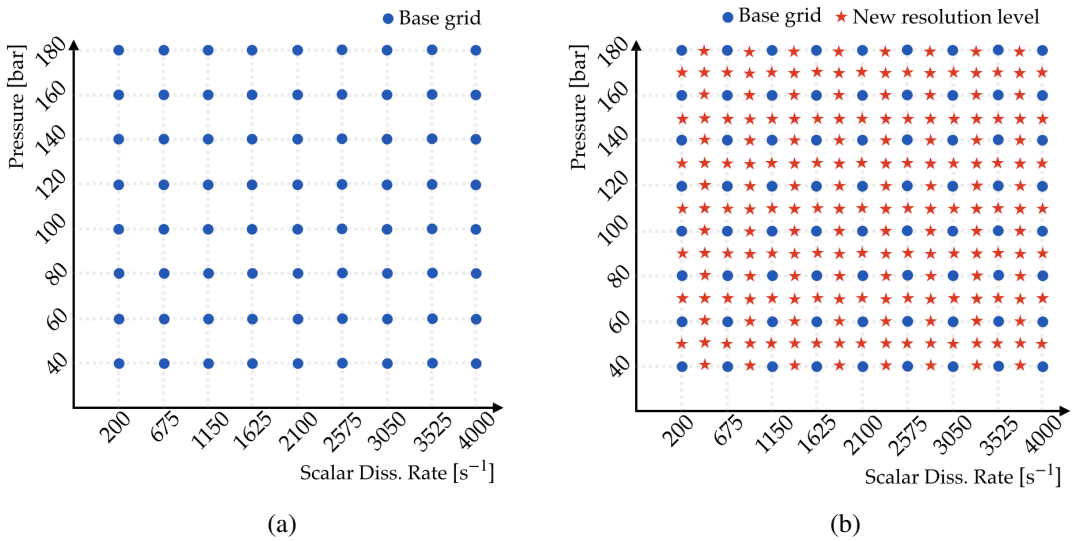


FIGURE 7.11: Full grid in the pressure - scalar dissipation rate plan for the first (base) and second resolution levels. The blue points are essential points, while the red stars represent neighbouring points that will be classified as essential or non-essential points after the wavelet analysis.

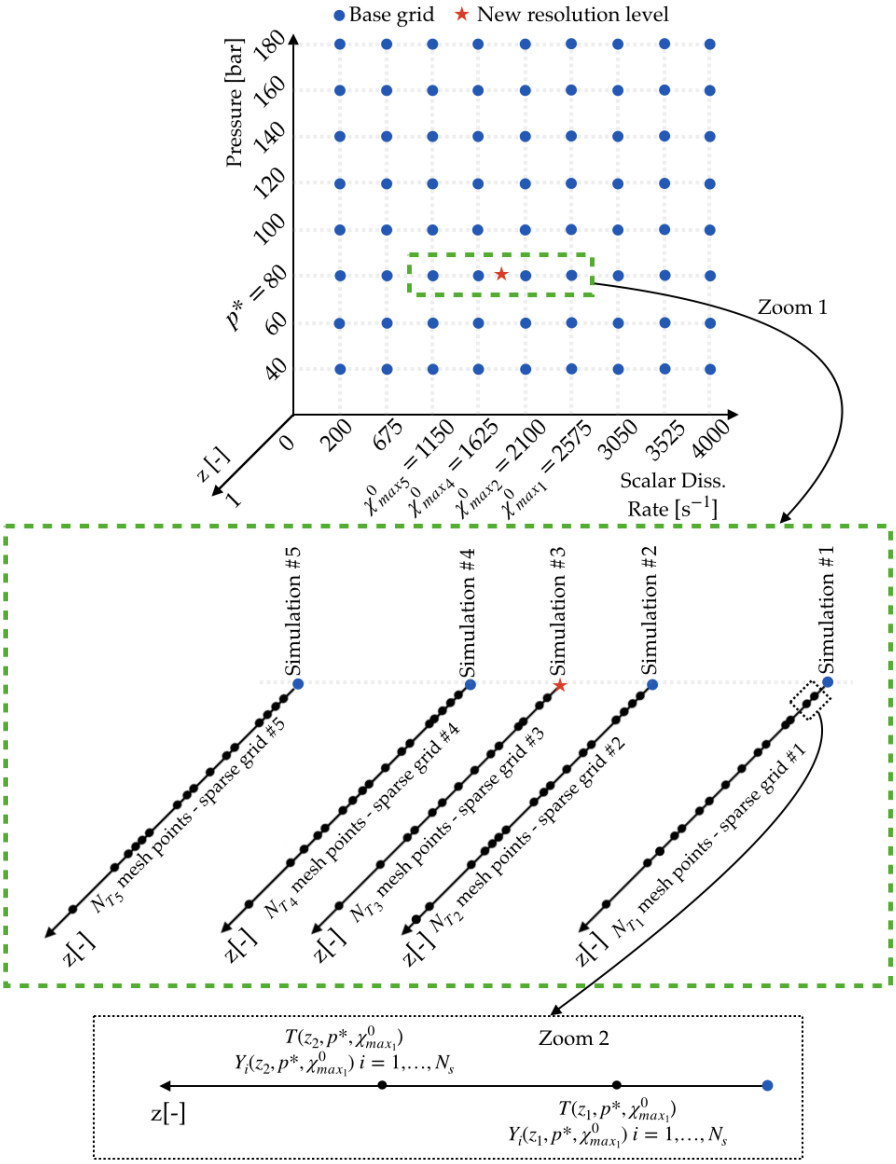


FIGURE 7.12: Simulations associated with four points of the base grid and a sole neighbouring point represented by the red star.

CHAPTER 8

Conclusions

The coupling of WAMR and the G-Scheme is proposed with the goal of achieving an adaptive model reduction in space and time. The mesh refinement, performed by WAMR, allows to obtain a reduction of DoFs with respect to those ones required by a reference uniform grid ensuring the same accuracy in space. The model reduction performed by the G-Scheme allows to identify the active equations to be integrated in time. To quantify the compression degrees, the compression ratios π_w, π_{gs} and π_O are introduced.

The Elezgaray-Arneodo reaction diffusion system is deeply studied with several purposes. First, the good implementation of WAMR is tested by studying the steady state of the system. Various tests are made, using different wavelet threshold parameters and DVODE as time integrator. The tests show that the errors between the adaptive solutions and the reference ones, evaluated in terms of infinity norm, are always lower than the prescribed accuracies in space.

A different system behaviour - namely, the periodic bursting - is studied with the aim to check the capability of the coupled scheme WAMR/G-Scheme to generate accurate time varying solutions with respect to reference solutions where uniform grids and a standard time integrator - DVODE - are used. Looking at one limit cycle, the tests show that an excellent agreement between the adaptive time varying solutions and the reference ones is achieved. Moreover, the performance of the solver WAMR/G-Scheme, evaluated in terms of computational cost required to complete a single limit cycle, are deeply investigated. As expected, the coupling between WAMR and the G-Scheme allows to greatly reduce the computational cost with respect to the case where the G-Scheme is employed on uniform grids: the workload goes from $\approx$ 1411 s (for the solver where the G-Scheme and uniform grid are employed) to $\approx$ 6 s (for the solver where WAMR and the G-Scheme are employed). Moreover, further reduction of the workload is obtain by reusing the basis vectors ($\approx$ 3.5 s for WAMR, G-Scheme with reuse of the basis), with minimal loss of accuracy with respect to the reference case (DVODE and uniform grid). The frequency of the reuse of the basis vectors is particularly high if the system behaviour is quite linear: by tracking the time variation of a single eigenvalue of the Jacobian matrix, it is shown that a nonlinear system behaviour correspond to a significant variation of the mode shapes associated with the selected eigenvalue.

Other tests are made starting from the simulation where WAMR, the G-Scheme and the reuse option are used, because of the just known capability of this combined solver to generate accurate solutions with a minimal workload. The relative wavelet threshold parameter is varied from $\varepsilon_r = 10^{-3}$ to $\varepsilon_r = 10^{-6}$. Simultaneously, the G-Scheme relative tolerances are changed too. As expected, if a larger accuracy is required - namely lower ε_r - the total CPU time associated with the solver WAMR/G-Scheme (with reuse of the basis) sharply increases. In particular, one has 3.571 s for $\varepsilon_r = 10^{-3}$ and 1919.129 s for $\varepsilon_r = 10^{-6}$.

In the last part of the work, several tests are made with the aim to compare the G-Scheme framework and the AIM approach. The importance to use a time varying tail as calculated by the G-Scheme theory is stressed and proved: the time varying solutions over one limit cycle are extremely accurate with respect to a reference time varying solution, obtained with a time varying tail dimension. About the performance one has that for the more conservative test where the tail dimension is $T = 73$, the total CPU time and the integration time steps are 102.438 s and 2739, respectively, while for the case where the tail dimension is time varying, one has 6.065 s and 101, respectively.

As a more complex reaction-diffusion system, the unsteady flamelets model for non-premixed combustion are firstly studied for pressure and temperature values far from the supercritical conditions. The coupled scheme WAMR/G-Scheme is used to generate adaptive solutions. The comparison with reference time varying solutions - built on uniform grid and integrating the flamelet equations with DVODE - is proposed: an excellent agreement can been clearly appreciated. The evaluation of the compression ratios shows that the G-Scheme and WAMR efficiency is minimum during the transient phase, where more DoFs are required to better described the temperature and concentrations variation in time. The eigenvalues analysis of the Jacobian matrix is proposed, identifying the active, fast and slow subspaces: the number of active modes is equal to one when the steady state is reached, according to the theory. The performance of the coupled scheme WAMR/G-Scheme is evaluated considering also an OS technique. As expected, strong advantages to use the OS technique can be obtained by evaluating the total computational cost required to perform the adaptive mesh refinement and the G-Scheme model reduction: looking at the orders of magnitude, one has that the Log_{10}(CPU Time) goes from 4.602 for the most expensive case of WAMR/G-Scheme without reuse of the basis vectors to 2.99 for the cheapest case of the G-Scheme is used in the OS technique.

Finally, the WAMR framework is used to generate steady state solutions of the flamelet equations in near-critical and supercritical conditions, using a detailed kinetic mechanism. To well describe the fluid behaviour at these conditions, the three-parameters RK-PR EoS is adopted. The density and c_p of the methane at different pressures and temperatures are evaluated through the RK-PR EoS. These results are in good agreement with reference data. By fixing the pressure ($p = 60$ bar), the scalar dissipation rate ($\chi^0_{max} = 200\ \mathrm{s}^{-1}$) and the threshold parameter ($\varepsilon_r = 10^{-3}$), the capability of the wavelet method to well capture strong variations of the solutions is shown: grid points are added only where higher resolution levels are required to ensure the achievement of the prescribed accuracy. The excellent capability of the wavelet compression with respect to

a reference uniform grid having the same minimal spacing is shown evaluating the compression degree, equal to $\pi_w = 98\%$. The evaluation of the thermodynamic properties at grid points in near-critical and supercritical conditions is particularly expensive. The associated computational time represents the 95% of the total computational time requested for the RHS evaluation at each grid point. By using dynamically adaptive grids the number of DoFs is greatly reduce with respect to a reference uniform grid having the same minimal spacing. As a result, the time integration of the flamelet equations results much less expensive. By increasing the values of the threshold parameters from $\varepsilon_r = 10^{-2}$ to $\varepsilon_r = 10^{-4}$, higher refinement levels are reached, more grid points are added and the compression ratios are larger. Furthermore, smaller solution variations are well captured with smaller threshold parameters. Finally, the pressure effects on steady state solutions are investigated, fixing $\varepsilon_r = 10^{-3}$, $\chi^0_{max} = 100$ s^{-1} and varying the pressure from $p = 60 - 160$ bar. Higher resolution levels are only reached in the vicinity of the critical conditions of the mixture, i.e. where the c_p variations are larger. Note that this work represents the first step to build the multi-dimensional tabulated flamelets through the use of the wavelet-based collocation method. They will be obtained through the simulation at different conditions (e.g., pressure and scalar dissipation rate). The wavelet method will be also used to determine if it is necessary to perform the simulation for given p and χ^0_{max} in order to obtain a multi-dimensional table with the defined accuracy. Preliminary results have been just obtained for the range $p = 40 - 180$ bar and $\chi^0_{max} = 200 - 4000$ s^{-1}.

Bibliography

[1] A. Adrover, F. Creta, M. Giona, and M. Valorani. “Stretching-based diagnostics and reduction of chemical kinetic models with diffusion”. In: *Journal of Computational Physics* 225.2 (2007), pp. 1442 –1471.

[2] Maryam Akram, Malik Hassanaly, and Venkatramanan Raman. “Reduced Description of Dynamical Systems by Approximate Inertial Manifolds”. In: *AIAA Scitech 2019 Forum*.

[3] S. Bertoluzza, G. Naldi, and J. C. Ravel. “Wavelet Methods for the Numerical Solution of Boundary Value Problems on the Interval”. In: *Wavelets: Theory, Algorithms,and Applications*. Ed. by C. K. Chui et al. et al. Vol. 5. Academic Press Inc., 1994, pp. 425–448.

[4] S.R. Brill, T. Grenga, J.M. Powers, and S. Paolucci. “Automatic error estimation and verification using an adaptive wavelet method”. In: *11th World Congress on Computational Mechanics (WCCM XI)* (2014).

[5] Martín Cismondi and Jørgen Mollerup. “Development and application of a three-parameter RK-PR equation of state”. In: *Fluid Phase Equilibria* 232.1 (2005), pp. 74 –89.

[6] A. Cohen, I. Daubechies, and J. C. Feauveau. “Biorthogonal Bases of Compactly Supported Wavelets”. In: *Communications on Pure and Applied Mathematics* 45 (1992), pp. 485–560.

[7] I. Daubechies. *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[8] G. Deslauriers and S. Dubuc. “Symmetric Iterative Interpolation Processes”. In: *Constructive Approximation* 5 (1989), pp. 49–68.

[9] D. Donoho. *Interpolating Wavelet Transform*. Report. Department of Statistics, Stanford University, 1992.

[10] J. Elezgaray and A. Arneodo. “Crisis-induced intermittent bursting in reaction-diffusion chemical systems”. In: *Phys. Rev. Lett.* 68 (5 1992), pp. 714–717.

[11] Juan Elezgaray and A Arneodo. “Modeling reaction-diffusion pattern formation in the Couette flow reactor”. In: *Chemical Physics - CHEM PHYS* 95 (July 1991), pp. 323–350.

[12] Mark O. McLinden Eric W. Lemmon and Daniel G. Friend. *Thermophysical Properties of Fluid Systems*. Eds. P.J. Linstrom and W.G. Mallard. NIST Chemistry WebBook, NIST Standard Reference Database Number 69. National Institute of Standards and Technology, Gaithersburg MD, 20899, 2019.

[13] Sharath S. Girimaji. “On the modeling of scalar diffusion in isotropic turbulence”. In: *Physics of Fluids* 4 (1992), pp. 2529 –2537.

[14] Stefan Goedecker. *Wavelets and their application for the solution of partial differential equations in physics*. Cahiers de physique, 1998.

[15] Dimitris A. Goussis, Mauro Valorani, Francesco Creta, and Habib N. Najm. "Inertial manifolds with CSP". In: *Computational Fluid and Solid Mechanics 2003*. Ed. by K.J. Bathe. Oxford: Elsevier Science Ltd, 2003, pp. 1951 –1954.

[16] T. Grenga. "Numerical solutions of multi-dimensional compressible reactive flow using a parallel wavelet adaptive multi-resolution method". PhD thesis. University of Notre Dame, Notre Dame, IN 46556, 2015.

[17] B. Jawerth and W. Sweldens. "An Overview of Wavelet Based Multiresolution Analyses". In: *SIAM Review* 36 (1994), pp. 337–412.

[18] Seong-Ku Kim, Hwan-Seok Choi, and Yongmo Kim. "Thermodynamic modeling based on a generalized cubic equation of state for kerosene/LOx rocket combustion". In: *Combustion and Flame* 159.3 (2012), pp. 1351 –1365.

[19] Taehoon Kim, Yongmo Kim, and Seong-Ku Kim. "Numerical analysis of gaseous hydrogen/liquid oxygen flamelet at supercritical pressures". In: *International Journal of Hydrogen Energy* 36.10 (2011), pp. 6303 –6316.

[20] S. H. Lam. "Using CSP to Understand Complex Chemical Kinetics". In: *Combustion Science and Technology* 89.5-6 (1993), pp. 375–404.

[21] S. H. Lam and D. A. Goussis. "The CSP method for simplifying kinetics". In: *International Journal of Chemical Kinetics* 26.4 (1994), pp. 461–486.

[22] S.H. Lam and D.A. Goussis. "Computational singular perturbation; Theory and applications". In: *Report 1986-MAE* (1991).

[23] S.H. Lam and D.A. Goussis. "Reduced kinetic mechanisms and asymptotic approximations for methane-air flames : a topical volume". In: *Report 1800-MAE* (1991), pp. 303 –306.

[24] S.H. Lam and D.A. Goussis. "The analytic foundation of CSP". In: *Report 1800-MAE* (1991).

[25] S.H. Lam and D.A. Goussis. "Understanding complex chemical kinetics with computational singular perturbation". In: *Symposium (International) on Combustion* 22.1 (1989), pp. 931 –941.

[26] Pasquale Eduardo Lapenna, Pietro Paolo Ciottoli, and Francesco Creta. "Unsteady Non-Premixed Methane/Oxygen Flame Structures at Supercritical Pressures". In: *Combustion Science and Technology* 189.12 (2017), pp. 2056–2082.

[27] J. C. Lee, H. N. Najm, S. Lefantzi, J. Ray, M. Frenklach, M. Valorani, and D. A. Goussis. "A CSP and tabulation-based adaptive chemistry model". In: *Combustion Theory and Modelling* 11.1 (2007), pp. 73–102.

[28] Juan Li, Zhenwei Zhao, Andrei Kazakov, Marcos Chaos, Frederick Dryer, and James J. Scire Jr. "A comprehensive kinetic mechanism for CO, CH2O, and CH3OH combustion". In: *International Journal of Chemical Kinetics* 39 (Mar. 2007), pp. 109 –136.

[29] Tianfeng Lu, Yiguang Ju, and Chung K Law. "Complex CSP for chemistry reduction and analysis". In: *Combustion and Flame* 126.1 (2001), pp. 1445 –1455.

[30] Tianfeng Lu and Chung K. Law. "A criterion based on computational singular perturbation for the identification of quasi steady state species: A reduced mechanism for methane oxidation with NO chemistry". In: *Combustion and Flame* 154.4 (2008), pp. 761 –774.

[31] Valorani M., Ciottoli P.P., and Malpica Galassi R. et al. "Enhancements of the G-Scheme Framework". In: *Flow Turbulence Combust* (2018).

[32] Y. Meyer. *Wavelet and Operators: Volume 1*. Cambridge University Press, 1995.

[33] Richard S. Miller, Kenneth G. Harstad, and Josette Bellan. "Direct numerical simulations of supercritical fluid mixing layers applied to heptane–nitrogen". In: *Journal of Fluid Mechanics* 461 (2002), pp. 411–412.

[34] *MPI: A Message-Passing Interface Standard*. URL http://www.mpi-forum.org/docs/mpi-11-html/mpi-report.html. June 1995.

[35] Samuel Paolucci, Zachary J. Zikoski, and Temistocle Grenga. "WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. Part II. The parallel algorithm". In: *Journal of Computational Physics* 272 (2014), pp. 842 –864.

[36] Samuel Paolucci, Zachary J. Zikoski, and Damrongsak Wirasaet. "WAMR: An adaptive wavelet method for the simulation of compressible reacting flow. Part I. Accuracy and efficiency of algorithm". In: *Journal of Computational Physics* 272 (2014), pp. 814 –841.

[37] Eric Petersen and Ronald K. Hanson. "Reduced Kinetics Mechanisms for Ram Accelerator Combustion". In: *Journal of Propulsion and Power - J PROPUL POWER* 15 (July 1999), pp. 591–600.

[38] H. Pitsch and N. Peters. "A Consistent Flamelet Formulation for Non-Premixed Combustion Considering Differential Diffusion Effects". In: *Combustion and Flame* 114.1 (1998), pp. 26 –40.

[39] Jens Prager, Habib N. Najm, Mauro Valorani, and Dimitris A. Goussis. "Skeletal mechanism generation with CSP and validation for premixed n-heptane flames". In: *Proceedings of the Combustion Institute* 32.1 (2009), pp. 509 –517.

[40] Y. A. Rastigejev. "Multiscale Computations with a Wavelet Adaptive Algorithm". PhD thesis. University of Notre Dame, Notre Dame, IN 46556, 2002.

[41] Maher Salloum, Alen Alexanderian, Olivier P. Le Maître, Habib N. Najm, and Omar M. Knio. "Simplified CSP analysis of a stiff stochastic ODE system". In: *Computer Methods in Applied Mechanics and Engineering* 217-220 (2012), pp. 121 –138.

[42] Bruno Sportisse. "An Analysis of Operator Splitting Techniques in the Stiff Case". In: *Journal of Computational Physics* 161.1 (2000), pp. 140 –168.

[43] Gilbert Strang. "On the Construction and Comparison of Difference Schemes". In: *SIAM Journal on Numerical Analysis* 5.3 (1968), pp. 506–517.

[44] Weiqi Sun, Temistocle Grenga, and Yiguang Ju. "Hybrid Multi-Timescale and G-Scheme Method for Efficient Modeling with Detailed Chemical Kinetics". In: *Proceedings of the 55th AIAA Aerospace Sciences Meeting*. Jan. 2017.

[45] Weiqi Sun, Liang Wang, Temistocle Grenga, and Yiguang Ju. "Comparative study of hybrid multi-timescale and g-scheme methods for MARCS with detailed chemical kinetics". In: *Proceedings of the 55th AIAA Aerospace Sciences Meeting*. Jan. 2017.

[46] Weiqi Sun, Liang Wang, Temistocle Grenga, and Yiguang Ju. "Development of a Multiscale Adaptive Reduced Chemistry Solver (MARCS) for Computationally Efficient Combustion Simulations". In: *Proceedings of the 26th International Colloquium on the Dynamics of Explosions and Reactive Systems*. July 2017.

[47] M. Valorani, D. A. Goussis, and H. N. Najm. "Using CSP to analyze computed reacting flows". In: *8th SIAM International Conference on Numerical Combustion* (2000).

[48] M. Valorani, D. A. Goussis, and H. N. Najm. "Using CSP to analyze computed reacting flows". In: *Eighth International Conference on Numerical Combustion* (Mar. 2000).

[49] M. Valorani and S. Paolucci. "The G-Scheme: A framework for multi-scale adaptive model reduction". In: *Journal of Computational Physics* 228.13 (2009), pp. 4665 –4701.

[50] Mauro Valorani, Habib N. Najm, and Dimitris A. Goussis. "CSP analysis of a transient flame-vortex interaction: time scales and manifolds". In: *Combustion and Flame* 134.1 (2003), pp. 35 –53.

[51] Mauro Valorani, Francesco Creta, Dimitris A. Goussis, Jeremiah C. Lee, and Habib N. Najm. "An automatic procedure for the simplification of chemical kinetic mechanisms based on CSP". In: *Combustion and Flame* 146.1 (2006), pp. 29 –51.

[52] Mauro Valorani, Samuel Paolucci, Emanuele Martelli, Temistocle Grenga, and Pietro P. Ciottoli. "Dynamical system analysis of ignition phenomena using the Tangential Stretching Rate concept". In: *Combustion and Flame* 162.8 (2015), pp. 2963 –2990.

[53] Mauro Valorani, Dimitris A. Goussis, Francesco Creta, and Habib N. Najm. "Higher order corrections in the approximation of low-dimensional manifolds and the construction of simplified problems with the CSP method". In: *Journal of Computational Physics* 209.2 (2005), pp. 754 –786.

[54] Mauro Valorani, Francesco Creta, Filippo Donato, Habib N. Najm, and Dimitris A. Goussis. "Skeletal mechanism generation and analysis for n-heptane with CSP". In: *Proceedings of the Combustion Institute* 31.1 (2007), pp. 483 –490.

[55] D. Wirasaet. "Numerical solutions of multidimensional partial differential equations using an adaptive wavelet method". PhD thesis. University of Notre Dame, Notre Dame, IN 46556, 2008.

[56] Antonios Zagaris, Hans G. Kaper, and Tasso J. Kaper. *Analysis of the CSP Reduction Method for Chemical Kinetics*. 2003.