ISAS - INTERNATIONAL SCHOOL FOR ADVANCED STUDIES

New Aspects of Multilevel Methods and Applications

Thesis submitted for the degree
"Doctor Philosophæ"

Candidate:
Alfio Borzì

Supervisor:
Dr. Antonio Lanza

October 1993

TRIESTE

SISSA - SCUOLA INTERNAZIONALE SUPERIORE STUDI AVANZATI
TRIESTE
Strada Costiera 11

New Aspects of Multilevel Methods and Applications

Thesis submitted for the degree
"Doctor Philosophæ"

Candidate:
Alfio Borzì

Supervisor:
Dr. Antonio Lanza

October 1993

ai miei genitori

Agata e Luigi

Contents

Acknowledgements

I wish to express my gratitude to Dr. A. Lanza who introduced me into the fascinating field of multilevel research. His excellent guidance acquainted me with a multitude of new techniques and ideas which constitute the wonders of multilevel methods.

I would like also to thank Prof. A. Brandt and Prof. W. Hackbusch who patiently read the reports of my study and made numerous comments and suggestions.

In addition, thanks to the Mathematical Physics Sector at SISSA which provided continuous encouragement and helped me to develop my own field of research.

Finally, I wish to thank Prof. K.W. Morton who gives me the possibility to continue my research on multilevel methods after this thesis.

Introduction

It is very common, in scientific research, to solve mathematical problems of high complexity. This fact stimulates the development of analytical methods to provide the solution of these problems. However, the problem considered could be rather difficult to solve by these methods. Therefore in most cases one should resort to numerical methods which give only partial information of the solution of the problem at hand. Nevertheless, it is often this information which is needed and, sometimes, enough for theoretical or practical purposes.

For this reason, various numerical algorithms have been developed. They produce the solution of a given problem in numerical form, in correspondence to a required accuracy. Clearly, together with the actual algorithm, many theoretical questions often arise in any computational approach. For instance, one should prove that the resulting solution approximates (in some sense) the exact solution. Also, one should provide a method which is "optimal", i.e. an algorithm for which the computational work required is taken at a minimum (proportional to the number of unknowns). This is necessary since there are problems which require a very large amount of computational time whose minimization is the only way to solve them even in nowadays computers. Behind these questions, though, there are many others. For example, the analysis of these solution procedures could lead to the formulation of new analytical tools for solving exactly the given problem.

In this thesis we will be concerned mainly with the second question, and describe the modern class of algorithms named *multilevel* (ML) methods [13, 37], which in a very large number of applications have been proved to be optimal in the above sense. Notice that, for example for the solution of linear systems of equations, two principal families of solvers existed: that is the the so-called *direct* and *iterative* methods [76]. These classes are of common use now, but for many important applications are not optimal. However, they played an important role also in the realization of the *multilevel strategy* [9, 10].

The basic concept of the multilevel approach is based on the following two remarks: *the amount of computational work should be proportional to the amount of real physical changes in the computed system* [13]. Also one should take into account that: *the solution*

of many problems is made of several components with different scales, which interact with each other [13]. Therefore, when one discretizes a continuous problem, one should require that the number of unknowns representing the solution should be proportional to the number of the physical features to be described. So, for example for a smooth function it is required the knowledge of only few of its values in order to be represented on the given domain, whereas high resolution is required for highly oscillating functions. As a consequence, to describe a solution of a given problem the interactive use of many scales of discretization would be suitable to resolve for all components of the solution. This is the essence of the multilevel strategy.

However, even if the above two concepts are well known, their contemporary implementation is not obvious and multilevel methods represent an example of a clear application of them. For this reason the ML approach has general applicability. In fact, the resulting ML algorithms have been formulated for many different problems of Mathematics and Physics, independently of the method of discretization used.

Already in the sixties R.P. Fedorenko [25, 26] developed the first multilevel scheme for the solution of the Poisson equation in a unit square. Since then, other russian mathematicians extended Fedorenko's idea to general elliptic boundary value problems with variable coefficients in the same domain [3]. However, the full efficiency of the ML approach was realized only after the works of A. Brandt [9, 10]. At the same time he introduced the so-called *full approximation storage* (FAS) scheme [10], which is a multilevel strategy to solve also non-linear problems. Another achievement in the formulation of ML methods was the *full multigrid* (FMG) [10] based on the systematic application of the nested iteration idea, that is to use coarser grids to obtain good initial approximations on finer grids.

In [10] the analysis of the efficiency of the multilevel methods is done by means of the so-called *local mode analysis* (LMA), then developed in [15]. This widely used theoretical approach provides sharp convergence estimates, but it requires some idealizing assumptions. Independently, another general ML convergence theory has been formulated by W. Hackbusch [34, 35]. This alternative theory has been developed in case of both *finite element* (FE) [58] and *finite difference* (FD) [29] approximation schemes for boundary

value problems in rectangular and non rectangular regions. This theory, however, seems not able to provide very sharp estimates.

In the first chapter we sketch only the fundamentals of LMA. Then we will use this investigative tool in chapter 3, to obtain very precise predictions of the convergence property of a FAS scheme designed to solve non linear integral equations.

The analytical tools described above served to construct and analyze the multilevel algorithms which extended to larger and larger field of applications. The first, historically, was to solve linear and non linear boundary value problems (scalar equations and systems) [9, 61, 10, 11, 32]. Then these applications have enlarged to consider eigenvalue [12, 33], bifurcation [59, 78], parabolic [18, 36, 80], and hyperbolic[1] [45] problems.

All these situations occur in *fluid dynamic* computations, the field of numerical investigation to which multilevel scientists dedicate their major efforts (see e.g. [39, 40, 53, 41, 43] and references therein). Also the ML method has been applied to several situations of relativistic astrophysics [16, 48, 49, 50]. In [16, 48] the multilevel method is properly defined to solve the stationary axisymmetric Einstein's equations, and this was applied to obtain sequences of equilibrium models for self-gravitating disks around black holes [49]. In order to extend the multilevel investigation for non stationary astrophysical problems, we begin study Burgers' equation in chapter 2 since it has been suggested [75] to be a good approximation for the study of the dynamics of the large scale structure of the universe.

In addition to partial differential equations (PDE), Fredholm's integral equations can also be efficiently solved by multilevel methods [38, 42, 17]. These schemes can be used to solve reformulated boundary value problems (see [37] and references therein) or for the fast solution of N-body problems [19, 4]. In particular, in order to investigate scattering theories in their ultra-violet limit, where they are supposed to correspond to a conformal field theory, we have developed a multilevel code, described in chapter 3, for the solution of the thermodynamic Bethe ansatz equations [5], which are a system of non linear integral equations. This is an example of application of ML schemes in field

[1] In this case, however, the purpose is not to define a multilevel method for *purely* hyperbolic problems since very efficient algorithms already exist, but to study mixed elliptic/hyperbolic problems.

theory. Another example are lattice field computations and optimization [21, 44, 20]. Especially in quantum electrodynamics and chromodynamics simulations the purpose of the multilevel approach is to accelerate Monte Carlo iterations [31, 52].

Though the multilevel methods are very efficient solvers, their construction requires expertise: *The only disadvantage seems to be the complex programming involved* [9]. In fact, for this reason a multilevel approach is still considered a specialized technique, even if simple versions of numerical ML codes are now available [68]. However, a multilevel library where the numerical codes can be used without expertise is still lacking. Therefore there is the attempt to formulate algebraic multilevel solvers, where the choice or construction of the many elements which compose a ML algorithm are done automatically by the code itself, following algebraic equations. This attempt has stimulated the research of multilevel methods for the resolution of pure algebraic problems [14, 71], and also to define blackbox solvers designed to handle specific discrete problems [23, 89]. Following this line of research, we have studied anew the algebraic features of a multilevel method [7]. Starting from the simple tridiagonal case, we have obtained the complete formulation of a twolevel procedure based only on the splitting matrix formalism, well known in the theory of matrix iterative procedures. With this result we are able to recover the standard multilevel algorithms which could be derived naturally from our formulation (see chapter 4). In addition to these results, we show how an algebraic approach could lead to the formulation of a new convergence theory which does not suffer of the limitations of the other existing theories mentioned above. Actually, we derive exact convergence estimates for the case when a multilevel method is applied to a tridiagonal system. Then we prove that this result extend to more general class of problems, and argue why these estimates should be valid also in the case of the standard multilevel approach. Finally, we obtain a confirmation of this analysis via numerical experiments.

Our splitting formulation of multilevel methods is fruitful of consequences. Another, for example, is that we are able to give a first example of an extension of the twolevel method for operator equations in Hilbert space [8] (see chapter 5).

Chapter 1

Multilevel Methods

1.1 Introduction

As said above two basic remarks constitute the essence of the multilevel idea. It was only after a careful study of iterative methods which allowed the concrete contemporary realization of these ideas. In fact in the earlier works [25, 26, 9, 32] on multilevel methods for elliptic difference equations, was showed that it is possible to define simple iterations which quickly converge with respect to the high frequency components of the solution error. We discuss this *smoothing property* that is utilized in the construction of ML algorithms by a simple example in the following section. Then we describe the twolevel and the multilevel method for solving linear equations. Therefore we introduce two transfer operators by means of which the solution processes of different levels may interact. Further we explain how to implement a multilevel method for the solution of non linear problems. Then, at the end of this chapter we will discuss the full multilevel method.

1.2 Multilevel Methods for Linear Problems

In the sections which follow, we present briefly the basic components used in the construction of a multilevel scheme. We start with the analysis of two standard iterative techniques: the Jacobi and the Gauss-Seidel (GS) schemes. These two methods are char-

acterized by a poor convergence rate. This is a common feature of standard iterative techniques, the trouble is related to the existence of residual error components with very different length scales. There are smooth components which can be approximated on coarse grids, but this conflicts with the high frequency components which must be approximated on finer grids. Then due to the local nature of the action of an iterative scheme only those components of the error with length scales comparable to the mesh size of the grid used (in case of finite-difference discretizations) are rapidly damped from one iteration to the next, leaving behind smooth, longer wave length errors which damp slowly and lead to slow convergence. The multilevel algorithm [10] by employing levels of grids of different mesh sizes, resolves all components of different scales, resulting in a rapid convergence rate. We describe this method in section 1.2.2 analyzing first a twolevel method, important for theoretical purposes, and then the multilevel scheme. Actually because the action of a suitable iteration leaves only smooth error components it is possible to represent them by the solution of a system of coarse defect (or residual) equations whose right hand sides are local averages of the fine grid defects. Once these equations are solved for the errors, the corrections are interpolated back to the fine grid to update the solution. Further the coarse equations can be solved recursively by combining relaxation sweeps with coarse correction involving coarser grids. This gives the multilevel cycle.

1.2.1 Iterative Methods and the Smoothing Property

Let us consider a large system of linear equations $Au = f$, in which the $n \times n$ matrix A is *sparse*, i.e., has relatively few non vanishing elements. This is a common occurrence, for example using finite difference or finite element approximation schemes for the discretization of elliptic problems. Because of its sparsity it is appropriate to use iterative methods. These are based on the definition of a recursive relation

$$u^{(\nu+1)} = Mu^{(\nu)} + Nf \ , \tag{1.1}$$

so that, beginning with an initial vector $u^{(0)}$, the sequence $u^{(\nu)}$, $\nu = 0, 1, \ldots$, converges to the exact solution $u = A^{-1}f$. In particular defining the solution error at the step ν as

$e^{(\nu)} = u^{(\nu)} - u$, iteration (1.1) is rewritten as $e^{(\nu+1)} = Me^{(\nu)}$. M is called the *iteration matrix.*

We can state the following convergence criterion based on the *spectral radius* $\rho(M)$ of the matrix M [76]

Theorem 1 *The method (1.1) converges if and only if* $\rho(M) < 1$.

In general, all such algorithms can be obtained through the *splitting* $A = B - C$, assuming that B is non singular. By putting $Bu^{(\nu+1)} - Cu^{(\nu)} = f$ and solving with respect to $u^{(\nu+1)}$ one obtains

$$u^{(\nu+1)} = B^{-1}Cu^{(\nu)} + B^{-1}f \ .$$

Let us denote with $D = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$, the diagonal part of the matrix A, and with $-L$ and $-U$ its strictly lower and upper part, respectively. Then we have $A = D - L - U$ and the choice $B := \frac{1}{\omega}D$ and $C = \frac{1}{\omega}[(1-\omega)D + \omega(L+U)]$, $0 < \omega \leq 1$, leads to the so called *damped* Jacobi iteration

$$u^{(\nu+1)} = (I - \omega D^{-1}A)u^{(\nu)} + \omega D^{-1}f \ . \tag{1.2}$$

Alternatively putting $B := D - L$ and $C := U$, one obtains the Gauss-Seidel iteration

$$u^{(\nu+1)} = (D-L)^{-1}Uu^{(\nu)} + (D-L)^{-1}f \ . \tag{1.3}$$

Later on we will denote the iteration matrices relative to (1.2) and (1.3) with $M_J(\omega)$ and M_{GS}, respectively.

To analyze the smoothing property of these iterations we introduce a simple model problem. We consider the finite difference approximation of a simple one-dimensional Dirichlet boundary value problem:

$$\begin{cases} -\frac{d^2u}{dx^2} = f(x) \ , & \text{in} \quad \Omega = (0,1) \\ u(x) = 0 \ , & \text{for} \quad x \in \Gamma = \{0,1\} \ . \end{cases} \tag{1.4}$$

In the finite difference formalism one chooses a *grid size* $h = \frac{1}{n+1}$, so that the corresponding grid consists of points $x_j = jh$, $j = 0, 1, \ldots, n+1$. Then a discretization scheme for the

second derivative at the point x_i is $h^{-2}[-u(x_{j-1})+2u(x_j)-u(x_{j+1})] = -u''(x_j)+O(h^2)$. Now setting $f_j^h = f(jh)$ and $u_j^h = u(jh)$ and denoting the column vectors $f^h = (f_j^h)_{j=1,n}$ and $u^h = (u_j^h)_{j=1,n}$, we obtain the following system of n equations

$$h^{-2}[-u_{j-1}^h + 2u_j^h - u_{j+1}^h] = f_j^h \quad , \quad j = 1,\ldots,n \quad . \tag{1.5}$$

This is a tridiagonal system and is well known that can be solved by direct methods. Nevertheless (1.5) provides a good example for the analysis of iterative schemes. For the moment let us omit the discretization parameter h and briefly denote (1.5) by $Au = f$.

Let us consider the eigenvalue problem for the damped Jacobi iteration, $M_J(\omega)v^k = \mu_k v^k$. This problem is simply solved and gives

$$v^k = (\sin(k\pi hj))_{j=1,n} \quad , \quad k = 1,\ldots,n \quad , \tag{1.6}$$

for the eigenvectors, and

$$\mu_k(\omega) = 1 - \omega(1-\cos(k\pi h)) \quad , \quad k = 1,\ldots,n \quad , \tag{1.7}$$

for the eigenvalues.

It is easy to see that $\rho(M_J(\omega)) < 1$, $0 < \omega \leq 1$ and, therefore, (1.2) is always convergent. In particular, for the simple Jacobi iteration ($\omega = 1$) we have $\rho(M_J(1)) = 1 - \frac{1}{2}\pi h^2 + O(h^4)$, which shows how the convergence of (1.2) deteriorates as $h \to 0$. However, the purpose of the iteration in a ML context is primarily to be a smoothing operator. In order to prove this property we need to distinguish *low* and *high* frequency eigenvectors.

- *low frequency* (LF) v^k with $1 \leq k < \frac{n}{2}$;
- *high frequency* (HF) v^k with $\frac{n}{2} \leq k \leq n$;

We now define the *smoothing factor* μ of $M_J(\omega)$ as the worst factor by which the amplitudes of HF components are damped per relaxation sweep:

$$\mu = \max\{|\mu_k|, \ \frac{n}{2} \leq k \leq n\} = \max\{1-\omega, \ |1-\omega(1+\cos(\pi h)|\} \leq \max\{1-\omega, \ |1-2\omega|\} \quad .$$

This inequality gives an optimal smoothing factor $\mu = 1/3$ for $\omega^* = 2/3$.

Now suppose to apply ν times the damped Jacobi iteration with $\omega = \omega^*$. The initial error $e^{(0)} = \sum_k e_k^{(0)} v^k$ will be damped accordingly to $e^{(\nu)} = M_J(\omega^*)^\nu e^{(0)}$. This implies for the Fourier coefficients that $e_k^{(\nu)} = \mu_k(\omega^*)^\nu e_k^{(0)}$. Therefore few steps of (1.2) will give $|e_k^{(\nu)}| << |e_k^{(0)}|$ for high frequencies. For this reason, although the global error decreases very slowly by iteration, it is smoothed very quickly and this process does not depend on h.

However the Jacobi iteration is used primarily for theoretical purposes, being easy to analyze. In practice other iterations are used (see e.g. [82]) which suppress the HF components of the error more efficiently. For example the Gauss-Seidel iteration 1.3. The smoothing property of this scheme is conveniently analyzed using the local mode analysis[1]. This is the most general tool for analyzing the ML process even thought it is based on certain idealized assumptions and simplifications. Regarding our model problem, first we must neglect boundaries and boundary conditions and consider the problem on an infinite grid $G^h = \{jh, \ j \in \mathbb{Z}\}$. Notice that on this grid only the Fourier components $e^{i\theta x/h}$ with $\theta \in (-\pi, \pi]$ are visible, i.e., there is no frequency $\theta_0 \in (-\pi, \pi]$ with $|\theta_0| < \theta$ such that $e^{i\theta_0 x/h} = e^{i\theta x/h}$, $x \in G^h$.

In LMA the notion of low and high frequency components on the grid G^h is related to the other grid which is introduced as coarse space, denoted by G^H. In this way $e^{i\theta x/h}$ is said to be an high frequency component, with respect to a coarse grid G^H, if its restriction (projection) to G^H is not visible there. Usually $H = 2h$ and the high frequencies are those with $\frac{\pi}{2} \leq |\theta| \leq \pi$.

We recall that the Gauss-Seidel scheme is based on the operator splitting $A = B - C$, where B an C are again difference operators. Then a relaxation sweep starting from an initial approximation $u^{(\nu)}$ produces a new approximation $u^{(\nu+1)}$ such that the corresponding errors satisfy

$$Be^{(\nu+1)}(x) - Ce^{(\nu)}(x) = 0 \ , \quad x \in G^h \ . \tag{1.8}$$

To analyze this iteration we define the errors in terms of their θ components $e^{(\nu)} = \sum_\theta \mathcal{E}_\theta^{(\nu)} e^{i\theta x/h}$ and $e^{(\nu+1)} = \sum_\theta \mathcal{E}_\theta^{(\nu+1)} e^{i\theta x/h}$. Where $\mathcal{E}_\theta^{(\nu)}$ and $\mathcal{E}_\theta^{(\nu+1)}$ denote the error ampli-

[1]In chapter 4 we analyze the convergence and smoothing properties of Jacobi and Gauss-Seidel iterations in a completely different way, without idealizing assumptions.

tudes of the θ component, before and after relaxation, respectively. Then (1.8) reduces to

$$(2 - e^{-i\theta})\mathcal{E}_\theta^{(\nu+1)} - e^{i\theta}\mathcal{E}_\theta^{(\nu)} = 0 \quad .$$

In this formalism the smoothing factor is defined by

$$\mu = \max\{|\frac{\mathcal{E}_\theta^{(\nu+1)}}{\mathcal{E}_\theta^{(\nu)}}| \ , \ \frac{\pi}{2} \leq \theta \leq \pi\} \ , \tag{1.9}$$

which does not depend on h but, on the relative size of the coarse grid. If we apply the local mode analysis to the Gauss-Seidel iteration used to solve our model problem obtain $\mu = 0.45$. Therefore this iteration is an "efficient smoother", i.e. it reduces high frequency error components by one order of magnitude in only a few relaxation sweeps.

1.2.2 The Two and Multilevel Methods

Let us suppose to apply ν_1 times an efficient smoother to (1.5). The resulting approximation $\tilde{u}^h$ will be affected by an error $\tilde{e}^h = \tilde{u}^h - u^h$ which is a smooth function. As a consequence $\tilde{e}^h$ can be well approximated on a "coarser" discretization space. Now we need to express this smooth error as a solution of a coarse problem, whose matrix A and right hand side will be now defined. For this purpose notice that because A^h is a difference operator of second order, also the *defect* $d^h = A^h\tilde{u}^h - f^h$ becomes a smooth function. Obviously the original equation $A^h u^h = f^h$ and the defect equation $A^h\tilde{e}^h = d^h$ are equivalent, and at first sight the solution of the latter equation appear to be no simpler than the former. Nevertheless, $\tilde{e}^h$ and d^h are smooth, hence we can find the solution error on a grid whose size is for example twice the previous one, $H = 2h$. Thus we define the function d^H on the coarse mesh as a restriction of the fine defect, that is by setting $d^H = Rd^h$, where R is a suitable *restriction* operator (for example the common injection).

This defines the right hand side of the coarse problem. On the other hand, $\tilde{e}^h$ is the solution of a difference operator which can be represented analogously on the new discretization level. For these reasons we solve the discrete equation

$$A^H\tilde{e}^H = d^H \ , \tag{1.10}$$

with homogeneous boundary conditions as for $\tilde{e}^h$. Here A^H represents the same discrete operator but relative to H, and the corresponding problem will be less expensive in terms of computational cost.

Reasonably one expects $\tilde{e}^H$ to be an approximation to $\tilde{e}^h$ on the coarse grid. From $\tilde{e}^H$ we obtain an approximation of the function $\tilde{e}^h$, which is defined on the original mesh, by means of some *prolongation* operator P, which in turns can be applied thanks to the smoothness of the functions involved. Therefore since $u^h = \tilde{u}^h - \tilde{e}^h$ is the exact solution we update the function $\tilde{u}^h$

$$\tilde{u}^h_{new} = \tilde{u}^h - P\tilde{e}^H \quad .$$

The step from $\tilde{u}^h$ to $\tilde{u}^h_{new}$ is called the *coarse level* (CL) correction. Notice that $\tilde{e}^h$ was a smooth function and the last step has amended $\tilde{u}^h$ by its smooth error. This explains why the CL correction is said to be complementary to the smoothing iteration. However, in practice, also the interpolation procedure may introduce HF errors on the fine grid. Therefore it is convenient to complete the twolevel cycle by to applying ν_2 times the smoothing iteration after the coarse level correction.

For clarity we summarize here the twolevel procedure described above:

Algorithm 1 (TL scheme)

- *Twolevel method for solving* $A^h u^h = f^h$.

  1. *ν_1 smoothing steps on the fine level (pre-smoothing):* $u^{(\nu_1)} = Mu^{(\nu_1-1)} + Nf^h$, *starting from* $u^{(0)}$;
  2. *Computation of the defect:* $d^h = A^h u^{(\nu_1)} - f^h$;
  3. *Restriction of the defect:* $d^H = Rd^h$;
  4. *Solution of the coarse problem:* $e^H = (A^H)^{-1} d^H$;
  5. *Coarse level correction:* $\tilde{u} = u^{(\nu_1)} - Pe^H$;
  6. *ν_2 smoothing steps on the fine level (post-smoothing):* $u^{(\nu_2)} = Mu^{(\nu_2-1)} + Nf^h$, *starting from* $u^{(0)} = \tilde{u}$.

Notice that the twolevel solution process can be seen as an iteration which applies to $u^{(0)}$ and gives the new approximation $u^{(\nu_2)}$. Actually it is of the form (1.1) as stated by the following [37]

Lemma 1 *The iteration matrix of the twolevel scheme is*

$$M_{TL} = M^{\nu_2}(I - P(A^H)^{-1}RA^h)M^{\nu_1} \quad , \tag{1.11}$$

where M is the smoothing iteration matrix.

Now let μ be the smoothing factor of the relaxation scheme used within a twolevel cycle. Assume that the coarse level correction step solves "exactly" the LF error components, and there is no interaction between high and low frequencies. This can be considered the "ideal" case. Then the error reduction, $\tilde{\rho}$, by one step of the TL method is determined by the reduction of the HF components on the fine grid. For this reason it can (roughly) be estimated by

$$\rho^* = \mu^{\nu_1+\nu_2} \quad . \tag{1.12}$$

This is the prediction of LMA. A sharper bound for the spectral radius of M_{TL} can be computed by the *twolevel local mode analysis* [15]. Here we report a twolevel convergence estimate for a TL method based on a damped Jacobi smoothing iteration with $\omega = 1/2$, assuming that P is the piecewise linear interpolation [37], and R is a simple weighted restriction [37] defined by $d^H(x_j) = (d^h(x_{j-1}) + 2d^h(x_j) + d^h(x_{j+1}))/4$, $j = 2, 4, \ldots, n-1$ (let us suppose n be an odd integer). Let us apply the twolevel algorithm 1 to solve (1.5). In this case the following theorem is proved [37] using discrete Fourier analysis

Theorem 2 *Let the twolevel iteration be defined by algorithm 1 with $\nu = \nu_1 + \nu_2 \geq 1$. The spectral radius of the iteration matrix M_{TL} given by (1.11) is bounded by*

$$\rho(M_{TL}) \leq \max\{\chi(1-\chi)^\nu + (1-\chi)\chi^\nu \ : \ 0 \leq \chi \leq 1/2\} =: \rho_\nu < 1 \quad ,$$

uniformly with respect to the mesh size h. Hence (1.11) is a convergent iteration.

In table 1.1 we list some values of the bounds predicted by local mode analysis (1.12) and by theorem 2 when the TL level method is applied to the problem above. We notice

ν	ρ^*	ρ_ν
1	0.5	0.5
2	0.25	0.25
3	0.125	0.125
4	0.0625	0.0832
5	0.03125	0.0671

Table 1.1: A comparison of error reduction factors as predicted by LMA or by theorem 2.

that the estimate ρ^* approximates well the bound ρ_ν provided that $\nu_1 + \nu_2$ is small. However for large ν, ρ^* has an exponential behaviour whereas $\rho_\nu \simeq \frac{1}{e\nu}$.

Since the secondary mesh is only twice larger than the original one, the solution of the coarse problem need not be simple at all. However the coarse problem to be solved in the TL scheme has the same form as the defect problem on the fine level. Thus one uses the same method to determine approximately $\tilde{e}^H$, i.e. equation (1.10) is itself solved by iteration combined with a further coarse level correction. This process can be repeated recursively until the coarsest grid is reached where the corresponding defect equation is inexpensive to solve. This is, roughly speaking, the qualitative description of the multilevel method.

For a more general description let us introduce a sequence of grids with mesh size $h_1 > h_2 > \ldots > h_M$, so that $h_{\ell-1} = 2h_\ell$. Here ℓ is called the *level number*. With Ω_{h_ℓ} we denote the set of grid points with grid spacing h_ℓ. The number of interior grid points will be n_ℓ.

Usually the discrete problem to be solved corresponding to level ℓ is denoted by $A^\ell u^\ell = f^\ell$. Clearly A^ℓ is a $n_\ell \times n_\ell$ matrix, and u^ℓ, f^ℓ are vectors of size n_ℓ. The connection between the levels ℓ and $\ell - 1$ is given by two linear mappings. The linear fine-to-coarse grid transfer operator, which we called simply restriction

$$I_\ell^{\ell-1} : \Omega_{h_\ell} \to \Omega_{h_{\ell-1}} \quad . \tag{1.13}$$

And the linear coarse-to-fine operator, called prolongation

$$I_{\ell-1}^{\ell} : \Omega_{h_{\ell-1}} \rightarrow \Omega_{h_\ell} \ . \tag{1.14}$$

Finally with $u^\ell = \hat{S}_\ell(u^\ell, f^\ell)$ we denote a general smoothing iteration. Hence is possible to define the multilevel solution procedure

Algorithm 2 (ML scheme)

- *Multilevel method for solving* $A^\ell u^\ell = f^\ell$.

  1. ν_1 *smoothing steps:* $u^\ell = \hat{S}_\ell^{\nu_1}(u^\ell, f^\ell)$;
  2. *Computation of the defect:* $d^\ell = A^\ell u^\ell - f^\ell$;
  3. *Restriction of the defect:* $d^{\ell-1} = I_\ell^{\ell-1} d^\ell$;
  4. *Set starting value* $e^{\ell-1} = 0$;
  5. *Application of* γ *times of the ML scheme to* $A^{\ell-1} e^{\ell-1} = d^{\ell-1}$ *(exact solution if* $\ell - 1 = 1$*)*;
  6. *Coarse level correction:* $u^\ell = u^\ell - I_{\ell-1}^{\ell} e^{\ell-1}$;
  7. ν_2 *smoothing steps:* $u^\ell = \hat{S}_\ell^{\nu_2}(u^\ell, f^\ell)$.

The multilevel algorithm involves a new parameter: γ is the number of times the ML procedure is applied to the coarse level problem. Since this procedure converges very fast, $\gamma = 1$ or $\gamma = 2$ are the typical values used. For $\gamma = 1$ the multilevel scheme is called "V-cycle", whereas $\gamma = 2$ is named "W-cycle". It turns out that for sufficiently large γ, the coarse problem is solved almost exactly. Therefore in this case the convergence factor of a multilevel cycle equals that of the corresponding TL method, i.e., approximately $\rho = \mu^{\nu_1+\nu_2}$. Actually in many problems $\gamma = 2$ or even $\gamma = 1$ are sufficient to retain the twolevel convergence.

1.3 Multilevel Methods for Non Linear Problems

Many problems of physical interest are non linear in character. And for most of them analytic solutions are not available. Therefore they are investigated numerically and for

this purpose the multilevel strategy provides an entire class of new and powerful algorithms. As we will see the ML approach described above applies with few modifications to non linear equations. Actually the simplest extension to these problems is the Newton multilevel approach [37]. That is by linearizing the non linear equations and then applying the multilevel algorithm described above. But another more suitable approach is normally followed, that is the so-called *full approximation scheme* (FAS) [10]. This method is an advanced and widely used multilevel technique which, although originally developed for non linear problems, is also applied equally to linear ones.

In the FAS scheme the global linearization in the Newton process is avoided and the only linearization is the local one used to define the relaxation procedure[2]. In order to describe the FAS method let us denote a non linear problem by $A(u) = f$, and its discretization by

$$A^\ell(u^\ell) = f^\ell \ , \tag{1.15}$$

where $A^\ell(\cdot)$ represents a non linear discrete operator.

The starting point for the FAS method is again to define a suitable smoothing iteration. We denote the corresponding smoothing process by $u = \hat{S}_\ell(u, f)$. Now suppose to apply a few times this iterative method to (1.15) obtaining some approximate solution $\tilde{u}^\ell$. The desired exact correction e^ℓ is defined by $A^\ell(\tilde{u}^\ell - e^\ell) = f^\ell$. Clearly here the coarse defect equation (1.10) makes no sense (no superposition). Nevertheless the "correction" equation can instead be written in the form

$$A^\ell(\tilde{u}^\ell) - A^\ell(\tilde{u}^\ell - e^\ell) = d^\ell \ , \tag{1.16}$$

if we define $d^\ell = A^\ell(\tilde{u}^\ell) - f^\ell$.

Then it is possible to perform the same steps as for the linear case but in terms of another coarse level variable, that is $\tilde{u}^\ell - e^\ell$ represented in the coarse space

$$\hat{u}^{\ell-1} := \hat{I}_\ell^{\ell-1}\tilde{u}^\ell - e^{\ell-1} \ . \tag{1.17}$$

Since $\hat{I}_\ell^{\ell-1}\tilde{u}^\ell$ and $\tilde{u}^\ell$ represent the same function but on different grids, the standard choice of the fine-to-coarse linear operator $\hat{I}_\ell^{\ell-1}$ is the straight injection.

[2]However one still uses Newton linearization in the local mode analysis.

Transferring (1.16) to the coarse level (replacing $A^\ell(\cdot)$ by $A^{\ell-1}(\cdot)$, $\tilde{u}^\ell$ by $\hat{I}_\ell^{\ell-1}\tilde{u}^\ell$, and d^ℓ by $I_\ell^{\ell-1}d^\ell$) we get the FAS equation

$$A^{\ell-1}(\hat{u}^{\ell-1}) = I_\ell^{\ell-1}f^\ell + \tau_\ell^{\ell-1} \quad , \tag{1.18}$$

where

$$\tau_\ell^{\ell-1} = A^{\ell-1}(\hat{I}_\ell^{\ell-1}\tilde{u}^\ell) - I_\ell^{\ell-1}A^\ell(\tilde{u}^\ell) \quad .$$

According to the twolevel philosophy we assume that (1.18) can be solved exactly. Observe that (1.18) without the $\tau_\ell^{\ell-1}$ term is the original equation represented on the coarse grid. Moreover at convergence $\hat{u}^{\ell-1} = \hat{I}_\ell^{\ell-1}u^\ell$, because $f^\ell - A^\ell(u^\ell) = 0$ and $A^{\ell-1}(\hat{u}^{\ell-1}) = A^{\ell-1}(\hat{I}_\ell^{\ell-1}u^\ell)$. Hence $\tau_\ell^{\ell-1}$ is the fine-to-coarse *defect correction.* That is the correction to (1.18) such that its solution coincides with the fine grid solution [3]. However to use directly $\tilde{u}_{new}^\ell = I_{\ell-1}^\ell\hat{u}^{\ell-1}$ would be worse since it introduces the interpolation errors of the full solution instead of the interpolation errors of only the correction $e^{\ell-1}$, which in principle is smooth because of the application of the smoothing iteration. For this reason the following coarse level correction is used

$$u^\ell = \tilde{u}^\ell - I_{\ell-1}^\ell(\hat{I}_\ell^{\ell-1}\tilde{u}^\ell - \hat{u}^{\ell-1}) \quad . \tag{1.19}$$

Notice that if the problem is linear the FAS steps are equivalent to the ones described by algorithm 1. And in fact numerical experiments show that for the same class of problems (for example elliptic problems) a standard ML solver and the corresponding FAS version exhibit the same efficiency. The complete FAS scheme is summarized below (we omit the accent $\tilde{}$):

Algorithm 3 (FAS scheme)

- *FAS method for solving* $A^\ell(u^\ell) = f^\ell$.

  1. ν_1 *smoothing steps:* $u^\ell = \hat{S}_\ell^{\nu_1}(u^\ell, f^\ell)$;

[3] This fact allows to reverse the point of view of the multilevel approach [13]. Instead of regarding the coarse level as a device for accelerating convergence on the fine grid, we can view the fine level as a device for calculating the correction $\tau_\ell^{\ell-1}$ to the FAS equation. In this way most of the calculation may proceed on coarser spaces.

2. *Transfer of the approximate solution:* $u^{\ell-1} = \hat{I}_{\ell}^{\ell-1} u^{\ell}$;

3. *Computation of the right hand side of the FAS equation:* $d^{\ell-1} = I_{\ell}^{\ell-1} f^{\ell} + [A^{\ell-1}(u^{\ell-1}) - I_{\ell}^{\ell-1} A^{\ell}(u^{\ell})]$;

4. *Application of* γ *times of the FAS scheme to* $A^{\ell-1}(\hat{u}^{\ell-1}) = d^{\ell-1}$ *(exact solution if* $\ell - 1 = 1$*);*

5. *Coarse level correction:* $u^{\ell} = u^{\ell} - I_{\ell-1}^{\ell}(u^{\ell-1} - \hat{u}^{\ell-1})$;

6. ν_2 *smoothing steps:* $u^{\ell} = \hat{S}_{\ell}^{\nu_2}(u^{\ell}, f^{\ell})$.

When we deal with non linear problems it is important to start the iterative procedure from a good initial approximation. This can be obtained by interpolating a (approximate) solution obtained on the next coarser level (suppose it exists). Namely

$$u_0^{\ell} = \tilde{I}_{\ell-1}^{\ell} u^{\ell-1} \quad , \tag{1.20}$$

which serves as starting point for the ML (FAS) solution process. The same process has been used to compute $u^{\ell-1}$, with $\ell - 1$ as the finest level. On the coarsest level one has to provide a starting guess u_0^1. However in case of non linear problems the choice decides which solution of the discrete problem is approximated by the multilevel solver.

1.3.1 The Full Multilevel Method

The idea of using coarse grid approximations as first guesses for the solution process on finer grids is known as *nested iteration.* The algorithm obtained by a combination of a multilevel scheme with the idea of nested iteration is called *full multi-grid* (FMG) method (see figure 1.1). In particular the operator (1.20) used in the FMG cycle is called FMG interpolator. Because of the improvement of the initial solution at each starting level, the FMG cycle results to be cheaper than the simple iterative application of a standard multilevel cycle.

Algorithm 4 (FMG scheme)

- *FMG method for solving* $A^{\ell}(u^{\ell}) = f^{\ell}$.

1. *Set the initial starting value on the coarsest grid:* u_0^1;

2. *FMG interpolation to the next finer grid:* $u^\ell = \tilde{I}_{\ell-1}^{\ell} u^{\ell-1}$;

3. *ML (or FAS) scheme applied to* $A^\ell(u^\ell) = f^\ell$, *starting from* u^ℓ;

4. $\ell := \ell + 1$ *go to* 2.

If on each current fine level ($\ell > 1$) N ML (or FAS) cycles are applied, then the algorithm is called N-FMG. With the N-FMG algorithm an estimate of the degree of accuracy can be obtained by comparing the solutions on each fine level. That is, denote with u^ℓ the approximate solution to the problem on the level ℓ after N ML (or FAS) cycles. Then in the FMG method this solution is interpolated to level $\ell + 1$ to serve as a first approximation for that level. At the end of N cycles on level $\ell + 1$, one obtains $u^{\ell+1}$. With these two solutions available an estimate of the solution error on level ℓ can be defined as

$$E^\ell = \max_{\Omega_{h_\ell}} |u^\ell - \hat{I}_{\ell+1}^{\ell} u^{\ell+1}| \ , \tag{1.21}$$

and so on finer levels.

In order to show the efficiency of the full multilevel approach we consider the application of this method to a three dimensional Poisson equation with Dirichlet boundary conditions. Namely

$$\begin{cases} -(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}) = 3\sin(x+y+z) & \text{in} \quad \Omega = (0,2)\times(0,2)\times(0,2) \\ u(x,y,z) = \sin(x+y+z) & \text{for} \quad (x,y,z) \in \Gamma = \partial\Omega \end{cases} \tag{1.22}$$

Finite difference approximations of (1.22) can be obtained in much the same way as was done in the one-space variable case. Thus, if on each level ℓ, we choose a uniform mesh with grid size $h_\ell = \frac{2}{n_\ell + 1}$, $n_\ell = 2^\ell - 1$, a direct application of the discretization scheme used in the 1D case can be made, and one obtains

$$-(\frac{u_{i+1jk}^\ell - 2u_{ijk}^\ell + u_{i-1jk}^\ell}{h_\ell^2} + \frac{u_{ij+1k}^\ell - 2u_{ijk}^\ell + u_{ij-1k}^\ell}{h_\ell^2} + \frac{u_{ijk+1}^\ell - 2u_{ijk}^\ell + u_{ijk-1}^\ell}{h_\ell^2})$$

$$= 3\sin(ih_\ell, jh_\ell, kh_\ell) \ , \quad i,j,k = 1,\ldots,n_\ell \ , \tag{1.23}$$

that is the 7-point approximation, which is $O(h^2)$ accurate [51].

level	10-FMG	3-FMG	1-FMG
3	$6.75\,10^{-4}$	$6.79\ 10^{-4}$	9.44×10^{-4}
4	$1.73\,10^{-4}$	$1.75\ 10^{-4}$	2.34×10^{-4}
5	$4.36\,10^{-5}$	$4.40\ 10^{-5}$	5.92×10^{-5}
6	$1.09\,10^{-5}$	$1.10\ 10^{-5}$	1.48×10^{-5}

Table 1.2: The behaviour of the solution error for various N-FMG cycles.

The problem is then defined and it remains to specify all the components which enter in the FMG algorithm. We use the FAS scheme (with $\gamma = 1$ and $M = 7$) discussed above.

The smoothing iteration used is the lexicographic Gauss-Seidel method, which is applied $\nu_1 = 2$ times for pre-smoothing and $\nu_2 = 1$ times as post-smoother and its smoothing factor computed with local mode analysis is $\mu = 0.567$. Hence, by this analysis, the expected reduction factor is $\rho^* = \mu^{\nu_1+\nu_2} = 0.18$. In fact, after 3 to 10 FAS cycles, the observed reduction is ~ 0.20.

The transfer operator of the approximate solution to the coarser level $\hat{I}_\ell^{\ell-1}$ is the simple injection. The other grid transfer operators $I_\ell^{\ell-1}$ and $I_{\ell-1}^\ell$ are made by the full weighting and linear interpolation, respectively (see e.g. [79]). Finally, the FMG operator $\tilde{I}_{\ell-1}^\ell$ is the cubic interpolation [13, 37]. Now let us discuss the optimality of the FMG algorithm constructed with these components.

We said that the FMG scheme is able to solve the given discrete problem at a minimal cost. In order to show this, let us remark that if a sufficiently large number N of ML (or FAS) cycles is applied at each level then in that level the convergence to the solution is obtained. Therefore u^ℓ and $u^{\ell+1}$ considered above should give a relative solution error E^ℓ proportional to h_ℓ^2, since they solve the same problem but with different $O(h^2)$ truncation errors [4]. For this reason we expect that the ratio E^l/E^{l+1} at convergence should be $h_\ell^2/h_{\ell+1}^2 = 4$.

[4]The error occurred by approximating the differential equation as a difference equation.

In fact, when we apply the N-FMG method with (large) $N = 10$ we have the confirmation of this behaviour (see the column 10-FMG of table 1.2), but this actually happens also with $N = 3$ and $N = 1$ FMG algorithms. Moreover, as reported in table 1.2, the 1-FMG scheme gives the same order of magnitude of error as the 10-FMG scheme. Therefore the choice $N = 1$ in a FMG cycle is suitable to solve the problem to the order of the truncation error.

Now, in order to estimate the amount of work invested in the FMG method described above, we denote with WU (Work Unit) [10] the computational work by one relaxation sweep on the finest level M. The number of computer operations in such a work is proportional to the number of finest grid points times the number of operations required to compute the new solution's value at each point. Then, on any level ($\ell \leq M$) the work involved is $(\frac{1}{2})^{3(M-\ell)}WU$, where the factor $\frac{1}{2}$ derives from the mesh size ratio $h_{\ell+1}/h_\ell$ and the exponent 3 is the spatial dimension of the problem's domain. Thus a multilevel cycle which uses $\nu = \nu_1 + \nu_2$ relaxation sweeps on each level requires

$$W_{cycle} = \nu \sum_{\ell=1}^{M} (\frac{1}{2})^{3(M-\ell)} WU < \frac{8}{7} \nu WU \quad ,$$

ignoring transfer operations. Hence the computational work employed in a N-FMG method is roughly

$$W_{FMG} = N \sum_{\ell=2}^{M} (\frac{1}{2})^{3(M-\ell)} W_{cycle} \quad ,$$

ignoring the FMG interpolation and work on the coarsest grid.

This means that, using the 1-FMG method, we solve the discrete 3D Poisson problem to the order of the truncation error with a number of computer operations proportional to the number of solution variables of the finest grid space (from the formula above this work is $\sim 4WU$).

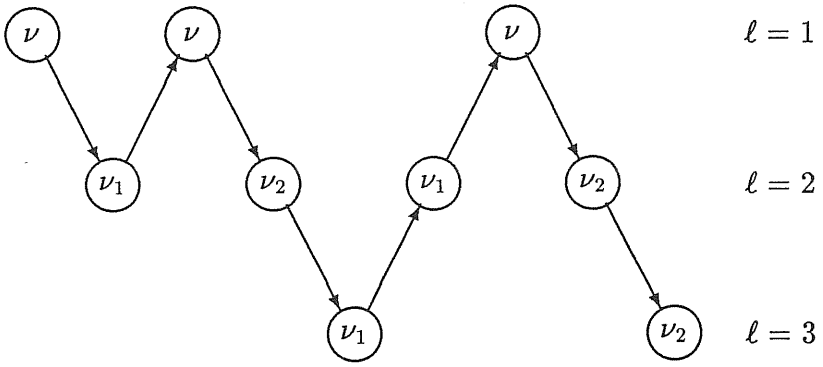

Figure 1.1: The FMG cycle.

Chapter 2

Burgers' Equation and Multilevel Techniques

2.1 Introduction

A multilevel solver for elliptic equations can be naturally adapted in order to solve parabolic problems. In fact in the simplest approach any implicit time discretization [70] of an equation of parabolic type leads to a sequence of (discrete) elliptic problems to be solved. Then a ML solver can be used to compute the solution on the finest level at each time step. However, since we have at disposal many levels of discretization it is possible to formulate more suitable procedures which we present here. These, in any case, are based on the fact that changes in the solution between problem steps are usually dominated by smooth components. For this reason one should solve on coarser levels before processing on the finest grid. Consequently, because of the smoothness, only coarse level approximation to the increment, not the CL approximation to the solution, will be interpolated to the finer level.

In order to study how to implement a multilevel method for evolution equations we have considered a simple fluid flow problem: that described by Burgers' equation [27]. This choice is also motivated by our intention to develop a multilevel code suitable to describe the density distribution arising at the non linear stage of gravitational instability which is similar to intermittence phenomena in acoustic turbulence which can be

described by Burgers' equation [75]. From a more mathematical point of view we notice that this equation is one of the simplest non linear partial differential equations for which it is possible to obtain an exact solution. Also, depending on the magnitude of the various terms in Burgers' equation, it behaves as an elliptic, parabolic or hyperbolic partial differential equation. Therefore, Burgers' equation has been used widely as a model equation for testing and comparing computational techniques [27].

2.1.1 Burgers' Equation

Let us present the nonlinear diffusive wave equation known as Burgers' equation [72]

$$\partial_t u + u\partial_x u - \delta\partial_{xx} u = 0 \quad , \tag{2.1}$$

where $\partial_x \phi \equiv \frac{\partial \phi}{\partial x}$, $\partial_t \phi \equiv \frac{\partial \phi}{\partial t}$ and $\partial_{xx}\phi \equiv \frac{\partial^2 \phi}{\partial x^2}$. δ is the so called *diffusion coefficient.* In a typical application, u is a velocity-like dependent variable. And δ is the inverse of the Reynolds number. In particular this equation describes a balance between the nonlinear convective term ($u\partial_x u$) and the linear dissipative term ($-\partial_{xx} u$). These terms are present in the same way in the incompressible Navier-Stokes (NS) equation, making (2.1) a qualitatively correct approximation to the one-dimensional NS equation. However, for some dissipative flow problems such as shock propagation, compressible turbulence, etc., (2.1) is considered to be the appropriate mathematical model. In addition Burgers' equation provides a suitable model for testing computational algorithms for problems involving the evolution of shocks. This also because for a wide range of initial and boundary conditions, exact solutions exist and can be obtained easily (see for instance [72]).

2.1.2 A Discretization of Burgers' Equation

We consider the implicit time discretization of the initial boundary value problem in a bounded domain Ω,

$$\begin{aligned} &\partial_t u + u\partial_x u - \delta\partial_{xx} u = 0 \quad , \\ &u(x,0) = u_0(x) \quad in \ \ \Omega \quad , \end{aligned} \tag{2.2}$$

$$u(x,t) = g_\Gamma(t) \quad , \quad \text{on the boundary} \quad x \in \Gamma \quad . \tag{2.3}$$

The functions $u_0(x)$ and $g_\Gamma(t)$ give the initial and boundary values respectively. Let us choose as Ω the bounded domain (a,b) and $\Gamma = \{a, b\}$. We construct a grid Ω_h on Ω with mesh size $h = \frac{(b-a)}{n+1}$, n being the number of interior grid points. We suppose to discretize Burgers' equation on the grid Ω_h at the time step $t = k\Delta t$, where $\Delta t > 0$ is the step size. Hence the solution is approximated by a grid function

$$u^{h,k} = (u_1^{h,k}, u_2^{h,k}, ..., u_n^{h,k})^T \quad ,$$

(T means transpose) where $u_i^{h,k}$ represents the value of the solution corresponding to the interior grid point $x = a + ih$ of a given grid space Ω_h, at the time step $k\Delta t$. For the discretization of (2.1) we focus our attention on two facts. Firstly, when the diffusion coefficient becomes small, equation (2.1) tends to coincide with the inviscid Burgers equation which can produce discontinuous solutions [72]. Therefore one must be careful in the discretization of the gradient $\partial_x u$ in the nonlinear term of (2.1). Secondly, as $\delta \to 0$ any centered discretization of (2.1) becomes unstable. Hence we use a non centered discretization for the nonlinear term because it makes the finite difference scheme stable, independently of the mesh size h. This is important in a multilevel context as will be explained below. In this way a numerical viscosity arises and there is no need to introduce an artificial viscosity (as it is done in [28] where one has the additional problem of choosing a viscosity parameter). In addition, the so obtained discrete equation is consistent with the same coarser or finer (grid) equation [37]. However starting from a scheme with $O(h^2)$ spatial truncation error the resulting discrete equation only has $O(h)$ truncation error.

Now let us denote with ∂_i^h and ∂_{ii}^h the difference-quotient operators approximating ∂_x and ∂_{xx} respectively. Following [28] we discretize the nonlinear term using a Taylor series expansion with respect to time. This procedure leads to the following discrete problem (for economy of notation we omit the discretization parameter h):

$$\frac{u_i^k - u_i^{k-1}}{\Delta t} + \frac{1}{2}\partial_i[u^k u^{k-1}] - \delta\partial_{ii}u^k = 0 \quad , \tag{2.4}$$

with initial condition

$$u_i^0 = u_0(a + ih) \quad , \quad 1 \le i \le n \tag{2.5}$$

and boundary conditions

$$u_0^k = g_0(k\Delta t) \quad , \quad u_{n+1}^k = g_{n+1}(k\Delta t) \ . \tag{2.6}$$

We obtain a stable discretization by using a backward (or forward) formulae (the reason is explained below), that is $\partial_i^h \equiv \frac{[-1,1,0]}{h}$ (or $\partial_i^h \equiv \frac{[0,-1,1]}{h}$) and $\partial_{ii}^h \equiv \frac{[1,-2,1]}{h^2}$ [1].

This produces a $O(\Delta t) + O(h)$ scheme. If we choose $\partial_i^h \equiv \frac{[-1,0,1]}{2h}$ then a stable discretization will require $h < 4\delta$, which inhibits the use of coarse discretization spaces.

Notice that (2.4) is a tridiagonal system of n linear equations for the n unknowns u_i^k , $i = 1, \ldots, n$. That is

$$-(\frac{u_{i-1}^{k-1}}{2h} + \frac{\delta}{h^2})u_{i-1}^k + (\frac{1}{\Delta t} + \frac{u_i^{k-1}}{2h} + \frac{2\delta}{h^2})u_i^k - (\frac{\delta}{h^2})u_{i+1}^k = \frac{u_i^{k-1}}{\Delta t} \ , \tag{2.7}$$

where u_0^k , u_{n+1}^k are known from the boundary conditions. Let us denote this system by $A(u^{k-1})u^k = f(u^{k-1})$. The boundary conditions are incorporated in the right hand side. Looking at this linear system it is easy to justify why we have chosen a backward discretization. In fact we have implicitly assumed that the solution u will be a positive function. Therefore the use of this non centered discretization gives the coefficients of (2.7) in parenthesis to be sums of positive terms which, with an additional condition specified below guarantees that the matrix of coefficients to be a "stable" $\mathcal{M}$-matrix (see e.g. [62]). On the other hand, if one assumes that $u \leq 0$ the same considerations above and the following results hold using a forward discretization. So let us restrict to consider only the case $u \geq 0$ for which the backward discretization is appropriate[2].

We can investigate the algebraic features of the problem $A(u^{k-1})u^k = f(u^{k-1})$ using the formalism of $\mathcal{M}$-matrices [62]. Then we state a condition so that $A(u^{k-1})$ is of this class of matrices and prove a lemma that allows to prove how this property is preserved during the evolution.

Definition 1 *A matrix $A \in L(\mathbb{R}^n)$ is an $\mathcal{M}$-matrix if A is invertible, $A^{-1} \geq 0$, and $A_{ij} \leq 0$ for all $i, j = 1, \ldots, n$, $i \neq j$.*

[1] This is the so-called *difference stencil notation.* That is the pattern of points involved in a difference operator, together with the numerical coefficients.

[2] This makes sense since $u \geq 0$ and $u \leq 0$ really occur in Burgers' equation.

for the following discussion we need also

Definition 2 *A matrix $A \in L(\mathbb{R}^n)$ is irreducible if and only if for any two distinct indices $1 \leq i \leq j \leq n$, there is a sequence of nonzero elements of A of the form $\{a_{ii_1}, a_{i_1 i_2}, \ldots, a_{i_m j}\}$.*

Therefore we can recall a useful theorem for $\mathcal{M}$-matrices [62],

Theorem 3 *Let $A \in L(\mathbb{R}^n)$ be irreducibly diagonally dominant and assume $A_{ij} \leq 0$, $i \neq j$, and that $A_{ii} > 0$, $i = 1, \ldots, n$. Then A is an $\mathcal{M}$-matrix.*

Then we can prove a lemma which gives a sufficient condition to be satisfied so that the above discretization of (2.1) results in a matrix $A(u^{k-1})$ which is an $\mathcal{M}$-matrix

Lemma 2 *If $u_i^{k-1} \geq 0$ and $\frac{1}{\Delta t} \geq \max_i \frac{1}{2}|\frac{u_i^{k-1} - u_{i-1}^{k-1}}{h}|$, then $A(u^{k-1})$ is an $\mathcal{M}$-matrix.*

Proof. The condition of (weak) diagonal dominance, where strict inequality holds for at least one i, is expressed by

$$|A(u^{k-1})_{ii}| \geq \sum_{j=1, j\neq i}^{n} |A(u^{k-1})_{ij}| \quad , \quad i = 1, \ldots, n \ . \tag{2.8}$$

This is immediately satisfied by $A(u^{k-1})$ because

$$\begin{aligned} \sum_{j=1, j\neq i}^{n} |A(u^{k-1})_{ij}| &= \frac{\delta}{h^2} + (\frac{u_{i-1}^{k-1}}{2h} + \frac{\delta}{h^2}) = \frac{2\delta}{h^2} - \frac{u_i^{k-1} - u_{i-1}^{k-1}}{2h} + \frac{u_i^{k-1}}{2h} \\ &\leq \frac{2\delta}{h^2} + |\frac{u_i^{k-1} - u_{i-1}^{k-1}}{2h}| + \frac{u_i^{k-1}}{2h} \leq A(u^{k-1})_{ii} \quad , \quad i = 1, \ldots, n \ . \end{aligned}$$

Then $A(u^{k-1})$ is diagonal dominant. In addition we have $A(u^{k-1})_{ii} > 0$, $i = 1, \ldots, n$ and $A(u^{k-1})_{ij} \leq 0$, $i \neq j$. Moreover, for any $i < j$ exists a sequence $\{a_{ii+1}, a_{i+1i+2}, \ldots, a_{j-1j}\}$ and for $j < i$ we have $\{a_{jj+1}, a_{j+1j+2}, \ldots, a_{i-1i}\}$, wherefrom follows that $A(u^{k-1})$ is irreducible. Hence, by theorem 3, $A(u^{k-1})$ is an $\mathcal{M}$-matrix.

From now on we will refer to the conditions $u_i^{k-1} \geq 0$ and $\frac{1}{\Delta t} \geq \max_i \frac{1}{2}|\frac{u_i^{k-1} - u_{i-1}^{k-1}}{h}|$, as the $\mathcal{M}$-conditions. We then are able to prove the following

Lemma 3 *If the $\mathcal{M}$-conditions are satisfied at level $k-1$ and the boundary values are positive or zero then u^k exists and is nonnegative.*

Proof. Because the $\mathcal{M}$-conditions are satisfied $A(u^{k-1})$ is an $\mathcal{M}$-matrix. Then it is invertible and the solution at the time step $k\Delta t$ for the discretization (2.4) exists and is given by $u^k = A(u^{k-1})^{-1} f(u^{k-1})$. Moreover if $u_i^{k-1} \geq 0$, $i = 0, \ldots, n+1$ and $u_0^k \geq 0$ and $u_{n+1}^k \geq 0$, one immediately sees that $f(u^{k-1})_i \geq 0$, $i = 1, \ldots, n$. Then the solution u^k is positive thanks to the positive definiteness of the inverse $A(u^{k-1})^{-1}$ and of the vector $f(u^{k-1})$.

The last lemma states that starting from a non negative initial data the solution will evolve remaining positive if at any time level k the boundary values are nonnegative and $\frac{1}{\Delta t} \geq \max_i \frac{1}{2} |\frac{u_i^{k-1} - u_{i-1}^{k-1}}{h}|$. Notice that this restriction on the time step Δt poses no severe limitations in practice. With these conditions $A(u^{k-1})$ will remain an $\mathcal{M}$-matrix during the evolution.

The above conditions pose, in principle, a limitation in the step size Δt. Depending on the behaviour of the solution with the time, Δt could be increased or must be reduced during the evolution. In any case the diffusive term in (2.1) prevents the development of steep wave profiles and tends to spread the sharp discontinuities into smooth profiles so that the step size Δt remains finite. For simplicity we shall consider an example where Δt is fixed. Nevertheless the multilevel method we use will also apply with variable step sizes [37, 36].

2.2 Multilevel Methods and Evolution Equations

The problem is to solve the sequence of equations (2.7). At each step k the right hand side is a function of the solution at the previous time step and this fact allows errors to propagate in time. For this reason, in these kinds of problems, if one uses an iterative method, to solve the equations at a given step, the part which is not effectively reduced by the iteration will propagate. Therefore it is evident the importance of a numerical scheme like the multilevel method which solves all error spectral components.

In particular, since the entries of both $A(u^{k-1})$ and $f(u^{k-1})$ depend on the solutions' value of the previous time step, this is a problem with variable coefficients which has its natural treatment using a FAS approach where the full solution and not the error is represented on each level of discretization.

With this scheme we use the Gauss-Seidel iteration to dump effectively the HF components of the error. We denote such smoothing iteration by:

$$u^h_{new} = \hat{S}^h(u^h_{old}, F^h) \ , \tag{2.9}$$

where we have omitted the time level k. Because $A(u^{k-1})$ is an $\mathcal{M}$-matrix, this is a sufficient condition that (2.9) applied to (2.7) converges to the solution [62]. However this global condition does not say anything about the smoothing properties. As we said, (2.7) has variable coefficients and this fact renders local mode analysis difficult to apply. But, as we prove in chapter 4, these problems are easily investigated using an algebraic approach. So, for example, if we apply our results, derived there, to analyze the smoothing properties of the pointwise GS iteration applied to (2.7) we find easily the value of the smoothing factor. That is

$$\mu = \max_{i=1,n}\left(\frac{\delta/h^2}{\frac{1}{\Delta t} + \frac{u_i^{k-1}}{2h} + \frac{2\delta}{h^2}}\right) < \frac{1}{2} \ .$$

that is a good smoother.

2.2.1 The Modified Nested Iteration and Other Methods

We consider now the problem of how to advance the solution in time using multilevel methods. Here we describe the *modified nested iteration* (MNI) proposed by W. Hackbusch [36]. The first step of this method is to solve the discrete problem associated to the coarsest level Ω_{h_1} where the solution is represented in few grid points. The corresponding system of equations $A^1(u^{1,k-1})u^{1,k} = f(u^{1,k-1})$, for the current time step $k\Delta t$, is solved by applying a few steps of the iteration procedure (2.9) obtaining the approximate solution $\tilde{u}^{1,k}$. On this coarsest grid the difference between the obtained solution and the one relative to the previous time step is computed, $\tilde{u}^{1,k} - \tilde{u}^{1,k-1}$. This difference is used to construct the approximation to the solution for the current time step in the following

finer level. That is increasing the level by one and then applying a prolongation operator $P_{\ell-1}^{\ell}$, so we have

$$\tilde{u}^{\ell,k} = \tilde{u}^{\ell,k-1} + P_{\ell-1}^{\ell}(\tilde{u}^{\ell-1,k} - \tilde{u}^{\ell-1,k-1}) \ . \tag{2.10}$$

This approximation is then used as a starting value for the multilevel solution process in order to get the solution on the finer level. The procedure is repeated until the finest level ($\ell = M$) is reached. The method is summarized below

Algorithm 5 (MNI scheme)

- *MNI method for solving* $A^{\ell}(u^{\ell,k-1})u^{\ell,k} = f(u^{\ell,k-1})$.

  1. *Compute the solution for the current time step in the coarsest level,* $\tilde{u}^{1,k}$;
  2. *Increase* ℓ *by one and use the computed values at level* $\ell - 1$ *to approximate the solution in the next finer grid* ℓ :

     $$\tilde{u}^{\ell,k} = \tilde{u}^{\ell,k-1} + P_{\ell-1}^{\ell}(\tilde{u}^{\ell-1,k} - \tilde{u}^{\ell-1,k-1});$$

  3. *Solve with FAS the discretized equation* $A^{\ell}(u^{\ell,k-1})u^{\ell,k} = f(u^{\ell,k-1})$ *at level* ℓ *using the given initial approximation* $\tilde{u}^{\ell,k}$;
  4. *Repeat the procedure starting from 2. until the finest level* M *is reached.*
  5. *Increase the time level* k *by one and go to 1.*

A different version of this algorithm is presented in [18]. There the equation to be solved in the actual finer level has a different right hand side. That is, in point 3. of the MNI scheme, one has to solve $A^{\ell}(u^{\ell,k-1})u^{\ell,k} = f(u^{\ell,k-1}) + \tau_{\ell+1}^{\ell}(u^{\ell+1,k-1})$, where $\tau_{\ell+1}^{\ell}$ ($\ell < M$) is the fine-to-coarse defect correction (relative to u^{k-1}) given as in sec. 1.3. In [18] the resulting algorithm is called *stationary*, and algorithm 5 is called *non stationary*. We want to mention that the stationary approach is a prototype of new ML methods, under developments, for evolution equations where the time-step is performed on coarser spaces while retaining the accuracy of the finest level. This is obtained by "visiting" the finest level to update $\tau_{\ell+1}^{\ell}$, adaptively or regularly.

Another method to be mentioned is the *multigrid waveform relaxation* [80]. With this method, after spatial discretization and incorporation of the boundary conditions, a parabolic problem is transformed into a system of ordinary differential equations with one equation defined at each grid point. Then a *waveform relaxation* [57] is used in combination with the multilevel idea to solve the resulting system of ordinary differential equations.

In the following numerical experiments we use the MNI scheme. The choice is motivated by the simplicity of the method and because it represents the base of many other ML methods for evolution equations.

2.2.2 Numerical Investigation

We solve Burgers equation in the domain $\Omega = (a, b)$ with the initial condition

$$u_0(x) = u_1 + \frac{u_2 - u_1}{2}\{1 - \tanh[\frac{u_2 - u_1}{4\delta}x]\} \ , \tag{2.11}$$

and boundary conditions $u_1 = u_0(b)$ and $u_2 = u_0(a)$. We choose a and b so that the boundary conditions approximate well the asymptotic values of the steady state solution (together with the asymptotic vanishing of the spatial derivative of u)

$$u(x,t) = u_1 + \frac{u_2 - u_1}{2}\{1 - \tanh[\frac{u_2 - u_1}{4\delta}(x - Ut)]\} \ , \tag{2.12}$$

where $U = \frac{u_1+u_2}{2}$. This solution is known as the Taylor shock solution [72], U being the velocity of the shock. Let us assume that $u_1 = 0$. With these conditions the sum of the n discrete equations (2.4) gives

$$S^k = S^{k-1} + \frac{\Delta t U}{h}u_2 \ , \tag{2.13}$$

where $S^k = \sum_{i=1}^n u_i^k$, where $u_i^k \geq 0$, $i = 1, \ldots, n$ by lemma 3. The term $\frac{\Delta t U}{h}$ can be interpreted as the number of new grid points reached by the shock in one time step. Then (2.13) means that the evolution of the discrete Burgers equation (2.4) is close to that of a Taylor shock satisfying the given initial-boundary conditions. This approximation improves as $\delta \to 0$. Actually we observe that for $\delta = 10^{-6}$, the analytical solution (2.12) solves exactly the discrete equation, as proved by (2.13).

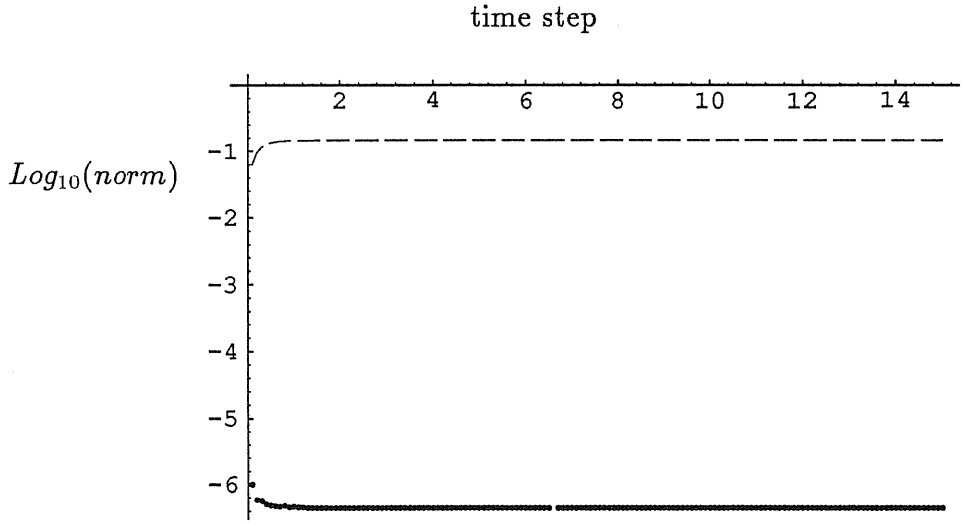

Figure 2.1: Time behaviour for solution error (dashed line) and residual norms (dotted line).

Now we consider the case $\delta = 0.01$. As for the experiment with $\delta = 10^{-6}$, the boundary values are given by $u_1 = 0$ and $u_2 = 1$, so that $U = 0.5$. The numerical domain is $\Omega = (-2, 10.8)$. The number of the interior points of the finest grid is 255 ($h_M = 0.05$) and $M = 7$, this means that the coarsest space has 3 interior grid points. In all numerical experiments the time step size is $\Delta t = \frac{1}{10}$. We verify numerically that with the above values of the discretization parameters the $\mathcal{M}$-conditions are satisfied during the evolution. As we expect the FAS scheme is very fast to converge. In particular, increasing the level number M, we notice that the asymptotic FAS convergence rate remains bounded by 0.2, whereas that of the GS iteration tends to 1.

We apply 150 times the MNI scheme to advance the solution in time. It results to be a shock which propagates with a velocity of 0.5, the theoretical one. This can be seen from the values of the error norm $\max_{i=1,n} |u_i^k - u(x_i, t_k)|$, reported in figure 2.1. In fact its value does not grow in time. In the same figure we also plot the time behaviour of the L^2 norm of the defect. This figure shows the good convergence in time of the modified nested iteration for the equation considered.

In order to show graphically the accuracy of the numerical solution obtained using the MNI scheme, we compare in figure 2.2 the numerical and analytical solutions after 150 times of the step $\Delta t = 0.1$.

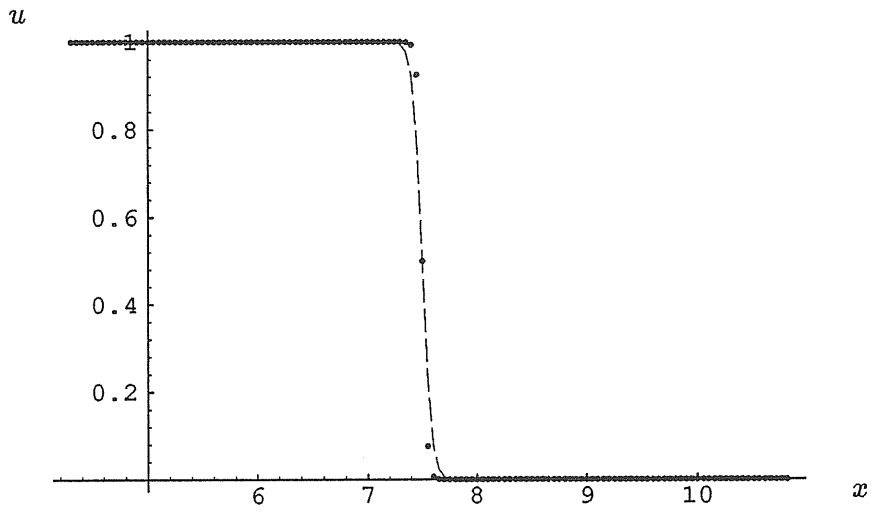

Figure 2.2: A portion of shock's profile after 150 time-steps. The numerical solution (dashed line) and the analytical one (dotted line).

2.2.3 Remarks and Conclusion

As we said in the introduction to this chapter, our purpose here was to commence the study of multilevel methods for evolution equations. However the choice of Burgers' equation in one spatial dimension is not interesting for practical calculations since it does not represent the main difficulties encountered in the numerical solution of higher dimensional flow problems. Nevertheless Burgers' equation has two merits. Firstly its algebraic investigation leads us to analyze multilevel methods algebraically. This study turned out to be very successful as we will see in chapter 4. For example the algebraic estimation of the smoothing factor given above.

Secondly, Burgers' equation has stimulated us to start to study and implement some special numerical techniques. Like for example, grid alignment [13] and local mesh refinements [1] within a ML scheme.

Chapter 3

The TBA Equations Solved by Multilevel Methods

3.1 Introduction

Although multilevel methods were originally introduced to solve elliptic problems, the same strategy can be applied to many other types of equations. This fact is not surprising since the ML's basic principles of computation are inspired by general considerations on the multi-scale nature of most of the physical problems. In this chapter we are concerned with a physical situation where the numerical solution of a system of integral equations is involved. For the moment let us refer to the general Fredholm integral equation of the second kind

$$u(x) = f(x) + \int_{\Omega} K(x, y, u(y))dy \quad , \quad x \in \Omega \subseteq \mathbb{E}^d \quad . \tag{3.1}$$

Let us assume that the kernel, the forcing term, and the solution itself are smooth in character. When this equation is discretized by any quadrature formula on a grid with $n = O(h^{-d})$ points, an algebraic system is obtained which is full. That is the unknowns are all connected to each other and for this reason the large scale component of the solution is the predominant part to be solved. In correspondence a relatively small number of unknowns is sufficient to exhibit this large scale dependence in the solution's values. This means that a rough estimate of the solution can be obtained on a coarse discretization space. On the other hand, in order to solve for the high frequency components of the solution

the number of points where the solution u must be evaluated can be very large. All these facts motivate the use of multilevel techniques. Actually the dominant part of the computation is the evaluation of the integration in (3.1). So a FMG scheme should be used since it provides a good initial guess by solving the problem on coarser grids where it is computationally cheaper. For the same reason, in the ML cycle used, the coarse level correction will provide a good improvement in the solution without costing too much. Regarding the relaxation, numerical experiments show that in almost all situations the convenient number of pre- and post- smoothing iterations on each level is 1.

Notice that the matrix of the (linearized) discrete system is full and a solution process based upon elimination would require $O(n^3)$ operations. On the other, iterative techniques require $O(n^2 \log n)$ operations [22]. With a FMG method the computing time for integral equations is reduced to $O(n^2)$ operations [38]. This would give only a small gain for small n, however, since the problem we solve here requires a very high accuracy the overall computing time is nevertheless reduced considerably. This gain results to be even more evident in our application because it requires the solution of a *system* of coupled non linear Fredholm integral equations of the second kind.

3.1.1 The Physical Problem

Massive relativistic field theories can be described on-shell by their scattering matrix. This approach is specially fruitful in two dimensions, where there exists a large class of models which are integrable, and their S-matrix can be computed exactly, being factorizable [86]. Unfortunately there is no general direct method in order to compute the S-matrix of a theory, but usually it is conjectured from general axioms and the underlying symmetries of the corresponding Hamiltonian.

The thermodynamic Bethe ansatz (TBA) was developed in order to provide a means to link a conjectured scattering theory with the underlying field theory [87]. It describes the finite temperature effects of the factorized relativistic field theory, using the S matrix as an input. If one studies the high temperature limit of the TBA equations, one can identify the conformal field theory (CFT) which governs the ultraviolet (UV) behaviour of the underlying field theory. One should though note, that it is not guaranteed that every

consistent S-matrix describes the scattering in some field theoretical model! Therefore the axiomatic bootstrap approach is only of limited value if not linked to field theory by some means, wherefrom the TBA is one of the most powerful ones.

Given the scattering data one can in most cases extract analytically the central charge of the CFT reached in the conformal limit, and in some cases the dimension of the perturbing operator, if the symmetry of the problem is known. Numerical calculations on the other hand can solve the TBA equations and therefore extract any measurable quantity.

In [87, 46] the TBA equations were resolved by an iterative method. We study here a multilevel algorithm, which is considerable faster, an important fact if many particles are involved. The heart of the program is the resolution of the coupled integral equations. Around this core we have designed some utility-programs, in order to make the tool easier to use. We specialize our application to the case of diagonal S-matrices, see e.g. [87, 46, 55]. As physical quantities we extract the central charge, the dimension of the perturbing field and the perturbation expansion. Note though, that one can easily calculate other quantities.

3.1.2 The TBA Equations

Consider an integrable massive scattering theory on a cylinder. This implies factorized scattering, and so one can assume that the wave function of the particles is well described by a free wave function in the intermediate region of two scattering processes. Take the ansatz

$$\psi(x_1 \dots x_N) = e^{i\sum p_j x_j} \sum_P A(P)\Theta(x_P) \quad ,$$

$A(P)$ are coefficients of the momenta whose ordering is specified by

$$\Theta(x_P) = \begin{cases} 1 & \text{if} \quad x_{p_1} < \dots < x_{p_N} \\ 0 & \text{otherwise} \end{cases} \quad .$$

Let the permutation P differ from P' by the exchange of the indices k and j. Then

$$A(P') = S_{kj}(\beta_k - \beta_j)A(P) \quad . \tag{3.2}$$

We impose antiperiodic boundary conditions for our wave functions, which provides that two particles cannot have equal momenta, leading to the condition

$$A(k, p_2, \ldots, p_N) = -e^{ip_k L} A(p_2, \ldots, p_N, k) \ , \tag{3.3}$$

L being the length of the strip on which we consider the theory. comparing (3.2) and (3.3), one realizes

$$e^{iLm_k \sinh\beta_k} \prod_{j\neq k} S_{kj}(\beta_k - \beta_j) = -1 \ \text{ for } \ k = 1, 2, \ldots N \ . \tag{3.4}$$

We introduce the phase $\delta_{kj}(\beta_k - \beta_j) \equiv -i \ln S_{kj}(\beta_k - \beta_j)$. In terms of these the equation become

$$Lm_k \sinh\beta_k + \sum_{j\neq k} \delta_{kj}(\beta_k - \beta_j) = 2\pi N_k \ \text{ for } \ k = 1, 2, \ldots, N \ , \tag{3.5}$$

N_k being some integers. These coupled transcendental equations for the rapidities are called the Bethe ansatz equations. One tries to solve these equations in the thermodynamic limit introducing densities of rapidities for each particle species and transferring the equations into integral equations. That is, let $\rho_1^{(a)}(\beta) = \frac{N}{\Delta\beta}$, where we assume that there are n particles in the small interval $\Delta\beta$, be the particle density and $\rho^{(a)}(\beta) = \frac{N_k}{\Delta\beta}$ be the level density corresponding to the particle a, then (3.5) become

$$m_a L \cosh\beta + \sum_{b=1}^{N} \int_{-\infty}^{\infty} \varphi_{ab}(\beta - \beta') \rho_1^{(a)}(\beta') d\beta' = 2\pi \rho^{(a)} \ , \tag{3.6}$$

with $\varphi_{ab}(\beta) = -i \frac{d}{d\beta} \log S_{ab}(\beta)$.

In order to compute the ground state energy one needs to minimize the free energy

$$RLf(\rho, \rho_1) = RH_B(\rho_1) + S(\rho, \rho_1) \ , \tag{3.7}$$

where $H_B = \sum_a m_a \int \cosh\beta \rho_1^{(a)} d\beta$ and S denotes the entropy. The extremum condition for a fermionic system[1] takes the form

$$-rM_a \cosh\beta + \epsilon_a(\beta) = \sum_{b=1}^{N} \int_{-\infty}^{\infty} \varphi_{ab}(\beta - \beta') \log(1 + e^{-\epsilon_b(\beta)}) \frac{d\beta'}{2\pi} \ , \tag{3.8}$$

[1] We use the fermionic TBA equations since in diagonal scattering up to now they turned out to be the relevant ones, see e.g.[87] for the general theory

$a = 1, 2, \ldots, N$, where we introduced the so-called pseudo-density $e^{-\epsilon_a} \equiv \frac{\rho_1^{(a)}}{\rho^{(a)} - \rho_1^{(a)}}$, the scaling length $r = Rm_1$ and the rescaled masses $M_a = \frac{m_a}{m_1}$; m_1 is the lightest particle mass. These coupled integral equations are called the TBA equations. The extremal free energy depends only on the ratios $\frac{\rho_1^{(a)}}{\rho^{(a)}}$ and is given by

$$f(r) = -\frac{r}{2\pi} \sum_{a=1}^{N} M_a \int_{-\infty}^{\infty} \cosh\beta \log(1 + e^{-\epsilon_a(\beta)}) d\beta \quad . \tag{3.9}$$

One can extract several physical quantities from the solution of the TBA equations ([87, 54, 46]). Since very little is known about non-critical systems, one tries to examine the equations in the ultraviolet limit, which corresponds to $r \longrightarrow 0$, where the underlying field theory should become a CFT. The central charge is related to the vacuum bulk energy, and is given by

$$c(r) = \frac{3r}{\pi^2} \sum_{a=1}^{N} M_a \int_{-\infty}^{\infty} \cosh\beta \log(1 + e^{-\epsilon_a(\beta)}) d\beta \quad . \tag{3.10}$$

Having calculated the central charge one would like to extract the conformal dimension of the perturbing operator. For small r, one expects that $f(r)$ reproduces the behaviour predicted by conformal perturbation theory, which in terms of $c(r)$ reads as

$$c(r) = c - \frac{3f_0}{\pi} r^2 + \sum_{k=1}^{\infty} f_k r^{yk} \quad . \tag{3.11}$$

The exponent y is related to the conformal dimension of the perturbing field Δ by $y = 2(1 - \Delta)$ if the theory is unitary and by $y = 4(1 - \Delta)$ if it is non-unitary. The coefficients are related to correlation functions of the CFT [87, 46], and even if one cannot read them off directly, this is an ultimate important check of the theory.

Note that the application chosen is not a limitation of the use of the program. Also non-diagonal S-matrices (see [88]) can be treated, since once one has diagonalized the transfer-matrix also in that case the numerical problem reduces to solving (3.8). Further quantities to measure can simply be added, and also one can study any range of r, being a parameter in the input-data.

3.2 Multilevel Methods for Integral Equations

The system of non linear Fredholm integral equations (3.8) has been solved using iterative methods [87, 46]. Even if these methods provide a satisfactory solution in terms of accuracy, the number of iterations and corresponding computer process time (CPU) required to reach a specified precision can become excessively large as the number of grid points n increases. A simple one level relaxation would require $O(n^2 \log n)$ operations. With a multilevel solution technique the computing time for integral equations is reduced to $O(n^2)$ [37]. But in particular cases it is possible to reduce to $O(n \log n)$ operations [17]. In any case this justifies the extra effort in programming.

3.2.1 The Numerical Problem

The first point to be analyzed for the discretization of the TBA equations is the truncation of the domain of integration. This is in principle possible because away of the origin the solution behaves like $\epsilon_a(\beta) \simeq rM_a \cosh\beta$. Therefore the integrand vanishes very rapidly. Actually because of this rapid decay we fix the size of the numerical domain so that at the boundaries the value of the integrand is of the order of the zero machine's precision (10^{-15}). And the corresponding truncation error is far below the error estimates for the quadrature rule used.

Now we define the numerical problem. In discretizing the TBA equations, we use the trapezoidal rule [2] on a grid with mesh size h so that our system yields

$$\epsilon_a(\beta) = rM_a \cosh\beta + \frac{h}{2\pi} \sum_{b=1}^{N} \sum_{\beta' \in \Omega_h} w(\beta') \varphi_{ab}(\beta - \beta') \log(1 + e^{-\epsilon_b(\beta')}) \ , \qquad (3.12)$$

$a = 1, 2, \ldots, N$, $\beta \in \Omega_h$, where Ω_h is the set of grid points with grid spacing h. The weights are $w(\beta) = 1$ unless on the boundary where $w(\beta) = 1/2$. Let us introduce a sequence of grids with mesh sizes $h_1 > h_2 > \ldots > h_M$, with $h_{\ell-1} = 2h_\ell$. The system (3.12) with discretization parameter h_ℓ will be denoted as

$$\epsilon_a^\ell = K_{ab}^\ell(\epsilon_b^\ell) + f_a^\ell \ , \quad a = 1, 2, \ldots, N \ , \qquad (3.13)$$

where a summation over b is intended and where

$$K^{\ell}_{ab}(\epsilon)(\beta) = \frac{h_\ell}{2\pi} \sum_{\beta' \in \Omega_{h_\ell}} w(\beta')\varphi_{ab}(\beta - \beta') \log(1 + e^{-\epsilon(\beta')}) \ . \tag{3.14}$$

3.2.2 Iterative Schemes and Local Mode Analysis

In the theoretical investigation of a multilevel algorithm the preferred relaxation method is in many cases the Jacobi iteration. This iterative scheme has a natural counterpart in case of integral equations: the Picard iteration (let us consider $N = 1$ and omit the particle index)

$$\epsilon^{(\nu+1)} = K(\epsilon^{(\nu)}) + f \ . \tag{3.15}$$

The purpose of this iteration is to smooth the error and in principle it is not necessary that (3.15) be a convergent iteration [38]. However, a good convergent iteration would be better. Therefore, since the convergence of (3.15) is very slow, a Gauss-Seidel like iteration is used in practice [73]. To lighten the notation we denote the value of each function, let us say $\Phi(\beta)$, at the point $\beta_j = jh$ simply by Φ_j. In particular $\varphi_{kj} \equiv \varphi(\beta_k - \beta_j)$. With the above notation the GS iterative method reads as

$$\begin{aligned} \epsilon_k^{(\nu+1)} &= rM \cosh\beta_k + \frac{h}{2\pi} \sum_{j<k} w_j \varphi_{kj} \log(1 + e^{-\epsilon_j^{(\nu+1)}}) + \\ &+ \frac{h}{2\pi} \sum_{j \geq k} w_j \varphi_{kj} \log(1 + e^{-\epsilon_j^{(\nu)}}) \ , \quad \beta_k \in \Omega_h \ . \end{aligned} \tag{3.16}$$

Now, using local mode analysis [15] (see also [63]), we investigate this iterative scheme for the integral equation. Then suppose the problem lies on an infinite grid $G^h = \{jh, \ j \in \mathbb{Z}\}$. On this grid the solution error is defined by $e_k^{(\iota)} = \epsilon_k^{(\iota)} - \epsilon_k$, $\iota = \nu, \nu+1$. Subtract (3.16) from (3.12) ($a = 1$, $N = 1$). Then we obtain

$$e_k^{(\nu+1)} - h \sum_{j<k} \omega_j \varphi_{kj} \log(\frac{1 + e^{-\epsilon_j^{(\nu+1)}}}{1 + e^{-\epsilon_j}}) - h \sum_{j \geq k} \omega_j \varphi_{kj} \log(\frac{1 + e^{-\epsilon_j^{(\nu)}}}{1 + e^{-\epsilon_j}}) = 0 \ .$$

Under the assumption that the error $|e| << 1$, is easy to prove that

$$\log(\frac{1 + e^{-\epsilon_j^{(\iota)}}}{1 + e^{-\epsilon_j}}) \simeq -\frac{e^{-\epsilon_j}}{1 + e^{-\epsilon_j}} e_j^{(\iota)} \ , \quad \iota = \nu, \nu + 1 \ .$$

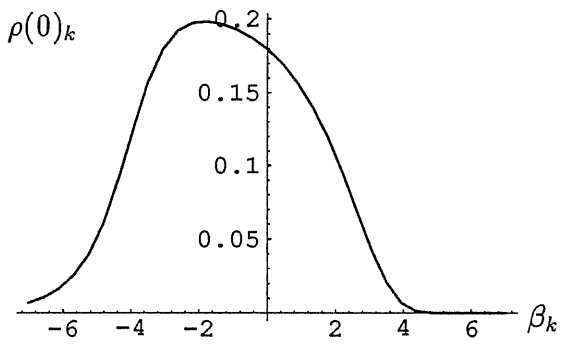

Figure 3.1: A plot of the values of the maximum of $\rho(\theta)_k$ versus the spatial coordinate β_k.

Now assume the decomposition $e_j^{(\iota)} = \sum_\theta \mathcal{E}_\theta^{(\iota)} e^{i\theta j}$ and finally obtain the reduction factor of the component θ:

$$\mu(\theta)_k = |\frac{\mathcal{E}_\theta^{(\nu+1)}}{\mathcal{E}_\theta^{(\nu)}}| = \frac{|h\sum_{j\geq k}\omega_j\varphi_{kj}\frac{e^{-\epsilon_j}}{1+e^{-\epsilon_j}}e^{i\theta(j-k)}|}{|1+h\sum_{j<k}\omega_j\varphi_{kj}\frac{e^{-\epsilon_j}}{1+e^{-\epsilon_j}}e^{i\theta(j-k)}|} \quad . \tag{3.17}$$

The dependence of the reduction factor on k is a consequence of the particular iterative method used. This can be seen in (3.16) where the numbers of variables involved in the two summations depend on the position kh of the variable is being changed. For this reason the above reduction factor must be analyzed at any k, which is simply done by a numerical investigation. So let us consider the case $r = 0.1$, $h_M = 10^{-2}$, at the given r corresponds a size of the numerical domain of 14 (centered in the origin) which results by assuming that $\epsilon_j = rM\cosh\beta_j$, ($M_1 = 1$). Notice that for high frequencies the factor $e^{i\theta(j-k)}$ changes rapidly in sign (that is the real and the coefficient of the imaginary part). Therefore the summations on both numerator and denominator of (3.17) tend to be small, and consequently $\mu(\theta)_k$. Actually this dependence on θ is confirmed by numerical calculations which show for any k that $\max\{\mu(\theta)_k,\ 0 \leq \theta \leq \pi\} = \mu(0)_k$ (see for example fig 3.2). However since we look for the worstest value of the reduction factor we have to compute it for all values of k. In this way we obtain that the largest value of $\mu(0)_k$ occurs whenever $\beta_k \sim 0$, as it results also from fig.3.1.

Then, the prediction of LMA regarding the convergence rate of the Gauss-Seidel iteration is $\rho_{GS}^* = \max\{\rho(0)_k,\ k \in \mathbb{Z}\} = \rho(0)_{k^*}$. From fig 3.1 we see that this value is approximated by 0.2. Finally the functional dependence on θ of $\rho(\theta)_{k^*}$ is reported in

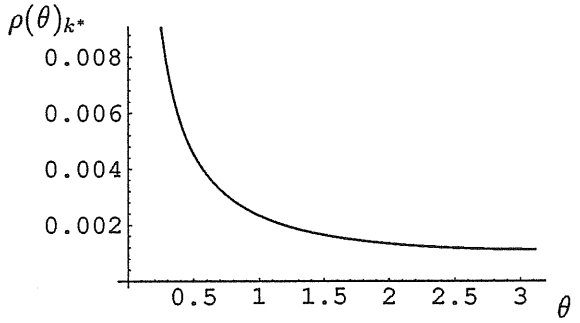

Figure 3.2: A plot of the values of $\rho(\theta)_{k^*}$ with $0 \leq \theta \leq \pi$. The function decreases monotonically from $\rho(0)_{k^*} = 0.2$.

figure 3.2.

Hence a good approximation of the smoothing factor is obtained

$$\mu = \max\{\rho(\theta)_{k^*}, \quad \frac{\pi}{2} \leq \theta \leq \pi\} \sim 1.6 \cdot 10^{-3} \quad . \tag{3.18}$$

This result allows us to estimate the error reduction factor. In if we apply one GS iteration to (3.13) for both pre- and post-smoothing, the resulting convergence factor is given by

$$\rho^* = \mu^{\nu_1+\nu_2} \sim 10^{-6} \quad . \tag{3.19}$$

3.2.3 The FAS-FMG Algorithm

In this section we describe briefly the particular code we have used in [5]. The smoothing iteration used is the GS method described above. Because the problem is non linear a FAS approach is followed. Moreover, as explained in the introduction, we use the FAS scheme in combination with the nested iteration idea. Then we obtain a very efficient FAS-FMG algorithm. We recall the FAS method as it applies to integral equations. Suppose to apply one sweep of (3.16), and to obtain the approximated solutions $\tilde{\epsilon}_a^\ell, \quad a = 1, 2, \ldots, N$. We then transfer them onto the next coarser level, $\tilde{\epsilon}_a^{\ell-1} = \hat{I}_\ell^{\ell-1}\tilde{\epsilon}_a^\ell$, where $\hat{I}_\ell^{\ell-1}$ is a straight injection. The coarse grid equations become

$$\hat{\epsilon}_a^{\ell-1} = K_{ab}^{\ell-1}(\hat{\epsilon}_b^{\ell-1}) + \hat{f}_a^{\ell-1} \quad , \quad a = 1, 2, \ldots, N \quad , \tag{3.20}$$

	$h_M = 0.02$			$h_M = 0.01$	
Iter. (ν)	*Residual*	*Obs. red.* ($\tilde{\rho}_\nu$)	*Iter.* (ν)	*Residual*	*Obs. red.* ($\tilde{\rho}_\nu$)
1	$0.92 \cdot 10^{-5}$	-	1	$0.27 \cdot 10^{-5}$	-
2	$0.11 \cdot 10^{-10}$	$0.66 \cdot 10^{-7}$	2	$0.95 \cdot 10^{-12}$	$0.58 \cdot 10^{-8}$
3	$0.49 \cdot 10^{-14}$	$0.14 \cdot 10^{-5}$	3	$0.24 \cdot 10^{-14}$	$0.95 \cdot 10^{-6}$
		$\tilde{\rho} = 0.31 \cdot 10^{-6}$			$\tilde{\rho} = 0.74 \cdot 10^{-7}$

Table 3.1: The FAS method.

where

$$\hat{f}_a^{\ell-1} = I_\ell^{\ell-1} f_a^\ell + \tilde{\epsilon}_a^{\ell-1} - K_{ab}^{\ell-1}(\tilde{\epsilon}_b^{\ell-1}) - I_\ell^{\ell-1}(\tilde{\epsilon}_a^\ell - K_{ab}^\ell(\tilde{\epsilon}_b^\ell)) \ , \tag{3.21}$$

and with $I_\ell^{\ell-1} = \hat{I}_\ell^{\ell-1}$. Having obtained the solution of the FAS equation $\hat{\epsilon}_a^{\ell-1}$, the difference $\tilde{\epsilon}_a^{\ell-1} - \hat{\epsilon}_a^{\ell-1}$ is the coarse level (CL) correction to the fine-grid solution

$$\tilde{\epsilon}_a^\ell \leftarrow \tilde{\epsilon}_a^\ell - I_{\ell-1}^\ell(\tilde{\epsilon}_a^{\ell-1} - \hat{\epsilon}_a^{\ell-1}) \ , \tag{3.22}$$

$a = 1, 2, \ldots, N$, and $I_{\ell-1}^\ell$ is a coarse-to-fine cubic interpolation operator. Finally we perform one relaxation at level ℓ, in order to smoothen errors coming from the interpolation procedure. To solve the system of equations (3.13) we employ a coarse level correction recursively, i.e. equation (3.20) is itself solved by iteration sweeps combined with a further CL correction.

In table 3.1 and 3.2, we give the norm of the residual $\tilde{r}_a(\beta) = (\epsilon_a - K_{ab}(\epsilon_b) - f_a)(\beta)$, $\beta \in \Omega_{h_M}$ defined in (3.23), and the *observed reduction factors*

$$\tilde{\rho}_\nu = \|\epsilon^{(\nu+1)} - \epsilon^{(\nu)}\| / \|\epsilon^{(\nu)} - \epsilon^{(\nu-1)}\| \ ,$$

with $\|\cdot\|$ the maximum norm. We also give the *mean reduction factor* [73]

$$\tilde{\rho} = \{\prod_{\nu=1}^{k} \tilde{\rho}_\nu\}^{1/k} \ .$$

The numerical data presented in table 3.1 and 3.2 show clearly that the predictions of local mode analysis about the convergence of both GS and FAS methods are very sharp. Further we have to mention that the ML convergence theory for integral equations [38]

	$h_M = 0.02$			$h_M = 0.01$	
Iter. (ν)	*Residual*	*Obs. red.* ($\tilde{\rho}_\nu$)	*Iter.* (ν)	*Residual*	*Obs. red.* ($\tilde{\rho}_\nu$)
1	0.40	-	1	0.57	-
2	$0.47 \cdot 10^{-1}$	0.100	2	$0.66 \cdot 10^{-1}$	0.099
3	$0.60 \cdot 10^{-2}$	0.135	3	$0.84 \cdot 10^{-2}$	0.135
4	$0.79 \cdot 10^{-3}$	0.145	4	$0.10 \cdot 10^{-2}$	0.144
5	$0.97 \cdot 10^{-4}$	0.137	5	$0.13 \cdot 10^{-3}$	0.136
		$\tilde{\rho} = 0.128$			$\tilde{\rho} = 0.127$

Table 3.2: The Gauss-Seidel method.

proves that $\tilde{\rho} = O(h^\delta)$, $\delta > 0$. In fact, comparing the results obtained with $h_M = 0.02$ and $h_M = 0.01$, we see that the mean reduction factor has been decreased by a factor of 4, that is $\delta = 2$.

In the FMG cycle the initial approximation on the coarsest level is taken $\epsilon_a = rM_a \cosh\beta$. Then a FMG interpolation and the subsequent FAS scheme on the following finer level is applied, recursively. Until the finest level is reached.

We compare the performance of the FMG and of the Gauss-Seidel iterative scheme in terms of CPU time in fig. 3.3, there the different initial residual error is due to the set up of the initial approximated solution, on the finest level, by the FMG cycle. As a norm for the residuals we use

$$\| \tilde{r} \|_M = \max_{1 \le a \le n} \sqrt{\sum_{\beta \in \Omega_{h_M}} \tilde{r}_a(\beta)^2} \ . \tag{3.23}$$

In order to outline how the multilevel algorithm becomes important as the number of particles increases we give in table 3.3 the CPU time required by the two methods to solve the discretized problem to a value of the residual norm $\|\tilde{r}\|_M \le 10^{-14}$.

For any scaling length r we use an initial approximation which behaves like $rM_a \cosh\beta$, wherefrom the program determines the numerical boundary at which the kernels vanish and verifies that the conditions for the existence of (at least) one solution given by the Schauder's fixed point theorem are satisfied [67]. Having determined the size of the

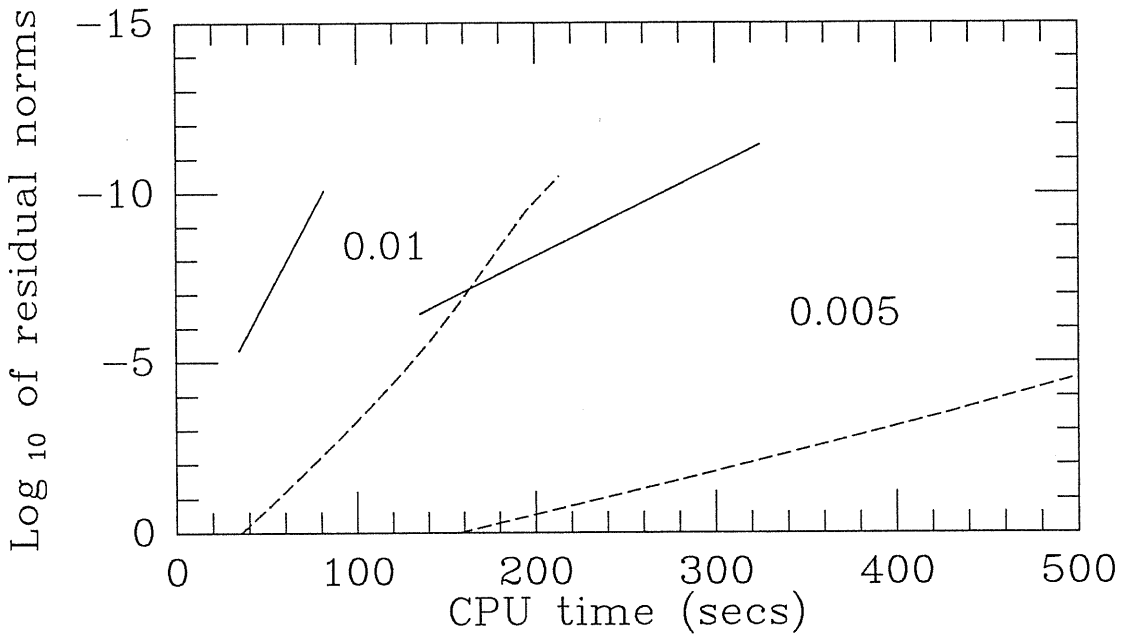

Figure 3.3: Evolution of residual error norm with CPU time for a 1-particle system at $r = 0.1$ for different h_M: solid line for FMG, dashed line for GS iteration.

numerical domain for a given r the number of levels M is set such that the finest level has the required mesh-size h_M.

3.2.4 Numerical Results for Diagonal Scattering Theories

As already mentioned in the introduction our program [2] consists of two parts: the core, which resolves the TBA equations (3.8) and the periphery, which on the one hand constructs the kernel and the initial solution, and on the other extracts from the solution the central charge, the dimension y and the coefficients f_i of the perturbation expansion (3.11). In [5] we specifically designed the numerical code for diagonal scattering theories, that is we were concerned with scalar S matrices of the form

$$S_{ab} = \prod_i f(\alpha_{ab}^i) \quad , \quad i = 1, \ldots, N_{ab} \quad , \quad a, b = 1, \ldots, N \tag{3.24}$$

with

$$f(\alpha) = \frac{\sinh \frac{1}{2}(\beta + i\pi\alpha)}{\sinh \frac{1}{2}(\beta - i\pi\alpha)}$$

[2]This program is obtainable from: CPC Program Library, Queen's University of Belfast, N. Ireland.

no. equations	*CPU time (secs)*	
	Relax	multilevel
1	4	3
2	34	22
3	508	331
4	1230	712
5	2530	1320

Table 3.3: A comparison of CPU time required to reach a particular value of the norm, for $r = 0.1$, $h_M = 0.1$.

N being the number of particles in the theory and N_{ab} is the number of factors f_x appearing in the S-matrix S_{ab} (for a recent review on this subject and many examples see [60]). The set of the numbers α, and the masses of the theory are sufficient to resolve the TBA equations.

As input-data we need the information about the range of r, the number of particles, the values of α, and the values of the masses of the particles. From the input, the kernel and the initial approximation are automatically defined and used by the multilevel algorithm.

We discuss here the specific structure of the algorithm in terms of a simple example. Consider the S-matrix

$$S_{11} = f_{\frac{2}{5}} f_{\frac{3}{5}}, \quad S_{12} = S_{21} = f_{\frac{1}{5}} f_{\frac{2}{5}} f_{\frac{3}{5}} f_{\frac{4}{5}}, \quad S_{22} = f_{\frac{1}{5}} f_{\frac{4}{5}} (f_{\frac{2}{5}} f_{\frac{3}{5}})^2 \ . \tag{3.25}$$

This scattering matrix has been conjectured to correspond to the (non unitary) minimal model $\mathcal{M}_{2,7}$, i.e. $c = \frac{4}{7}$, perturbed by the field with dimension $\Delta = -\frac{3}{7}$. Since this S-matrix is symmetric one needs to specify only the elements $S_{1,1}, S_{1,2}, S_{2,2}$.

The masses of the two particles are

$$M_1 = 1 \ , \quad M_2 = 2\cos(\frac{\pi}{5}). \tag{3.26}$$

The purpose of our calculations is to verify the correspondence of the results predicted by the scattering theory with those obtained in conformal field theory in the UV limit,

that is $r \to 0$. This verification goes through (3.10) and (3.11), and therefore we need first to solve the TBA equations for a given set of r close to zero. In our computation we take $r = 0.01 + 0.01 \cdot (p-1)$, with $p = 1, \ldots, (p_{max} =)30$. For each r we find the solution $\epsilon_a(\beta)$, $a = 1, 2$, and use (3.10) to compute the corresponding $c(r)$. The resulting list of data is reproduced in OUTPUT.DAT (see below). Then using this data we make use of extrapolation and fitting procedures [81] to extract from (3.11) the important values y and f_i, respectively. We find that in order to get a sensible result for the dimension y it is enough to use $p_{max} \sim 5$. Whereas, if one wants to calculate the coefficients f_i a larger number is required; for example with $p_{max} \geq 10$ one can get f_1 up to an error of $O(10^{-5})$. Clearly the more data is used the better the fit-procedure works and more coefficients can be obtained. It is also clear that this computations require a high accuracy. In this respect the choice $h_M = 10^{-2}$ allows very accurate results.

The following file was produced using the input described in this section:

1. OUTPUT.DAT

```
computed cexact =     .57142857D+00
r=    .100000000D-01  central charge=    .571403656E+00
r=    .200000000D-01  central charge=    .571329513E+00
r=    .300000000D-01  central charge=    .571206952E+00
r=    .400000000D-01  central charge=    .571036718E+00
r=    .500000000D-01  central charge=    .570819520E+00
r=    .600000000D-01  central charge=    .570556042E+00
r=    .700000000D-01  central charge=    .570246948E+00
r=    .800000000D-01  central charge=    .569892886E+00
r=    .900000000D-01  central charge=    .569494492E+00
r=    .100000000D+00  central charge=    .569052390E+00
r=    .110000000D+00  central charge=    .568567192E+00
r=    .120000000D+00  central charge=    .568039505E+00
r=    .130000000D+00  central charge=    .567469925E+00
r=    .140000000D+00  central charge=    .566859042E+00
```

```
 r=    .150000000D+00  central charge=    .566207440E+00
 r=    .160000000D+00  central charge=    .565515695E+00
 r=    .170000000D+00  central charge=    .564784379E+00
 r=    .180000000D+00  central charge=    .564014059E+00
 r=    .190000000D+00  central charge=    .563205295E+00
 r=    .200000000D+00  central charge=    .562358645E+00
 r=    .210000000D+00  central charge=    .561474660E+00
 r=    .220000000D+00  central charge=    .560553889E+00
 r=    .230000000D+00  central charge=    .559596875E+00
 r=    .240000000D+00  central charge=    .558604159E+00
 r=    .250000000D+00  central charge=    .557576277E+00
 r=    .260000000D+00  central charge=    .556513761E+00
 r=    .270000000D+00  central charge=    .555417143E+00
 r=    .280000000D+00  central charge=    .554286946E+00
 r=    .290000000D+00  central charge=    .553123695E+00
 r=    .300000000D+00  central charge=    .551927909E+00
error in extrapolation  .3365E-09
estimated exponent        .285714287D+01
theoretical exponent      .285714286D+01
estimated dimension of the corresponding operator
for a     unitary theory: DELTA=     .285714D+00
for a non-unitary theory: DELTA=    -.428571D+00
 fitted f_i
f( 1)= .9643967331341316D-01
f( 2)=-.1538311769518447D-02
f( 3)= .6222166295705919D-04
f( 4)=-.3197939259001748D-05
 chi-square value of the fitting=  .5164E-29
 total cpu time (secs)    .113E+06
```

3.2.5 Conclusion

We presented a multilevel scheme for the resolution of the thermodynamic Bethe ansatz equations. The TBA is a means to describe the finite temperature effects of relativistic factorized scattering theories. The numerical code described is specifically designed for theories having a scalar S-matrix. These theories exhibit a unique form, and the only input needed in order to carry out the TBA are the locations of the poles and zeros of the single S-matrix elements.

The program calculates the central charge and in the ultra-violet limit the dimension of the perturbing field and the coefficients of the perturbation expansion. These are the most crucial tests in verifying a conjectured S-matrix. However other physical quantities could be computed, as for example for magnetic systems the moments of the total magnetization, or the convergence-region of the perturbation series in (3.11) [87, 46].

In order to get sensible results for the physical quantities one needs to resolve the integral equations with the highest possible accuracy. This unfortunately renders the calculation extremely time consuming. Therefore the use of an efficient multilevel algorithm gives the possibility to reach high accuracy in the computation together with a sensible reduction of the CPU time, in confrontation with standard iterative techniques.

Chapter 4

Algebraic Multilevel Methods

4.1 Introduction

Suppose one has a problem $\tilde{A}\tilde{u} = \tilde{f}$, with $\tilde{A}$ being a differential operator. Numerically, the resolution of this problem is achieved defining a sequence of approximate solutions u_n, $n \in \mathbb{N}$, to $\tilde{u}$. These again are expressed as solutions of operator equations $A_n u_n = f_n$, $n \in \mathbb{N}$, and A_n and f_n should be approximations to $\tilde{A}$ and $\tilde{f}$ respectively and the solution u_n, $n \in \mathbb{N}$ should (in some sense) approximate the solution $\tilde{u}$ [69]. That is, for any fixed n (called approximation or discretization parameter), a linear or a linearized continuous problem will be translated to a set of linear algebraic equations $Au = f$ (we will omit n from now on) to be solved.

We know that in case of partial differential equations of elliptic type the corresponding linear system of equations (to a given n) can be solved successfully using multilevel methods. Essentially multilevel solution processes consist of a *smoothing* iteration, *transfer* of defects from one "solution level" to another one where the defect problem is easier to solve, and a *prolongation* of the resulting "coarse" solution to the "finer" level. All these components are usually defined taking care of the geometrical features of the problem at hand. Of course, it is not always possible to handle all these geometrical aspects as, for example, when the domain is complex enough or the differential operator presents anisotropies.

In these cases one attempts to construct a multilevel solution process where all op-

erators of relaxation, transfer of defects, and prolongation of corrections are to be automatically defined using entries of the given matrix A. This is the so called *algebraic multilevel* (AML) method [14, 71]. However, in those cases where a discretized differential problem is considered, it is difficult to distinguish among the standard ML approach[1] and the algebraic one. It is a matter of definition if a given prolongation or restriction has been decided based on the geometrical features of the problem to be solved or the (corresponding) algebraic properties. Nevertheless a distinction can be made since the two approaches, even if close together, lead (in principle) to different operators.

From the computational point of view an algebraic approach is followed in order to define blackbox ML solvers, where all components are set up by the code itself. Another aspect of AML is to stimulate the investigation of multilevel methods from an algebraic point of view. Actually this analysis leads to the formulation of very powerful methods with large applicability [23, 89].

The first papers on AML were devoted to investigate especially the case of symmetric, positive definite matrices [14, 71]. In these works symmetry and positive definiteness of A are used to define norms in terms of which it is possible to characterize the smoothing property of a given relaxation operator and to prove convergence of multilevel iteration. This analysis using suitable norms requires only general properties of the multilevel operators. On the other hand, this generality leaves the problem of how to define explicitly these operators in order to construct a multilevel algorithm.

So the first purpose of our investigation [7], presented here, is to analyze exhaustively and to define explicitly the ML operators mentioned above. Secondly we want to show how these components can be recovered defining a suitable twolevel substitution method which results to be very similar to the standard (linear) twolevel method. This is indeed what we commence here considering the "model class" of irreducible tridiagonal, diagonally dominant $\mathcal{M}$-matrices presented in section 4.1.1. Later we will show how these results obtained for the above class extend to more general classes of matrices.

In general, tridiagonal matrices arise in connection with elliptic one-dimensional prob-

[1]That is the formulation of a multilevel procedure where the coarser grids and the corresponding discrete problem are obtained by doubling the mesh size.

lems, here we choose them since they have many properties which turn out to be very useful for our discussion. Obviously, one can notice that the computational work for the resolution of the algebraic problem related with these matrices is done in only $O(n)$ operations using Gaussian elimination [76]. However the purpose of this study is not to give a competitive tool to solve algebraic problems related to the above class, but to do a careful algebraic investigation of the multilevel process starting from a simple model problem. One will recognize that many of the obtained results are generalizations of those obtained studying the simplest one-dimensional Dirichlet problem [37], using discrete Fourier analysis. The purpose is indeed the same: a simple introductory problem where all the components of a multilevel algorithm are showed. The difference is that here we cannot use any property which comes from geometric considerations so that, for example, the prolongation and restriction operators cannot be immediately defined.

However, as in the 'geometric' case, the justification for the entire algebraic multilevel process should be based on a careful analysis of the iterative methods used. So we analyze the algebraic properties of Jacobi and Gauss-Seidel iterations. For the first one we give some necessary conditions to the sign structure of the elements of the corresponding eigenvectors. This allows to recover the notion of frequency in a pure algebraic setting. The analysis is also valid for the eigenvectors of the Gauss-Seidel iteration through the explicit relation between the eigenvectors of the two matrices which we report here. However, the second eigenproblem is more involved. We show that the GS iteration matrix is defective and the formalism of principal vectors must be introduced. Then we show some properties of these generalized eigenvectors and we prove the convergence of the iteration also with respect to them.

In section 4.2.2 we use the above analysis to deduce a prolongation operator. Then we recognize that this operator describes a substitution procedure which, in turn, is used to define the corresponding restriction. Moreover, we find that these operators can be expressed in terms of the splitting formalism, which was introduced in order to construct iterations like Jacobi or Gauss-Seidel. Using this formalism we are also able to define the coarse problem and the consequent coarse level correction. The algebraic definition of a twolevel method in terms of the splitting is completed proving that also the choice of the

coarse variables is to be done in terms of the splitting. In this way we achieve two important results. Firstly we are able to explain the standard multilevel procedure in terms of a general algebraic formulation. Secondly this approach simply allows the computation of sharp bounds for the rate of convergence of multilevel iterations. These results first derived for the tridiagonal case, will be proved to be valid for more general linear systems, as for example those which arise from the 5-point and 7-point discretization of Dirichlet boundary value problems in two and three dimensions, respectively.

4.1.1 The Model Class $\mathcal{A}$

In this section we start defining a model class $\mathcal{A}$ of matrices. We will use a simple notation which does not take into account the concept of levels to be introduced later. Here useful definitions will be stated and some properties relative to the matrix $A \in \mathcal{A}$ will be shown.

Consider the tridiagonal system of algebraic linear equations

$$Au = f \quad , \quad u, f \in \Omega \subseteq \mathbb{R}^n \quad . \tag{4.1}$$

We say that A will belong to the model class $\mathcal{A}$ if $A \in L(\Omega)$ is a $n \times n$ tridiagonal matrix with positive diagonal elements such that $a_{ii} \geq \sum_{j \neq i} |a_{ij}|$, $i = 1, \ldots, n$ (where strict inequality holds for at least one i), and negative off-diagonal elements $a_{ij} < 0$, $i \neq j$, $|i - j| = 1$. Then we recall some properties [76], introduced also to study Burgers' equation, which will become very useful in proceeding our discussion

Definition 3 *A matrix $A \in L(\Omega)$ is irreducible if and only if for any two distinct indices $1 \leq i, j \leq n$, there is a sequence of non zero elements of A of the form*

$$\{a_{ii_1}, a_{i_1 i_2}, \ldots, a_{i_m j}\} \quad . \tag{4.2}$$

Obviously matrices $A \in \mathcal{A}$ are irreducible because for any $i < j$ there is the sequence $\{a_{ii+1}, a_{i+1i+2}, \ldots, a_{j-1j}\}$ of negative (non zero) elements, and if $j < i$ one has $\{a_{jj+1}, a_{j+1j+2}, \ldots, a_{i-1i}\}$. The other property required is that of *irreducible diagonal dominance* stated in

Definition 4 *A matrix $A \in L(\Omega)$ is diagonally dominant if*

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \quad , \quad i = 1, \ldots, n \quad , \tag{4.3}$$

and irreducible diagonally dominant if it is irreducible, diagonally dominant, and strict inequality holds for at least one i.

The class $\mathcal{A}$ defined above seems to be very restrictive, nevertheless the discretization of (one-dimensional) boundary value problems leads naturally to it. In particular, one notices that the above class can be referred simply as the class of irreducible, tridiagonal, diagonally dominant $\mathcal{M}$-matrices [62].

4.2 Analysis of Multilevel Components

4.2.1 Smoothing Iterations

In section 1.2.1 we showed how an iterative method can be defined through the splitting $A = D - L - U$. Denoting with $D = \mathrm{diag}(a_{11}, a_{22}, \ldots, a_{nn})$, and with $-L$ and $-U$ the strictly lower and upper part of A, one constructs two well known iterations

$$u^{(\nu+1)} = (I - D^{-1}A)u^{(\nu)} + D^{-1}f \quad , \quad \text{Jacobi iteration} \tag{4.4}$$

and

$$u^{(\nu+1)} = (I - (D-L)^{-1}A)u^{(\nu)} + (D-L)^{-1}f \quad , \quad \text{Gauss-Seidel iteration} \quad . \tag{4.5}$$

A special case of (4.4) is the *damped Jacobi iteration*: $u^{(\nu+1)} = (I - \omega D^{-1}A)u^{(\nu)} + \omega D^{-1}f$, where ω is a damping constant $0 < \omega \leq 1$.

Because A is assumed to be irreducible diagonally dominant, both the (even damped) Jacobi and the Gauss-Seidel iteration matrices, which we denote with $M_J(\omega)$ (M_J if $\omega = 1$) and M_{GS} respectively, have a spectral radius smaller than one [2]. This means, they satisfy a global condition for iterations to converge to $A^{-1}f$ starting from any initial vector $u^{(0)}$, as stated by theorem 4. In general it is a matter of iterative convergence

[2]The spectral radius of an iteration matrix is also called the rate of convergence.

theory to define iterative matrices M whose spectral radius is as small as possible. On the other hand, a multilevel algorithm necessitates an iterative method which solves quickly only for some solution components, so that the remaining ones can be seen as components of a smaller algebraic problem.

For this purpose we need to analyze the eigenvalues and the eigenvectors of the iteration matrices relative to (4.4) and (4.5). First we introduce the notion of *consistently ordered matrices* [62]

Definition 5 *A matrix $A \in L(\Omega)$ is consistently ordered if the eigenvalues of the matrices*

$$B(\alpha) = \alpha D^{-1}L + \alpha^{-1}D^{-1}U \tag{4.6}$$

for $\alpha \neq 0$ are independent of α.

This property allows some useful result stated in the following [84]

Theorem 4 *Let $A \in L(\Omega)$ be a consistently ordered matrix. Then*

1. *If μ is an eigenvalue of M_J of multiplicity m, then $-\mu$ is also an eigenvalue of M_J of multiplicity m;*
2. *λ satisfies*

$$\lambda^2 = \mu^2\lambda \tag{4.7}$$

for some eigenvalue μ of M_J if and only if λ satisfies

$$\lambda = \mu\lambda^{\frac{1}{2}} \tag{4.8}$$

for some eigenvalue μ of M_J;

3. *If λ satisfies either, and hence both of the relations (4.7) and (4.8), then λ is an eigenvalue of M_{GS};*
4. *If λ is an eigenvalue of M_{GS}, then there exists an eigenvalue μ of M_J such that (4.7) and (4.8) hold.*

In particular, the above theorem states that the set of eigenvalues of M_{GS} includes the number zero together with the numbers $\mu_1^2, \mu_2^2, \ldots, \mu_q^2$ where $\pm\mu_1, \pm\mu_2, \ldots, \pm\mu_q$ are the nonzero eigenvalues of M_J with multiplicity $m_1, m_2, \ldots, m_q$. Moreover $\rho(M_{GS}) = \rho(M_J)^2$.

We recall [62] that any tridiagonal matrix A for which $a_{ii+1}a_{i+1i} > 0$, $i = 1, \ldots, n-1$, can be transformed into a symmetric, tridiagonal matrix B, using a diagonal similarity transformation. The elements of B are given by $b_{ii} = a_{ii}$ and $b_{ii+1} = b_{i+1i} = \sqrt{a_{ii+1}a_{i+1i}}$. This fact make easy to prove that every $A \in \mathcal{A}$ is consistently ordered [62], and then the corresponding Jacobi iteration matrix has a symmetric spectrum. Moreover, M_J can be reduced to a diagonal form, so one can always select a set of eigenvectors which span the whole n-space and can therefore be used as a base in which to express an arbitrary vector [83].

Let us write the eigenvalue problem for the damped Jacobi iteration, $M_J(\omega)v^k = \tilde{\mu}_k v^k$, where $\tilde{\mu}_k$ and v^k, $k = 1, \ldots, n$ are the eigenvalues and eigenvectors respectively. Notice that $M_J(\omega) = (1-\omega)I + \omega M_J$, this means that $\tilde{\mu}_k = 1 - \omega + \omega\mu_k$, and that M_J and $M_J(\omega)$ have the same eigenvectors which do not depend on ω.

Hence, we look at the eigenproblem $M_J v^k = \mu_k v^k$ which componentwise reads as

$$\begin{cases} -\frac{a_{12}}{a_{11}} v_2^k = \mu_k v_1^k \ , \\ -\frac{a_{ii-1}}{a_{ii}} v_{i-1}^k - \frac{a_{ii+1}}{a_{ii}} v_{i+1}^k = \mu_k v_i^k \ , \quad i = 2, \ldots, n-1 \ , \\ -\frac{a_{nn-1}}{a_{nn}} v_{n-1}^k = \mu_k v_n^k \ . \end{cases} \tag{4.9}$$

These equations lead to the following considerations:

1. $\mu_k = 0$ (which occurs when n is an odd integer). Then (4.9) implies that $v_i^k = 0$, $i = 2, 4, \ldots, n-1$ and the sequence $\{v_i^k\}_{i=1,3,\ldots,n}$ is alternating in sign;
2. $\mu_k > 0$. Then (4.9) implies that the sequence $\{v_i^k\}_{i=1,2,\ldots,n}$ does not admit sign changes as $+,-,+$ or $-,+,-$;
3. $\mu_k < 0$. Then (4.9) implies that the sequence $\{v_i^k\}_{i=1,2,\ldots,n}$ does not admit permanence of sign as $+,+,+$ or $-,-,-$.

Then a comparison with geometric multilevel tells us that $\mu_k > 0$ corresponds to the

case of low frequency eigenvectors and $\mu_k \leq 0$ is the case of high frequency eigenvectors. This allows us to recover the formalism of geometric multilevel: a suitable chosen damping parameter provides an iteration which reduces better the HF components of the solution error. That is, it acts as a smoother. For example, in our case, with $\omega = \frac{1}{2}$ the rate of convergence of $M_J(\frac{1}{2})$ restricted to the space spanned by the HF eigenvectors v^k is at least $\frac{1}{2}$. This means that after few times damped Jacobi iterations the error solution consists essentially of those components corresponding to $\mu_k > 0$. We will see later how to use this result in order to define a proper prolongation operator.

Now we are ready to consider the eigenproblem for the Gauss-Seidel iteration. We know that $\lambda = 0, \mu_1^2, \mu_2^2, \ldots, \mu_q^2$, are real eigenvalues of M_{GS}. Notice that we can write explicitly the eigenproblem in another form

$$(\lambda_k D^{-1}L + D^{-1}U)w^k = \lambda_k w^k \quad , \tag{4.10}$$

where w^k is the eigenvector corresponding to the eigenvalue λ_k. All eigenvalues λ_k are obtained solving the equation $\det(\lambda D^{-1}L + D^{-1}U - \lambda I) = 0$. In case of tridiagonal matrices of our model class it is easy to see that the characteristic equation, is of the form

$$\lambda^n + c_1\lambda^{n-1} + c_2\lambda^{n-2} \ldots + c_{\tilde{p}}\lambda^{n-\tilde{p}} = 0 \quad , \tag{4.11}$$

where c_ℓ, $\ell = 1, \ldots, \tilde{p}$ are nonzero constant. If n is an even integer we have $\tilde{p} = \frac{n}{2}$, whereas if n is an odd integer we have $\tilde{p} = \frac{n-1}{2}$. Then the zero eigenvalue has multiplicity $p = \frac{n}{2}$ or $p = \frac{n+1}{2}$ if n is even or odd respectively. Then the eigenvalues of M_{GS} are $\mu_1^2, \mu_2^2, \ldots, \mu_q^2$ and the zero eigenvalue with multiplicity p.

We continue our analysis investigating the eigenvectors of M_{GS}. The system of n equations (4.10) reads as

$$\begin{cases} -\frac{a_{12}}{a_{11}} w_2^k = \lambda_k w_1^k \quad , \\ -\lambda_k \frac{a_{ii-1}}{a_{ii}} w_{i-1}^k - \frac{a_{ii+1}}{a_{ii}} w_{i+1}^k = \lambda_k w_i^k \quad , \quad i = 2, \ldots, n-1 \quad , \\ -\lambda_k \frac{a_{nn-1}}{a_{nn}} w_{n-1}^k = \lambda_k w_n^k \quad . \end{cases} \tag{4.12}$$

Multiplying the ith equation of (4.12) by $\lambda_k^{-\frac{i+1}{2}}$ and using the relation $v_i^k = \lambda_k^{-\frac{i}{2}} w_i^k$, one recovers (4.9) with $\mu_k = \lambda_k^{\frac{1}{2}}$. Then the eigenvectors corresponding to the nonzero eigenvalues λ_k are given by $w_i^k = \lambda_k^{\frac{i}{2}} v_i^k$ where v^k is the eigenvector of the Jacobi iteration corresponding to the eigenvalue $\mu_k = \lambda_k^{\frac{1}{2}}$. Because of this relation we can deduce that the corresponding sequence $\{w_i^k\}_{i=1,2,\ldots,n}$ does not admit sign changes as $+,-,+$ or $-,+,-$.

The eigenvectors corresponding to the eigenvalue zero are the nonzero vectors in the null space of the matrix M_{GS}. In order to find these vectors we look at all entries of M_{GS} which we give in the following

Remark 1 *All $n \times n$ entries m_{ij} of the Gauss-Seidel iteration matrix M_{GS}*

$$\begin{aligned}
m_{i1} &= 0 \ , \\
m_{ii+1} &= -\frac{a_{ii+1}}{a_{ii}} \ , \\
m_{ii+k} &= 0 \ , \quad 2 \leq k \leq n-i \ , \\
m_{ii} &= \frac{a_{i-1i} a_{ii-1}}{a_{ii} a_{i-1i-1}} \ , \quad i = 2, \ldots, n \ , \\
m_{ii-k} &= (-1)^k \frac{a_{ii-1} a_{i-1i-2} \cdots a_{i-k-1i-k}}{a_{ii} a_{i-1i-1} \cdots a_{i-k-1i-k-1}} \ .
\end{aligned}$$

We find that the null space of M_{GS} consists only of the eigenvector $w = (1, 0, 0, \ldots, 0)$. Then the geometric multiplicity of the zero eigenvalue is only 1 and M_{GS} has fewer than n linearly independent eigenvectors so it is *defective* [85]. However it is still possible [85] to find the so-called *principal vectors* (p.v.) $y^1, y^2, \ldots, y^{p-1}$ (not unique), so that

$$\left\{ \begin{array}{l} M_{GS} w = 0 \ , \\ M_{GS} y^1 = w \ , \\ M_{GS} y^k = y^{k-1} \quad k = 2, \ldots, p-1 \ , \end{array} \right. \tag{4.13}$$

where the kth p.v. satisfies $M_{GS}^{k+1} y^k = 0$ and $M_{GS}^k y^k \neq 0$. Then the set of p linearly independent principal vectors $w, y^1, \ldots, y^{p-1}$ together with the $n-p$ eigenvectors w_k cor-

responding to nonzero eigenvalues, span the whole n-space [85].

We prove a lemma which states the sign structure of the principal vectors of M_{GS}.

Lemma 4 *Let y^k, $k = 1, \ldots, p-1$ be the p.v. of M_{GS} and let w as above. Then the y^k can be chosen such that*

$$(\mathrm{sign}(y_i^k))_{i=1,\ldots,n} = (\overbrace{0,\ldots,0}^{k}, \overbrace{+,-,+,\ldots}^{k+1}, \overbrace{0,\ldots,0}^{n-2k-1}) \quad , \tag{4.14}$$

where the nonzero elements of y^k are alternating in sign.

Proof. We prove the lemma by induction.
First consider y^1, that is $k = 1$. From remark 1 one sees immediately that the first element y_1^1 can be chosen to be zero, while $y_2^1 = -\frac{a_{11}}{a_{12}}$, which is positive, and $y_3^1 = -\frac{a_{21}}{a_{23}}$, which is negative. Moreover, one obtains that $y_i^1 = 0$ for $i = 4, \ldots, n$.

Now consider $2 \leq k \leq p-1$, and notice that (4.13) reads componentwise as follows

$$y_i^{k-1} + \frac{a_{ii-1}}{a_{ii}} y_{i-1}^{k-1} = -\frac{a_{ii+1}}{a_{ii}} y_{i+1}^k \quad , \quad i = k, \ldots, 2k-1 \quad . \tag{4.15}$$

Therefore assume that (4.14) is true for y^{k-1}. Immediately from (4.15) one obtains $y_i^k = 0$, $i = 1, \ldots, k$ and $y_i^k = 0$, $i = 2k+2, \ldots, n$. Then the first nonzero element of y^k is $y_{k+1}^k = -\frac{a_{kk}}{a_{kk+1}} y_k^{k-1}$, that is the first nonzero element of y^k has the same positive sign of the corresponding first nonzero element of y^{k-1}. Moreover, by hypothesis y_i^{k-1} and y_{i-1}^{k-1} have opposite sign, then we have

$$\mathrm{sign}(y_i^{k-1}) = \mathrm{sign}(y_i^{k-1} + \frac{a_{ii-1}}{a_{ii}} y_{i-1}^{k-1}) = \mathrm{sign}(-\frac{a_{ii+1}}{a_{ii}} y_{i+1}^k) = \mathrm{sign}(y_{i+1}^k) \quad . \tag{4.16}$$

This means that the nonzero elements of y^k are alternating in sign as those of y^{k-1}. Because (4.14) is true for $k = 1$ the lemma is proved.

For the principal vectors one cannot refer to the rate of convergence because they are not eigenvectors nor combinations of them. However, in that case, it is still possible to compute a "contraction" number with respect to a given norm in $\mathbb{R}^n$. We will use the

maximum norm defined as $\|z\|_\infty = \max_i |z_i|$. With the natural induced matrix norm given by $\|A\|_\infty = \max_{\|x\|_{infty}=1} \|Ax\|_\infty = \max_i \sum_{j=1,n} |a_{ij}|$.

From lemma 4 we know that the two addendum on the left hand side of (4.15) are of the same sign, then in terms of absolute values we have the strict inequality between the nonzero elements of y^k and y^{k-1}, that is $|y_i^{k-1}| \leq (-\frac{a_{ii+1}}{a_{ii}})|y_{i+1}^k|$, $i = k, \ldots, 2k-1$. Denoting with $\sigma = \max_i(-\frac{a_{ii+1}}{a_{ii}})$ we finally obtain

$$\|y^{k-1}\|_\infty \leq \sigma \|y^k\|_\infty \quad . \tag{4.17}$$

In particular, for matrices of class $\mathcal{A}$ such that $a_{ii-1} = a_{ii+1}$ (and $a_{11} \geq 2|a_{12}|$) we recover the well known result obtained with local mode analysis [10] that is $\sigma \leq \frac{1}{2}$. This comparison is appropriate since we have shown that the high oscillating components are represented by the principal vectors, and σ is the corresponding rate of convergence of M_{GS} applied to these vectors. Finally notice that $\sigma = \|D^{-1}U\|_\infty$.

A similar result can be obtained for the defect as follows. For, if $e = \tilde{u} - u$ is the solution error then one has the "defect equation" $Ae = d$. Further let us denote with $d^{(\nu)}$ the defect corresponding to $e^{(\nu)}$. Notice that if M is the iteration matrix for the error then $\tilde{M} = AMA^{-1}$ is the iteration matrix for the defect, that is $\tilde{M}d^{(\nu)} = d^{(\nu+1)}$, and the two matrices M and $\tilde{M}$ are equivalent. This implies that they have the same spectral radius, and therefore there is a correspondence in the convergence estimates.

However we want deeper results about the relation of e and d. Let us consider the simple case when e coincides with an eigenvector of M_J corresponding to the eigenvalue μ. Because $D^{-1}A = I - M_J$ we obtain $Ae = (1-\mu)De$, that is $d = (1-\mu)De$. In the same way, considering e to be coincident with an eigenvector of M_{GS} relative to the eigenvalue $\lambda \neq 0$, it results that $d = \frac{1-\lambda}{\lambda}Ue$. It remains to find the defect corresponding to the principal vectors. Let $e = y^{k-1}$; proceeding as above using (4.13) we obtain $Ay^{k-1} = U(y^k - y^{k-1})$, that is the defect $d_k = U(y^k - y^{k-1})$. Now recall lemma 4 and notice that $M_{GS}(y^k - y^{k-1}) = y^{k-1} - y^{k-2}$ where from one obtains componentwise $|y_i^{k-1} - y_i^{k-2}| \leq (-\frac{a_{ii+1}}{a_{ii}})|y_{i+1}^k - y_{i+1}^{k-1}|$. Multiplying the ith inequality by $-a_{i-1i}$, we obtain the following

$$\|d_{k-1}\|_\infty \leq \sigma' \|d_k\|_\infty \quad , \tag{4.18}$$

where $\sigma' = \max_i(-\frac{a_{i-1i}}{a_{ii}})$.

In both cases of defects relative to eigenvectors we have obtained that each element of d will be relatively smaller than e whenever μ (or λ) will be closer to 1. Notice that for negative μ we have d relatively larger than e than the positive case. Then one necessitates an *a priori* estimate of the eigenvalues' location. This problem has been investigated and for a beautiful review on the subject we refer to [56]. Here we report only a theorem which helps to locate the largest eigenvalue of M_J:

Theorem 5 *Let $\lambda_1 \geq \lambda_2 \geq, \ldots, \geq \lambda_n$ be the eigenvalues of a $n \times n$ real symmetric matrix A. Let us denote with $s(A) = \max_{j,k} |\lambda_j - \lambda_k|$ the spread of A. Then*

$$2 \max_{j<k} |a_{jk}| \leq s(A) \leq [2(1 - \frac{1}{n})(\sum_j a_{jj})^2 - 4 \sum_{j<k} (a_{jj}a_{kk} - a_{jk}a_{kj})]^{1/2} \quad . \tag{4.19}$$

Then we can prove a lemma which gives bounds for the largest eigenvalue μ_{max} of M_J (this bounds are obviously extended to the largest eigenvalue of M_{GS}).

Lemma 5 *Let $A \in \mathcal{A}$, M_J as above. Then*

$$\max_j \sqrt{\frac{a_{jj+1}a_{j+1j}}{a_{jj}a_{j+1j+1}}} \leq \mu_{max} < 1 \quad . \tag{4.20}$$

Proof. Because of theorem 4 $s(M_J) = 2\mu_{max}$. We showed that M_J is equivalent to a symmetric tridiagonal matrix B with zero diagonal elements and off-diagonal elements given by

$$b_{ii+1} = b_{i+1i} = \sqrt{\frac{a_{ii+1}a_{i+1i}}{a_{ii}a_{i+1i+1}}} \quad , \quad i = 1, \ldots, n-1 \quad . \tag{4.21}$$

Then, because of theorem 5, the left inequality (4.20) immediately holds for $\frac{s(B)}{2}$. But $2\mu_{max} = s(B)$. The right inequality is obvious, since the convergence of the Jacobi iteration.

The consequence of lemma 5 is that μ_{max} will be closer to 1 whenever A is a "weakly" diagonal dominant matrix.

4.2.2 Prolongation and Restriction

We have seen that iteration matrices act differently on different components of the error. We have showed that in case of damped Jacobi iteration, with suitable chosen damping parameter ω, and of Gauss-Seidel iteration, the high oscillating part of the solution error is reduced more quickly than the low oscillating part. Then, within a multilevel strategy, one tries to solve the remaining *smoother* part by means of a complementary solution process. This is done defining a smaller algebraic problem, then easier to solve, such that its solution approximates well that part of the error not efficiently reduced by the iteration itself. Then this solution can be used as a correction for the low frequency components of the error.

In 'geometric' multilevel methods the smaller algebraic problem is normally obtained discretizing the differential problem at hand using a mesh size which is twice as large as that used to define the original linear problem. For example, given a problem on $\Omega \subseteq \mathbb{R}^n$, $n = 2^\ell - 1$ the smaller equations will be defined on $\mathbb{R}^{n'}$, $n' = \frac{n-1}{2}$. Because we want to analyze the algebraic features of a multilevel solver being as close as possible to the standard "geometric" procedures, we restrict ourselves to consider algebraic problems with $n = n_\ell = 2^\ell - 1$. The positive integer ℓ is the level number. With this notation the equation for the error at level ℓ will be denoted by $A^\ell e^\ell = d^\ell$, and the relative smaller equation will be $A^{\ell-1}e^{\ell-1} = d^{\ell-1}$. Now the first problem is how to define the coarse error, $e^{\ell-1}$. We notice that in a geometric context the elements of this vector are a subset of the elements of e^ℓ. They must be selected so that one can recover, by means of a prolongation operator $P : \mathbb{R}^{n_{\ell-1}} \rightarrow \mathbb{R}^{n_\ell}$, the set of all fine elements. Hence, as a first step we have to define properly this operator.

Let us suppose to apply the damped Jacobi iteration with $\omega = \frac{1}{2}$. After few iterations the error consists mainly of combinations of eigenvectors of M_J corresponding to positive eigenvalues, and among them the principal components are those associated to eigenvalues relatively closer to 1. Then equations (4.9) can be used to deduce P. Actually, choosing

arbitrarily the subset of coarse variables to be $e^{\ell-1} = (e_2^\ell, e_4^\ell, \ldots, e_{n_\ell-1}^\ell)^T$, from (4.9) one deduces a $n_\ell \times n_{\ell-1}$ prolongation matrix (for simplicity of presentation we will write explicitly all operators for the case $n_\ell = 7$, but the discussion is valid for all n_ℓ as defined above)

$$P \sim \begin{bmatrix} -\frac{a_{12}}{a_{11}} & 0 & 0 \\ 1 & 0 & 0 \\ -\frac{a_{32}}{a_{33}} & -\frac{a_{34}}{a_{33}} & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{a_{54}}{a_{55}} & -\frac{a_{56}}{a_{55}} \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{a_{76}}{a_{77}} \end{bmatrix} . \tag{4.22}$$

The above heuristic discussion can be repeated in case of Gauss-Seidel iterations and leads to the same prolongation operator. For the moment we suppose that P as given by (4.22) is correct. That is $e^\ell \cong Pe^{\ell-1}$. Another way to represent the coarse space is to represent it as a subspace of Ω, that is $e^{\ell-1} = (0, e_2, 0, e_4, \ldots, e_{n-1}, 0)^T \in \Omega_c$. So $\Omega_c \subseteq \mathbb{R}^{n_\ell}$ contains all vectors v such that $v_{2p-1} = 0$, $p = 1, 2, \ldots, 2^{\ell-1}$. In this way the following remark is easily formulated

Remark 2 *The prolongation given by (4.22) can be written in a compact form in terms of the splitting of A^ℓ, that is*

$$P = \{I + D^{-1}(L+U)\}|_{\Omega_c} . \tag{4.23}$$

Further, to construct the coarse equation properly one needs a restriction operator R so that defining $A^{\ell-1} = RA^\ell P$ and $d^{\ell-1} = Rd^\ell$ the solution of $A^{\ell-1}e^{\ell-1} = d^{\ell-1}$ coincides with the fine solution in the sense defined above. If P is correct we should have $A^\ell Pe^{\ell-1} \cong d^\ell$ so let us look at the $n_\ell \times n_{\ell-1}$ matrix $A^\ell P$. It is such that all its elements of rows with odd index are zero [24]. Now suppose that the $n_{\ell-1} \times n_\ell$ restriction matrix is of the following type

$$R \sim \begin{bmatrix} r_{11} & 1 & r_{13} & 0 & 0 & 0 & 0 \\ 0 & 0 & r_{23} & 1 & r_{25} & 0 & 0 \\ 0 & 0 & 0 & 0 & r_{35} & 1 & r_{37} \end{bmatrix} . \tag{4.24}$$

Because of the zero rows the action of R on $A^\ell P$ is just to eliminate these zero (odd) rows, leaving unchanged those with even index and this result does not depend on the choice of the value of the entries r_{ij} (with odd j). Then, in principle, the simplest injection operator can be used as restriction. However defining $d^{\ell-1}$ this independence is lost.

Let us, for a moment, suppose to solve the linear system $A^\ell e^\ell = d^\ell$ by substitution as follows (later on we will refer to it as the substitution). Take the ith equation of the system with i even integer. Multiply the $(i-1)$-th and $(i+1)$-th equations by $-\frac{a_{ii-1}}{a_{i-1i-1}}$ and by $-\frac{a_{ii+1}}{a_{i+1i+1}}$ respectively. Sum the resulting equations to the ith one. You will recognize immediately that the matrix of coefficients (of the resulting system) for the variables $(e^\ell_2, e^\ell_4, \ldots, e^\ell_{n_\ell - 1})$, coincides with $RA^\ell P$. In addition we obtain that the $n_{\ell-1} \times n_\ell$ restriction matrix is (up to a multiplicative constant):

$$R = \begin{bmatrix} -\frac{a_{21}}{a_{11}} & 1 & -\frac{a_{23}}{a_{33}} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{a_{43}}{a_{33}} & 1 & -\frac{a_{45}}{a_{55}} & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{a_{65}}{a_{55}} & 1 & -\frac{a_{67}}{a_{77}} \end{bmatrix} . \tag{4.25}$$

We notice that the above restriction can be defined as the transpose of the prolongation matrix relative to $(A^\ell)^T$. Then in case of symmetric matrices we have simply that $R = P^T$ in accordance with [14, 71]. Furthermore, also the restriction operator derived above, can be written in terms of the splitting. But now we need also to define the projection operator on Ω_c, let us denote it with $P_{red} = \mathrm{diag}(0, 1, 0, 1, \ldots)$. Then we have

Remark 3 *The restriction given by (4.25) can be written in a compact form in terms of the splitting of A^ℓ, that is*

$$R = P_{red} + (L + U)D^{-1}P_{black} \ , \tag{4.26}$$

where $P_{black} = I - P_{red}$.

To complete this section it remains to analyze the matrix $A^{\ell-1}$. As we said it is given by $RA^{\ell}P$. It is easy to compute that $A^{\ell-1}$ is tridiagonal and the diagonal entries are given by

$$a_{II}^{\ell-1} = a_{ii}^{\ell} - \frac{a_{ii-1}^{\ell} a_{i-1i}^{\ell}}{a_{i-1i-1}^{\ell}} - \frac{a_{ii+1}^{\ell} a_{i+1i}^{\ell}}{a_{i+1i+1}^{\ell}} \quad , \quad I = 1, \ldots, n_{\ell-1} \quad , \quad i = 2I \quad , \tag{4.27}$$

and the off-diagonal elements are

$$a_{II-1}^{\ell-1} = -\frac{a_{ii-1}^{\ell} a_{i-1i-2}^{\ell}}{a_{i-1i-1}^{\ell}} \quad , \quad I = 2, \ldots, n_{\ell-1} \quad , \tag{4.28}$$

and

$$a_{II+1}^{\ell-1} = -\frac{a_{ii+1}^{\ell} a_{i+1i+2}^{\ell}}{a_{i+1i+1}^{\ell}} \quad , \quad I = 1, \ldots, n_{\ell-1} - 1 \quad , \tag{4.29}$$

where $i = 2I$. Because $A^{\ell} \in \mathcal{A}$, it is easy to prove that all diagonal entries are positive whereas the off-diagonal one are negative. Moreover the diagonal dominance of A^{ℓ} implies the same property for $A^{\ell-1}$. Then this matrix belongs to the model class $\mathcal{A}$ so, in particular, it is invertible and the coarse solution is $e^{\ell-1} = (A^{\ell-1})^{-1} d^{\ell-1}$. Finally, A^{ℓ}, R, and P, are now all expressed in terms of the splitting, and the same occurs for the matrix $A^{\ell-1}$. In fact we have the following

Remark 4 *The matrix of the coefficients of the coarse equation can be written in terms of the splitting as*

$$A^{\ell-1} = \{D - (L+U)D^{-1}(L+U)\}|_{\Omega_c} \quad . \tag{4.30}$$

Notice that at this stage all multilevel components are given in terms of the splitting except the choice of the coarse space Ω_c.

4.3 Algebraic Multilevel Algorithms

We have analyzed the tridiagonal case which, despite of its simplicity, exhibits most of the algebraic features of a ML approach. In this way we have given, for example, the

prolongation, restriction, and corresponding coarse problem in terms of the splitting of the matrix A. In this way a multilevel scheme can be defined by means of matrix theoretic methods [62], and this fact is not restricted to the tridiagonal case. In addition, thanks to this formulation, we find later that also the choice of the coarse space can be based also on the above splitting. That is a pure algebraic multilevel approach.

In the following section we continue the analysis of the tridiagonal case, with the formulation of the multilevel scheme. The purpose is to complete the study of the tridiagonal case, and also to provide an example of how the algebraic approach can be used to compute sharp estimates of the ML rate of convergence.

4.3.1 The Tridiagonal Case

A multilevel solution process requires a smoothing iteration, the prolongation of the coarse solution and the restriction of the fine defects. In the last section we have seen that the last two components we have found, are "correct" in the sense that they reproduce exactly the substitution procedure described above. However, the same procedure suggests that P given by (4.22), even if correct to define $A^{\ell-1}$, produces an error when used to prolongate the coarse variables $e^{\ell-1}$. Here we compute this error so that the twolevel solution process is exhibited.

Suppose, again, to apply to the system $A^\ell e^\ell = d^\ell$ the substitution. Then solve exactly (with any direct method) the coarse problem $A^{\ell-1}e^{\ell-1} = d^{\ell-1}$. In this way we obtain the subset of fine variables $(e^\ell_2, e^\ell_4, \ldots, e^\ell_{n_\ell-1})$. The remaining fine variables are then found as follows (let us omit the superscript ℓ):

$$\begin{cases} e_1 &= -\frac{a_{12}}{a_{11}}e_2 + \frac{d_1}{a_{11}} \ , \\ e_i &= -\frac{a_{ii-1}}{a_{ii}}e_{i-1} - \frac{a_{ii+1}}{a_{ii}}e_{i+1} + \frac{d_i}{a_{ii}} \ , \quad i = 3,5,\ldots,n_\ell - 1 \ , \\ e_{n_\ell} &= -\frac{a_{n_\ell n_\ell-1}}{a_{n_\ell n_\ell}}e_{n_\ell-1} + \frac{d_{n_\ell}}{a_{n_\ell n_\ell}} \ . \end{cases} \tag{4.31}$$

But notice that (4.31), together with the identities $e^\ell_i = e^{\ell-1}_I$ for $I = 1,\ldots,n_{\ell-1}$, $i = 2I$, can be written in a compact form using the splitting formulation

$$e^\ell = Pe^{\ell-1} + D^{-1}P_{black}d^\ell \ . \tag{4.32}$$

Hence we are able to formulate first the substitution twolevel algorithm (STL). Let u^ℓ be an approximation to $u = A^{-1}f$ (it can be $u^\ell = 0$), the corresponding solution error is $e^\ell = u^\ell - u$ and the relative defect is $d^\ell = Au^\ell - f$. Solve exactly the linear system of equations $RAPe^{\ell-1} = Rd^\ell$. Then by means of (4.32) return to the fine level obtaining e^ℓ. The solution is then given using e^ℓ as a correction, that is

$$u = u^\ell - (Pe^{\ell-1} + D^{-1}P_{black}d^\ell) \ . \tag{4.33}$$

The last step is the "exact" coarse level correction.

Now we compare this solution scheme with the standard twolevel method given in section 1.2.2, and reported below without post-smoothing

- Twolevel method for solving $A^\ell u^\ell = f^\ell$.

  1. Smoothing step: let $\tilde{u}^{(0)}$ be the result of ν "smoothing" iterations (so that the solution error is smooth enough);
  2. calculation of the defect: $d^\ell = A^\ell \tilde{u}^{(0)} - f^\ell$;
  3. restriction of the defect: $d^{\ell-1} = Rd^\ell$;
  4. solution of the coarse (defect) equations: $e^{\ell-1} = (A^{\ell-1})^{-1}d^{\ell-1}$;
  5. coarse level (CL) correction (of $\tilde{u}^{(0)}$): $\tilde{u}^{(1)} = \tilde{u}^{(0)} - Pe^{\ell-1}$.

There are only two differences between the STL and the TL method. The first is that the latter requires a smoothing step whereas the former does not. The second difference is in the CL correction: that of the twolevel method approximates that of the STL method whenever the defect is approximately zero. But these differences are very easy to explain.

In section 4.2.1 we showed that the (smoothing) damped Jacobi iteration reduces efficiently that part of the solution error which can be expressed as a combination of eigenvectors of M_J corresponding to the negative eigenvalues. The remaining part of the error is mainly that corresponding to the larger positive eigenvalues. Then, as we have proved in section 4.2.1, after the application of a smoothing iteration the remaining error

is that for which the defect is relatively smaller. Hence, the smoothing iteration in the twolevel method is necessary in order to have a good coarse level correction.

The most favourable situation occurs whenever the larger eigenvalues are relatively closer to 1. But this corresponds to an inefficient iterative solution process. The two facts are indeed correlated and this is why the coarsening procedure is complementary to the iterative procedure. A more delicate situation occurs with the Gauss-Seidel iteration because of the principal vectors. For simplicity we do not consider this problem in detail, and we assume that the initial solution error is a combination of eigenvectors of M_{GS} corresponding to its nonzero eigenvalues.

The coarsening procedure of the twolevel method can be written in a condensed form as

$$\tilde{u}^{(1)} = \tilde{u}^{(0)} - P(A^{\ell-1})^{-1}R(A^{\ell}\tilde{u}^{(0)} - f^{\ell}) \ . \tag{4.34}$$

Then also the twolevel solution process can be interpreted as an iteration whose matrix is given by

$$M_{TL} = (I - P(A^{\ell-1})^{-1}RA^{\ell})M^{\nu} \ , \tag{4.35}$$

where the smoothing iteration matrix M is applied ν times.

Now let us consider the case when $M = M_J(\omega)$, and analyze M_{TL} in detail. Let $u^{(0)}$ be the initial approximation to the solution of the linear system $A^{\ell}u^{\ell} = f^{\ell}$. We can express the corresponding solution error $e^{(0)}$ as a combination of eigenvectors of M_J, $e^{(0)} = \sum_k e_k v^k$. After ν times the damped Jacobi iteration one obtains a new approximation $\tilde{u}^{(0)}$ whose relative error is given by $\tilde{e}^{(0)} = \sum_k e_k \tilde{\mu}_k^{\nu} v^k$, where $\tilde{\mu}_k = (1 - \omega + \omega\mu_k)$. Moreover the relative defect is given by $d = D\sum_k e_k \tilde{\mu}_k^{\nu}(1 - \mu_k)v^k$. Then we apply the coarsening procedure and obtain $\tilde{u}^{(1)}$. Thanks to (4.32) we have immediately the solution error $\tilde{e}^{(1)}$:

$$\tilde{e}_i^{(1)} = 0 \ , \quad i = 2, 4, \ldots, n_{\ell} - 1 \ , \tag{4.36}$$

$$\tilde{e}_i^{(1)} = -(D^{-1}P_{black}d)_i = -\sum_k e_k \tilde{\mu}_k^{\nu}(1 - \mu_k)v_i^k \ , \quad i = 1, 3, \ldots, n_{\ell} \ . \tag{4.37}$$

This result allows us to obtain two inequalities which will be used to estimate a bound to the spectral radius of M_{TL}. Let us fix the parameters ω and ν. Denote with

$\gamma = \max_k(|\tilde{\mu}_k^\nu(1-\mu_k)|)$, then we obtain

$$\|\tilde{e}^{(1)}\|_\infty \leq \gamma \|e^{(0)}\|_\infty \ . \tag{4.38}$$

It is clear that larger ν will give smaller γ. However for moderate ν, one can optimize (minimize) the above bound along ω. We find that choosing $\omega = \frac{1}{1+\mu_{max}}$, γ is bounded by

$$\gamma_b = \frac{\nu^\nu}{(\nu+1)^{(\nu+1)}}(1+\mu_{max}) \ . \tag{4.39}$$

The above discussion can be repeated in case of Gauss-Seidel smoothing iteration. The resulting bound is given by

$$\gamma_b = \sigma \ , \quad \nu = 1 \ , \tag{4.40}$$

$$\gamma_b = \frac{(\nu-1)^{(\nu-1)}}{\nu^\nu}\sigma \ , \quad \nu \geq 2 \ . \tag{4.41}$$

Notice that both (4.39) and (4.41) behave asymptotically as $\gamma_b \simeq c/\nu$, in agreement with the prediction of theorem 5 given in section 1.2.2. Observe that the computed γ_b (n independent) are always smaller than one. Then the TL iteration converges to the solution for any initial starting approximation.

The value of γ_b provides a sharp bound to the rate of convergence of the twolevel iteration as it results by applying the following theorem [76]

Theorem 6 *Consider a linear iterative method $u^{(\nu+1)} = Mu^{(\nu)} + Nf$ to solve $Au = f$. Then the errors $e^{(\nu)} = u^{(\nu)} - u$ satisfy*

$$sup_{e^{(0)}\neq 0} \limsup_{\nu\to\infty} \sqrt[\nu]{\frac{\|e^{(\nu)}\|}{\|e^{(0)}\|}} = \rho(M) \ . \tag{4.42}$$

Here $\|\cdot\|$ is an arbitrary norm.

Using this theorem and the above results we are able to estimate a bound on the rate of convergence of M_{TL}. For if $e^{(0)} \neq 0$ we can write in terms of the maximum norm the following

$$\sqrt[\nu]{\frac{\|e^{(\nu)}\|_\infty}{\|e^{(0)}\|_\infty}} \leq \sqrt[\nu]{\frac{\gamma_b^\nu \|e^{(0)}\|_\infty}{\|e^{(0)}\|_\infty}} = \gamma_b \tag{4.43}$$

Then from (4.42) using (4.43) we have

$$\rho(M_{TL}) \leq \gamma_b \ . \tag{4.44}$$

There is, however, a special iteration for which the TL and the STL methods coincide. That is, based on the split of the set of fine variables into a 'red' and a 'black' subset. In our case the former is the set of e_i^ℓ with i even, the latter contains the remaining ones with i odd, and both are ordered lexicographically. Therefore the Gauss-Seidel iteration is applied first to the red subset and subsequently to the black one. This iteration is called the red-black (RB) Gauss-Seidel iteration [85]. It is clear that after this iteration one obtains $d_i^\ell = 0$, $i = 1, 3, \ldots, n_\ell$. Hence the prolongation (4.31) is equivalent to (4.22) and the TL algorithm employing only one RB GS iteration solves exactly the algebraic problem.

Let us remark that in this section the entire twolevel process is described in terms of the splitting matrices D, L and U. Further we write also the error after the coarse level correction in terms of these matrices. Therefore no special use has been done of the fact that the matrix is tridiagonal except for the spectral properties of the corresponding Jacobi iteration matrix. Actually the symmetry of the spectrum $\sigma(M_J)$ will result sufficient in order to extend the results given above to larger classes of algebraic problems. For this extension it remains to explain how to select the coarse space automatically starting from the algebraic features of the problem. We solve this last problem in the following section in case of some concrete examples. The general answer to this question will be given in the following chapter.

4.3.2 Extension to More General Cases

An important feature of the tridiagonal case is that after coarsening the resulting problem remains the same, i.e. tridiagonal. This implies that the twolevel procedure can be repeated on the coarser level, and so on recursively. Another feature of this case is that at each level one can repeat the choice of a coarser space using red-black ordering. Actually the two things are intimately correlated, and it turns out that the splitting formulation explains this connection and provides also the choice of the coarse space automatically.

Now we prove this connection, first for the tridiagonal case and then for more general cases, through the symmetry of the spectrum of the Jacobi iteration matrix. Later we discuss the relationship between our splitting formulation of the coarse problem and that obtained by the standard coarsening procedure.

In order to present our results we reconsider the tridiagonal case. Let us add some more remarks on the eigenproblem $M_J v^k = \mu_k v^k$, written explicitly in (4.9). This has a symmetric spectrum, that is if $\mu_k \in \sigma(M_J)$ then $-\mu_k \in \sigma(M_J)$ (with the same multiplicity). We denote with v^k and $\tilde{v}^k$, the corresponding eigenvectors. Then it is easy to verify by simple changes of sign in (4.9) that $v_i^k = (-1)^i \tilde{v}_i^k$. Therefore defining the operator $\mathcal{B}$ as

$$\mathcal{B} = \text{diag}(-1, 1, -1, 1, \ldots) \ , \tag{4.45}$$

we have $v^k = \mathcal{B}\tilde{v}^k$. Clearly $\mathcal{B}^2 = I$ and the following relations hold

$$\mathcal{B} = P_{red} - P_{black} \ , \tag{4.46}$$

and

$$P_{red} = \frac{I + \mathcal{B}}{2} \ . \tag{4.47}$$

In particular it is immediate to verify that $\mathcal{B}$ maps the null space of M_J into itself.

It is clear that the "a priori" choice of the coarser space Ω_c made in section 4.2.2, is also dictated by the algebraic features of the problem. Therefore the entire formulation of the twolevel procedure can be done in terms of the splitting, the coarse space being given by $\Omega_c = \frac{I+\mathcal{B}}{2}\Omega$.

Notice that the operator (4.45) has the two following properties which will prove useful in the following [3]

$$\mathcal{B}D = D\mathcal{B} \quad \text{and} \quad \mathcal{B}(L+U) + (L+U)\mathcal{B} = 0 \ . \tag{4.48}$$

Now we are able to extend the above formalism to more general situations. For example let us consider the Poisson equation with Dirichlet boundary conditions in two

[3] In the following chapter these properties will be used to construct $\mathcal{B}$.

dimensions

$$\begin{cases} -(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}) = f(x,y) \ , & \text{in} \quad \tilde{\Omega} = (0,1) \times (0,1) \\ u(x,y) = 0 \ , & \text{for} \quad x \in \Gamma = \partial\tilde{\Omega} \ . \end{cases} \tag{4.49}$$

Then, let $h = \frac{1}{n+1}$, and introduce the grid of points $\tilde{\Omega}_h = \{(ih, jh), \ i,j = 0,1,\ldots,n+1\}$. A well known difference scheme for the discretization of (4.49) is the 5-point formula

$$-[u_{ij+1} + u_{ij-1} + u_{i+1j} + u_{i-1j} - 4u_{ij}] = h^2 f_{ij} \ . \tag{4.50}$$

Here the value of the discrete solution at the point (ih, jh) is represented by u_{ij}. The entire solution is stored in a vector where the unknowns are located in a *natural order*

$$u = (u_{11} \ldots u_{n1}; u_{12} \ldots u_{n2}; \ldots; u_{1n} \ldots u_{nn}) \in \Omega \ . \tag{4.51}$$

In order to obtain an operator which acts as $\mathcal{B}$ above, we consider the eigenproblem $M_J v^{k,m} = \mu_{k,m} v^{k,m}$, where M_J is the convergent [84] Jacobi iteration matrix associated to (4.50). Namely

$$\frac{1}{4}[v^{k,m}_{ij+1} + v^{k,m}_{ij-1} + v^{k,m}_{i+1j} + v^{k,m}_{i-1j}] = \mu_{k,m} v^{k,m}_{ij} \ . \tag{4.52}$$

Because the matrix of coefficients of (4.50) is consistently ordered [85], M_J has a symmetric spectrum. Notice that in (4.52) the variables which appear in the two sides of the equation are always defined on two distinct sets: those with $i+j$ even and $i+j$ odd. As before we call these two subsets of fine variables red and black, respectively. Since the spectrum is symmetric we change $\mu_{k,m} \to -\mu_{k,m}$ and denote with $\tilde{v}^{k,m}$ the eigenvector corresponding to $-\mu_{k,m}$. Then (4.52) is satisfied with $\tilde{v}^{k,m}_{ij} = (-1)^{i+j} v^{k,m}_{ij}$. This means that we can define the operator[4]

$$\mathcal{B} = \text{diag}(1, -1, 1, \ldots; -1, 1, -1, \ldots; \ldots) \ , \tag{4.53}$$

such that $\tilde{v}^{k,m} = \mathcal{B} v^{k,m}$.

Then it is easy to check that (4.46), (4.47) and (4.48) hold with $\mathcal{B}$ given by (4.53). The resulting coarse space is given by $\Omega_c = P_{red}\Omega$. That is the well known coarsening in the red points when a red-black ordering of the meshes is used.

[4]Also in this case $\mathcal{B}$ maps the null space of M_J into itself.

Now we have to verify that the prolongation, restriction and the coarse matrix, given above using the splitting formalism, are properly defined also in this case. Actually we notice that the matrices $(L+U)$, $D^{-1}(L+U)$ and $(L+U)D^{-1}$ have a common property: they map each element of $P_{red}\Omega$ into $P_{black}\Omega$, and vice versa. This is an immediate consequence of (4.47) and (4.48). Therefore the operators P, R, $A^{\ell-1}$, given by (4.23), (4.26) and (4.30) map the right spaces. Moreover, the coarse matrix results to be invertible, as is proved by

Lemma 6 *Since M_J is a convergent iteration matrix then $(A^{\ell-1})^{-1}$ exists.*

Proof. For the proof notice that (4.30) can be written in the form

$$A^{\ell-1} = D(I - M_J^2)|_{\Omega_c} \ .$$

Convergence implies that the spectral radius $\rho(M_J) < 1$, hence $(I - M_J^2)$ is invertible, and the lemma follows.

Finally we need to discuss the coarse level correction step. As for the tridiagonal case one uses the correction given through $e^{\ell} = Pe^{\ell-1} + D^{-1}P_{black}d^{\ell}$. In this way no iteration sweeps are needed. However the standard prolongated correction, i.e. $e^{\ell} = Pe^{\ell-1}$, can be used contemporary with a suitable iterative method. So, for example, a RB GS iteration can be used and the resulting coarse level correction will give the exact solution in the fine level. However the damped Jacobi scheme can also be applied and the analysis given in the previous section can be repeated straightforwardly, giving equivalent convergence results. More precisely we get the same functional dependence of the rate of convergence of the resulting twolevel cycle with respect to the number of smoothing sweeps ν.

Let us summarize the above results. We have considered the algebraic problem which arises from the discretization of a Dirichlet boundary value problem in one or two spatial dimensions. In the second case we have studied explicitly only the case given by (4.50). But the corresponding results apply without modifications to any algebraic problem which arises from the 5-point discretization of elliptic problems in two dimensions (for example linear convection-diffusion problems). We have seen that all the components

of the twolevel iteration are given in terms of the splitting of the matrix of coefficients. To this splitting, via the symmetry of the spectrum of the resulting Jacobi iteration, we also have related the choice of the coarse space, which becomes a feature of the algebraic problem. Therefore the TL procedure results to be completely defined in terms of the entries of the linear system to be solved.

Regarding algebraic systems which derive from elliptic problems one should notice that the above results extend also to three dimensional Dirichlet boundary value problems. In this case each grid point (ih, jh, kh) will be red or black if $i+j+k$ is an even or odd integer, respectively. Hence in correspondence we define $\mathcal{B}$ which now acts on the element u_{ijk} multiplying it by $(-1)^{i+j+k}$, and the discussion above can be repeated also for the 7-point formula which discretizes the three dimensional Laplacian.

4.3.3 Some Approximations and Remarks

With the transfer operators given in terms of the splitting, and the choice of the corresponding coarse space, we obtain a well defined coarse problem as results by lemma 6. However, except for the tridiagonal case, this approach leads to coarse matrices which correspond to a stencil different from that of the fine difference operator. For example, starting from (4.50), that is a 5-point operator, we obtain a 9-point one. And its stencil (acting on the red points) is

$$\begin{array}{ccccc} \cdot & \cdot & -\frac{1}{4} & \cdot & \cdot \\ \cdot & -\frac{1}{2} & \cdot & -\frac{1}{2} & \cdot \\ -\frac{1}{4} & \cdot & 3 & \cdot & -\frac{1}{4} \\ \cdot & -\frac{1}{2} & \cdot & -\frac{1}{2} & \cdot \\ \cdot & \cdot & -\frac{1}{4} & \cdot & \cdot \end{array} \tag{4.54}$$

For this reason it is impossible to apply the above procedures recursively. But as we now show the exact twolevel method just described and the one which results by the standard multilevel formulation are comparable. However the discussion which follows is in part heuristic and deserves further analysis. Nevertheless some numerical experiments confirm our estimates of the convergence rate of a multilevel iteration.

Let us first repeat the parallelism of the TL coarsening with the substitution procedure. That is, take (4.50) on a red point, then express the "black" variables which appear there and eliminate them using the same equation corresponding to these black points. Hence (4.54) is obtained [5]. From this procedure or directly from (4.26), we have the following stencil for the restriction operator

$$\begin{array}{ccccc} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \frac{1}{4} & \cdot & \cdot \\ \cdot & \frac{1}{4} & 1 & \frac{1}{4} & \cdot \\ \cdot & \cdot & \frac{1}{4} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{array} \tag{4.55}$$

The resulting (coarse) algebraic problem, at a given (red) point, reads

$$\begin{aligned} -\tfrac{1}{4}[u_{i-2j} + u_{i+2j} + u_{ij-2} + u_{ij+2}] - \tfrac{1}{2}[u_{i+1j+1} + u_{i+1j-1} + u_{i-1j+1} + u_{i-1j-1}] + 3u_{ij} \\ = h^2 f_{ij} + \tfrac{h^2}{4}[f_{ij+1} + f_{ij-1} + f_{i+1j} + f_{i-1j}] \ . \end{aligned} \tag{4.56}$$

Now assume that the solution u is sufficiently smooth. We can try to modify the stencil (4.54) so that the resulting equations provide the fine level solution with possible $O(h^2)$ errors. This is possible if we notice that (4.49) can be discretized also by another (rotated) 5-point formula [29]

$$-\frac{1}{2}[u_{i+1j+1} + u_{i+1j-1} + u_{i-1j-1} + u_{i-1j+1} - 4u_{ij}] = h^2 f_{ij} \ , \tag{4.57}$$

with $O(h^2)$ truncation error.

Further take the subset of red points with both i and j even indices. We call them the red-red points. These are exactly the points of a coarse grid obtained doubling the mesh size, $h \to H = 2h$ (standard coarsening). Then take (4.56) centered in a point of this subset, and use (4.57) to eliminate those variables which are not defined on the red-red subset. Hence we obtain the coarse equation

[5] This is an obvious result since with the splitting formulation we have reduced the problem of the fine level to one defined only on the red points.

$$-[u_{i-2j}+u_{i+2j}+u_{ij-2}+u_{ij+2}-4u_{ij}]=\frac{H^2}{4}[f_{ij+1}+f_{ij-1}+f_{i+1j}+f_{i-1j}] \quad . \qquad (4.58)$$

Clearly the above substitution requires a smooth function u [6], but this would restrict the range of applications of the ML procedure. For this reason, in order to overcome this limitation one considers the defect equation (instead of the solution equation) after the application of a few sweeps of a relaxation scheme. Therefore we replace in (4.58) the solution u with the corresponding error e, and the defect on the right hand side.

In this way we should remain close to the convergence results obtained in the tridiagonal case. This is confirmed by numerical experiments. To this purpose let us describe the other multilevel component not yet discussed, the prolongation operator. From (4.58) we obtain the red-red variables only. The remaining red variables are obtained by the well known linear interpolation

$$e_{ij}=\frac{1}{4}(e_{i+1j+1}+e_{i+1j-1}+e_{i-1j-1}+e_{i-1j+1})^{red-red} \quad , \qquad (4.59)$$

this operator, however, can be derived from the rotated equation (4.57) as discussed in [37].

Finally, it remains to compute the black variables. A black point lies in the middle of two red-red points and the following interpolation can be used [29]

$$e_{ij+1}^{black}=\frac{(e_{ij}+e_{ij+2})^{red-red}}{2} \quad . \qquad (4.60)$$

With the operators described above a multilevel scheme is completely defined. As an iterative scheme we use the pointwise Gauss-Seidel method. Therefore we expect a convergence factor which behaves like

$$(\nu-1)^{(\nu-1)}/\nu^{\nu} \quad ,$$

up to a constant which now remains undetermined because of the approximations used above. However this constant is expected to be of the order of $\|D^{-1}U\|_{\infty}=1/2$ as it would be if no approximations are introduced.

[6] In this case the truncation errors are typically of the form $O(h^2)$ = *const* $\times$ h^2[fourth order derivatives of u] [29].

ν_1	Obs. $\tilde{\rho}_{2D}$	Obs. $\tilde{\rho}_{3D}$	$\frac{(\nu-1)^{(\nu-1)}}{\nu^{\nu}}$
2	0.194	0.264	0.250
3	0.144	0.157	0.148
4	0.105	0.111	0.105
5	0.083	0.088	0.081
6	0.069	0.074	0.066
7	0.059	0.065	0.056
8	0.049	0.055	0.049

Table 4.1: Results of numerical experiments. In the first column we report the number of (pre-) smoothing Gauss-Seidel sweeps at each level. In the last column we write the values of the convergence rates predicted by the splitting formulation. These are compared with the observed reduction factors of the 2D- and 3D cases.

In order to verify this qualitative prediction we use a multilevel cycle with the above components. We have solved a Dirichlet Poisson problem with $f(x,y) = 2\sin(x+y)$, and boundary conditions given by $u(x,y) = \sin(x+y)$, in a $(0,2)\times(0,2)$ square. The coarsest (uniform) grid has $h_1 = 1$ and the number of levels involved was $M = 6$. The reduction factors reported in table 4.1 resulted from a number (> 10) of ML V-cycles without post-smoothing. Since we claimed that the theoretical results of the previous section are valid also in case of a 3D Poisson problem discretized by a 7-point formula, we have constructed a multilevel algorithm which solves this problem. This code uses a pointwise Gauss-Seidel relaxation and the transfer operators employed are obtained in the same way followed in the 2D case. We use this algorithm to solve a 3D Poisson problem with $f(x,y,z) = 3\sin(x+y+z)$, and boundary conditions given by $u(x,y,z) = \sin(x+y+z)$, in a $(0,2)\times(0,2)\times(0,2)$ cube. The coarsest mesh size was $h_1 = 1$ and the number of levels involved was $M = 4$. Using the same V-cycle described above we obtain the reduction factors reported in table 4.1.

4.3.4 Conclusion

A great merit of multilevel methods is to have stimulated a careful analysis of classical iteration schemes. In fact in the past the objective of Numerical Analysts was to develop iterative schemes with best convergence rates possible. But for multilevel purposes the smoothing property of these iterations is important. In this respect we have completed the analysis of the Gauss-Seidel method showing that the formalism of principal vectors must be introduced and proving the good damping of this scheme with respect to these components. Further we have shown how a careful study of the matrix iterative eigenproblem leads naturally to the construction of the transfer operators necessary for the ML approach. In this way we were able to give an algebraic interpretation of these components proving that the classical multilevel method can be derived (at least approximately) from the splitting approach used to define the classical iterative procedures. This is a new result which allows precise convergence estimates of standard multilevel methods applied to linear Dirichlet boundary value problems in 1, 2 and 3 spatial dimensions.

Chapter 5

A Twolevel Method in Hilbert Space

5.1 Introduction

At the very beginning of this thesis we mentioned that the analysis of existing numerical schemes may lead to the formulation of new analytical methods. That is to obtain techniques which apply at the same time to finite systems of algebraic equations or equally to differential, integral, and other type of equations. Therefore it is natural to ask whether or not a multilevel scheme can be formulated to handle continuous problems. However, apart from this abstract motivation, this question can be raised by several reasons. The first one originates by the simple observation that ML methods provide good convergence behaviour in many cases where classical iteration schemes do not converge at all. Therefore a multilevel method for continuous problems could provide a contractive iteration procedure when other methods are not available. A second reason arises observing how simple is the ML approach to a non linear problem, which could result useful if one succeed to extend this method also in that case. Finally, among many other motivations, a general formulation of ML methods for both continuous and discrete equations will result obviously useful to understand more deeply the multilevel idea.

However this ambitious goal requires time and a systematic approach. It is therefore useful to generalize the multilevel method by starting from simple cases and trying to

distinguish the general principles from the special techniques designed to solve specific problems. The first step is to extend the simplest ML scheme studied, that is the twolevel method for linear problems reported in the first chapter of this thesis. But also the choice of the problem to be considered should remain as close as possible to that which normally arises from the discretization of a simple elliptic equation. In fact we will consider the class of bounded, positive definite, self-adjoint continuous operators.

In this way we are able to give a first extension of multilevel methods for operator equations in infinite dimensional Hilbert spaces [8]. This objective is achieved by analyzing our results [7], reported and extended in chapter 4. There we prove how a twolevel method can be constructed using the splitting formalism. Further we have the fact that the same formalism was used by W.V. Petryshyn [65, 66], in order to define new classes of iterative methods in (generally complex) Hilbert spaces. Actually he extended many of the classes of iterative methods and, in particular, the Jacobi and Gauss-Seidel iterative procedures. For this reason the following results could be considered in the line of those of Petryshyn, as the TL method represents a development of classical iterative methods.

The essential point of our extension is then represented by the suitable definition of "levels" in an infinite dimensional space. Then we show that the choice of the coarse space, based in the standard ML approach on the concept smoothness, can be recovered within the splitting formulation based on the spectral properties of the Jacobi iteration matrix. So it is this general property which we will now use to define the coarse Hilbert space.

Having defined this suitable auxiliary space, we need to construct, in correspondence, an operator equation whose solution represents a part of the whole solution of the original problem. Therefore, first we need a suitable restriction procedure which allows to the construct the right hand side of the coarse problem. And second we have to find a prolongation operator which guarantees the connection of the coarse with the fine solution. That is an operator which is used contemporarily with the restriction to define the coarse operator equation, and then is used to map the corresponding solution on the entire space. Clearly also for this purpose we have the help of our results obtained in the algebraic case. In fact in that case the restriction and prolongation operator result

automatically defined as soon as the splitting is given. And then the coarse operator equation is obtained following the Galerkin formulation used above.

First of all, in the following section we review the results of Petryshyn for the extension of some classical iterative methods in Hilbert space. In this way a suitable splitting of a bounded, positive definite, self-adjoint linear operator will be introduced. Thereby, since this formalism is available, we can use it to generalize the results obtained in the case of algebraic linear systems of equations. So, in section 5.2 we show how to select two subspaces which can be used to introduce the concept of levels in Hilbert space. This result also serves to generalize the discussion reported in the previous chapter to define a suitable problem dependent coarsening procedure. Then we prove in section 5.2.1 that the two transfer operators, given in terms of the splitting in that chapter, are right for their purpose also in the continuous case. Moreover we find that the resulting coarse operator equation is well defined and the twolevel procedure is obtained.

5.1.1 Iterative Methods in Hilbert Space

In this section we report the analysis of a class of iterative methods in an infinite dimensional Hilbert space $\mathcal{H}$. This class contains, in particular, the two classical Jacobi and Gauss-Seidel schemes. However the principal purpose here is to define our model problem and to introduce the concept of splitting for this case.

So let us denote with $A : \mathcal{H} \to \mathcal{H}$ a bounded, continuously invertible, self-adjoint linear operator. Moreover assume it is of the splitted form $A = D - L - U$ with bounded operators D, L, and U subject to the conditions

(a) D is self-adjoint;

(b) The operators $G_\omega = \frac{2-\omega}{\omega} D - L^* + U$, are positive definite, i.e., there exists $\beta = \beta(\omega) > 0$ such that

$$(G_\omega u, u) \geq \beta \|u\|^2, \quad \text{for } \omega \in \Omega \ ,$$

where L^* be the adjoint of L and Ω a set of reals $\omega > 0$;

(c) $(D - \omega L)$ has a bounded inverse defined on all of $\mathcal{H}$ for $\omega \in \Omega$.

Under these conditions and the positivity of the operator A it is possible to define an iterative method that determines an approximate solution u_n of the equation

$$Au = f \ , \quad f \in \mathcal{H} \ , \tag{5.1}$$

by the iteration

$$u_n = (D - \omega L)^{-1}\{(1-\omega)D + \omega U\}u_{n-1} + \omega(D - \omega L)^{-1}f \ , \tag{5.2}$$

where u_0 is an arbitrary initial approximation. We denote the *iteration operator* with $T = (D - \omega L)^{-1}\{(1-\omega)D + \omega U\}$.

In fact the following theorem [65, 66] has been proved, which states a necessary and sufficient condition so that (5.2) defines a sequence $\{u_n\}$ which converges to the unique solution of (5.1).

Theorem 7 *If D, L, U and Ω satisfy the conditions (a), (b) and (c), then the iterative method (5.2) converges to the unique solution of Eq. (5.1), for every f in $\mathcal{H}$ and any u_0 in $\mathcal{H}$, if and only if $A = D - L - U$ is positive definite.*

Furthermore, with these conditions one is able to estimate a bound for the spectral radius of the iteration operator. It is given by the following theorem [66]

Theorem 8 *Let D, L, U and Ω satisfy the conditions (a), (b) and (c). Suppose also that $A = D - L - U$ is positive definite. Then the spectral radius $r(T)$ of the operator T satisfies the inequality*

$$r(T) \leq 1 - \frac{\gamma_\omega}{\|D - \omega L\|} \ , \quad \omega \in \Omega \ ,$$

where

$$\gamma_\omega = \inf_{\|u\|=1}\{|((D - \omega L)u, u)| - |(P(\omega)u, u)|\} > 0 \ , \ \forall \omega \in \Omega \ ,$$

with $P(\omega) = (1-\omega)D + \omega U$.

Notice that depending on the choice of D, L, U and ω, many different special cases are obtained. For example, if $\omega = 1$, then (5.2) is the generalization of the Gauss-Seidel iteration, and when $L = 0$ one gets the Jacobi method.

However, there are many other iterative methods for the solution of a linear equation in Hilbert space widely used in numerical computations. For a beautiful survey of the subject we refer to [64].

5.2 Two Levels in Hilbert Space

In the previous chapter on algebraic multilevel methods, we have analyzed three classical linear systems of equations. That is those which arise from a specific finite difference discretization of Dirichlet boundary value problems in one, two and three spatial dimensions. Then we have used a common feature of these problems, that is the fact that the corresponding Jacobi iteration matrices have a symmetric spectrum (and each two eigenvalues λ and $-\lambda$ have the same multiplicity). Therefore it was easy to prove that there exist a (diagonal) operator $\mathcal{B}$ which maps each eigenvector of an eigenvalue λ to an eigenvector of $-\lambda$. Having $\mathcal{B}$, we then constructed a projection operator whose range results to be a suitable subspace to represent the coarse space in a multilevel approach.

In particular, denoting with Ω_+ and Ω_-, the range of this projection and its orthogonal complement, respectively, we have noticed that $D^{-1}(L+U)$ maps each element of Ω_+ into Ω_- and vice versa (to simplify the discussion which follows, we assume that the null space of M_J consists of the zero vector only). This last observation will be now used to extend the results obtained in the algebraic case.

So let us consider the infinite dimensional case. We assume from now on that D and $(L+U)$ have a bounded inverse. Following the above examples we consider the existence of two subspaces $\mathcal{H}_\pm$ of $\mathcal{H}$ such that

$$\mathcal{H} = \mathcal{H}_+ \oplus \mathcal{H}_- \ , \tag{5.3}$$

and

$$(L+U)\mathcal{H}_\pm = \mathcal{H}_\mp \ , \qquad D^{-1}(L+U)\mathcal{H}_\pm = \mathcal{H}_\mp \ . \tag{5.4}$$

We will refer to (5.3) and (5.4) as the *anti-reduction* properties.

Hence, in the coming part of this section we investigate some additional conditions on the operators D and $(L+U)$ so that (5.3) and (5.4) hold. First let us prove the following theorem

Theorem 9 *Let $B : \mathcal{H} \to \mathcal{H}$ be a bounded self-adjoint linear operator with bounded inverse such that*

$$B(L+U) + (L+U)B = 0 \ , \quad \textit{and} \quad BD = DB \ . \tag{5.5}$$

Then there exist subspaces $\mathcal{H}_+$ and $\mathcal{H}_-$ of $\mathcal{H}$, which satisfy (5.3) and $(L+U)\mathcal{H}_\pm = \mathcal{H}_\mp$, and $D^{-1}(L+U)\mathcal{H}_\pm = \mathcal{H}_\mp$.

Proof. Let us define $\mathcal{B} = (\sqrt{B^2})^{-1}B$, and denote $\tilde{T}_1 = (L+U)$ and $\tilde{T}_2 = D^{-1}(L+U)$. First of all we have $BD^{-1} = D^{-1}B$ and $B^2\tilde{T}_i = -B\tilde{T}_iB = \tilde{T}_iB^2$, $i = 1,2$. In particular the square root, as the inverse of the square root, of B^2, commutes with every bounded linear operator which commutes with B^2. This implies that $\mathcal{B}$ has the following properties

$$\mathcal{B}\tilde{T}_i + \tilde{T}_i\mathcal{B} = 0 \ , \ i = 1,2 \ \text{, and} \quad \mathcal{B}^2 = I \ , \quad \sigma(\mathcal{B}) = \pm 1 \ .$$

By means of $\mathcal{B}$, we can construct two projection operators denoted by $P_+ = \frac{1+\mathcal{B}}{2}$, $P_- = I - P_+$, with which we define $\mathcal{H}_\pm = P_\pm\mathcal{H}$. Clearly we have $\mathcal{H} = \mathcal{H}_+ \oplus \mathcal{H}_-$. Moreover, we obtain

$$\tilde{T}\mathcal{H}_\pm = \tilde{T}P_\pm\mathcal{H} = \tilde{T}\frac{1\pm\mathcal{B}}{2}\mathcal{H} = \frac{1\mp\mathcal{B}}{2}\tilde{T}\mathcal{H} = \mathcal{H}_\mp \ .$$

This completes the proof.

As discussed in the beginning of this section the existence of the operator $\mathcal{B}$ should be related with the spectrum of $(L+U)$. In fact, we prove now a theorem which is useful in order to define the splitting $A = D - L - U$.

Theorem 10 *Let $(L+U) : \mathcal{H} \to \mathcal{H}$ be a bounded, continuously invertible, self-adjoint linear operator. Then there exists a bounded linear operator $\mathcal{B}$, with bounded inverse, which anti-commutes with $(L+U)$ if and only if the spectrum $\sigma(L+U)$ is symmetric and each two eigenvalues λ and $-\lambda$ have the same multiplicity.*

Proof.

Necessity. Assume there exists an operator $\mathcal{B}$ so that $\mathcal{B}(L+U) + (L+U)\mathcal{B} = 0$. We have that $\lambda \in \sigma(L+U)$ if and only if there exists a sequence of elements $\psi_n \in \mathcal{H}$, $\|\psi_n\| = 1$, so that $\|((L+U) - \lambda)\psi_n\| \to 0$. Now consider the sequence $\phi_n = \mathcal{B}\psi_n/\|\mathcal{B}\psi_n\|$ in correspondence we have $((L+U) + \lambda)\phi_n = -\mathcal{B}((L+U) - \lambda)\psi_n/\|\mathcal{B}\psi_n\|$. That is, in terms of norms

$$\|((L+U)+\lambda)\phi_n\| = \|\mathcal{B}((L+U)-\lambda)\psi_n\|/\|\mathcal{B}\psi\| \leq \frac{\|\mathcal{B}\|}{\|\mathcal{B}\psi_n\|}\|((L+U)-\lambda)\psi_n\| \ .$$

This implies that $\|((L+U)+\lambda)\phi_n\| \to 0$, whenever $\|((L+U)-\lambda)\psi_n\| \to 0$, and $-\lambda \in \sigma(L+U)$ whenever $\lambda \in \sigma(L+U)$. Then $\sigma(L+U)$ is symmetric.

Sufficiency. Suppose $(L+U)$ has a symmetric spectrum as specified above. Then its spectral representation becomes

$$(L+U) = \int_0^{\|(L+U)\|} \lambda(dE_{+\lambda} - dE_{-\lambda}) \ ,$$

where $\mathcal{E} = (E_{\pm\lambda})_{\lambda>0}$ is the spectral family associated with $(L+U)$.

Now let us define the operator $\mathcal{B}$ as $\mathcal{B}dE_{\pm\lambda} = dE_{\mp\lambda}$. We obtain

$$\begin{aligned} \mathcal{B}(L+U) &= \mathcal{B}\int_0^{\|(L+U)\|} \lambda(dE_{+\lambda} - dE_{-\lambda}) \\ &= -\int_0^{\|(L+U)\|} \lambda(dE_{+\lambda} - dE_{-\lambda})\mathcal{B} = -(L+U)\mathcal{B} \ . \end{aligned}$$

That is, $\mathcal{B}$ anti-commutes with $(L+U)$, and the proof is complete.

Because of this theorem we assume that in the splitting $A = D - L - U$ the operator $(L+U)$ has a symmetric spectrum. Then the anti-reduction properties are satisfied.

Notice that the two subspaces $\mathcal{H}_\pm$ are in some sense identical. Therefore the coarse space could be identified with either $\mathcal{H}_+$ or $\mathcal{H}_-$. Hence we take (arbitrarily) $\mathcal{H}_+$ as our coarse or auxiliary space, whereas $\mathcal{H}$ is in the multilevel terminology the fine space.

5.2.1 The Twolevel Algorithm

We now assume that the anti-reduction properties hold and consequently describe the resulting twolevel scheme. That is, in order to solve the operator equation $Au = f$ on the space $\mathcal{H}$, we first have to define the coarse problem $A_c u_c = f_c$ on $\mathcal{H}_c \subset \mathcal{H}$, such that u_c represents part of the entire solution u.

This coarse problem is constructed by means of the two operators of prolongation and restriction operators which are defined using the splitting formulation. The first of them

will be denoted by $P : \mathcal{H}_c \to \mathcal{H}$, and the second by $R : \mathcal{H} \to \mathcal{H}_c$. Then we can use the Galerkin approach and define $A_c = RAP$ and $f_c = Rf$.

So since we have chosen $\mathcal{H}_+$ as the 'coarse' space, we denote it by $\mathcal{H}_c$. Then P and R were given in the previous chapter by (4.23) and (4.26), which we rewrite here

$$P = \{I + D^{-1}(L+U)\}|_{\mathcal{H}_c} \ , \tag{5.6}$$

and

$$R = P_+ + (L+U)D^{-1}P_- \ . \tag{5.7}$$

Hence we compute the operator product RAP using the splitting $A = D - L - U$ and the fact that the anti-reduction properties hold. Therefore we obtain

$$A_c = \{D - (L+U)D^{-1}(L+U)\}|_{\mathcal{H}_c} \ . \tag{5.8}$$

Notice that the subspace $\mathcal{H}_c$ is *invariant* under the action of $\tilde{A} = \{D-(L+U)D^{-1}(L+U)\}$, i.e., $\tilde{A}\mathcal{H}_c \subset \mathcal{H}_c$ [47]. Moreover $\tilde{A}\mathcal{H}_- \subset \mathcal{H}_-$ and $\mathcal{H}_+$ (that is $\mathcal{H}_c$) is said to *reduce* $\tilde{A}$ [47]. The point is that we can consider solely the restriction of this operator given by (5.8). Now, in order to construct a twolevel procedure we need to prove the invertibility of A_c. This is done by the following

Lemma 7 *Assume that the conditions of theorem 8 are satisfied together with the requirement that both D and $(L+U)$ have bounded inverse and the anti-reduction properties hold. Then A_c^{-1} exists and is bounded.*

Proof. Let us denote $T = D^{-1}(L+U)$. Because of theorem 8 (which applies to T with the substitution $L \to 0$ and $U \to L+U$), the spectrum $\sigma(T)$ of T lies in the interior of the unit circle. Therefore $I - T^2$ is continuously invertible [65]. Further we can rewrite $\tilde{A} = D(I - T^2)$ and the existence and boundedness of $\tilde{A}^{-1}$ follows. This means that $\tilde{A}\mathcal{H}_c = \mathcal{H}_c$, and A_c^{-1} is just given by $\tilde{A}^{-1}|_{\mathcal{H}_c}$.

So we have completed the presentation of all components of a twolevel algorithm. With the prolongation and restriction operators, given by (5.6) and (5.7), we have obtained a coarse problem which is well defined. It remains to investigate the relationship

between the coarse solution $u_c = A_c^{-1} f_c$ and the solution of $Au = f$. For this purpose we now prove a theorem which shows this connexion

Theorem 11 *Assume that the conditions of lemma 7 are satisfied. Denote with $u_c \in \mathcal{H}_c$ the solution of the coarse problem $A_c u_c = f_c$, $f_c = Rf$. Then $u = Pu_c + D^{-1}P_- f$ solves the fine equation (5.1).*

Proof. Let us apply A to $Pu_c + D^{-1}P_- f$. We have

$$\begin{aligned} A(Pu_c + D^{-1}P_- f) &= APu_c + AD^{-1}P_- f \\ &= (D - L - U)(I + D^{-1}(L + U))u_c + (D - L - U)D^{-1}P_- f \\ &= Du_c - (L + U)D^{-1}(L + U)u_c + (I - (L + U)D^{-1})P_- f \\ &= \{D - (L + U)D^{-1}(L + U)\}u_c + (I - (L + U)D^{-1})P_- f \\ &= f_c + (I - (L + U)D^{-1})P_- f \\ &= (P_+ + (L + U)D^{-1}P_-)f + (I - (L + U)D^{-1})P_- f \\ &= (P_+ + P_-)f = f \ . \end{aligned}$$

This theorem shows how a twolevel solution procedure works. First a coarse problem must be defined and solved and second the result has to be "prolongated" to obtain the solution of the given equation (in this form the method could be considered of an extension of the reduction algorithm).

We remark that the exact result given above has been obtained because of the special choice of P and R and thanks to the anti-reduction properties. However, it is interesting to consider some approximations to the method just described. For this purpose let us denote with $d = A\tilde{u} - f$ and $e = \tilde{u} - u$ the *defect* and the *error* relative to the approximation $\tilde{u}$ to the solution u. Obviously the two equations $Au = f$ and $Ae = d$ are equivalent.

Now, suppose that there exists a suitable choice of ω and of the splitting operators D, L and U such that the application of the iterative scheme (5.2) gives a solution $\tilde{u}$ whose error e, and corresponding defect d provides the approximation $e = Pe_c + D^{-1}P_- d \simeq Pe_c$.

Then the coarse level correction becomes $u = \tilde{u} - Pe_c$ and a new approximation to the solution is obtained. In this way an iterative procedure is defined which mimics the the standard twolevel method.

5.2.2 Conclusion

With the help of the generalization of classical iterative methods for the approximate solution of $Au = f$ on a Hilbert space $\mathcal{H}$ we have extended the twolevel method to solve this operator equation. This result has been possible through the study of algebraic multilevel methods. In fact, we have found that, in the algebraic case, the definition of the coarse space can be done by means of the same splitting formulation which is used to define some of the iterative methods used in ML computations. Following this result we have proved that it can be extended also to the infinite dimensional case. In this way we have constructed a subspace $\mathcal{H}_c \subset \mathcal{H}$ which in many respects looks like the coarse level of the ML theory. In particular we have proved that the existence of this space can be related to the spectral properties of the Jacobi iterative operator. Then we have explicitly given suitable prolongation and restriction operators relative to $\mathcal{H}_c$. In the multilevel formulation these operators provide, using the Galerkin approach, a well defined "coarse" problem. Finally we have obtained the solution of the fine equation on $\mathcal{H}$ in terms of that of the reduced problem given on the coarse space.

One should notice that this algorithm applies, for example, to the interesting class of the Fredholm integral equations of the second kind with symmetric kernels ($D = I$, $U = L^*$). In addition these results apply to a large part of the class of *consistently ordered* matrices [76], because of the spectral properties of the associated Jacobi iteration. Notice that these matrices are very important in applications since the discretization of boundary value problems leads naturally to them.

Conclusions

In this thesis we have presented our results on the application and theoretical investigation of multilevel methods. We started out with the development of the full multilevel scheme which we have applied to solve some classes of problems. The fist application was to solve three dimensional Dirichlet boundary value problems, through which we have shown the actual efficiency of the ML approach. Then, using Burgers' equation as model problem, we have described the implementation of a multilevel method for evolutionary equations. After this development of ML methods for partial differential equations we have applied the ML computational principles to integral equations. In fact, we have analyzed and have used the FAS-FMG algorithm for the resolution of a specific system of non linear integral equations, namely the thermodynamic Bethe ansatz equations.

In all these applications we have experienced the role of each component of a multilevel scheme and have observed their mutual and problem dependence. This last remark was then the starting point of our study of algebraic multilevel methods. Actually, the choice of the various ML components is based on the many features of the problem at hand, but sometimes, it is difficult to handle all aspects of the system to be solved. Therefore the algebraic approach helps to construct a ML method which adapt itself to process correctly the given problem. Through this study we have found that the splitting formalism, which is used to define many of the standard iterative techniques, can be used also to define a pure algebraic method. That is a numerical scheme where the coarse space, the transfer operators and the relaxation procedure are derived by means of this formalism.

Thanks to this result we were able to derive the standard multilevel approach and estimated sharply the rate of convergence of the resulting ML cycle. Moreover the splitting approach has allowed us to present a first extension of a twolevel method for operator equations in infinite dimensional Hilbert spaces. That is we have formulated a method which applies to systems of linear equations or equally to continuous problems.

We see that the application of multilevel methods is necessary in order to enlarge and improve our knowledge of these methods and its use is convenient to solve efficiently a given problem. On the other hand, the theoretical investigation is always source of new

ideas which help to improve the implementation and allow the extension of the multilevel strategy to new areas of scientific research.

Bibliography

[1] D. Bai and A. Brandt, *Local Mesh Refinement Multilevel Techniques. SIAM J. Sci. Stat. Comput.* 8 (1987) 109-134.

[2] C.T.H. Baker, *The numerical treatment of integral equations.* Oxford University Press, London, 1977.

[3] N.S. Bakhvalov, *On the convergence of a relaxation method with natural constraints on the elliptic operator. USSR Computational Math. and Math. Phys.* **6** (1966) 101.

[4] D.S. Balsara and A. Brandt, *Multilevel Methods for Fast Solution of N-Body and Hybrid Systems.* In: Hackbusch-Trottenberg [41].

[5] A. Borzì and A. Koubek, *A multi-grid method for the resolution of thermodynamic Bethe ansatz equations. Comp. Phys. Commun.* **75** (1993) 118-126.

[6] A. Borzì, *Burgers Equation and Multi-Grid Techniques.* SISSA-ISAS 204/92/FM. Submitted to [43].

[7] A. Borzì, *Algebraic Twolevel Methods and Tridiagonal $\mathcal{M}$-Matrices.* SISSA-ISAS 28/93/FM. Submitted to *IMA Journal of Numerical Analysis.*

[8] A. Borzì, *On the Extension of the Twolevel Method for Operator Equations in Hilbert Space.* SISSA-ISAS 81/93/FM. Submitted to [43].

[9] A. Brandt, *Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solution to Boundary-Value Problems.* In: H. Cabannes and R. Temam (eds.), Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris 1972. *Lecture Notes in Physics* **18**, Springer-Verlag, Berlin, 1973.

[10] A. Brandt, *Multi-Level Adaptive Solutions to Boundary-Value Problems. Math. Comp.* **31** (1977) 333-390.

[11] A. Brandt and N. Dinar, *Multigrid Solutions to Elliptic Flow Problems.* In: S.V. Parter (ed.), *Numerical Methods for Partial Differential Equations*, Academic Press, New York, 1979.

[12] A. Brandt, S. McCormick and J. Ruge, *Multigrid Methods for Differential Eigenproblems. SIAM J. Sci. Stat. Comput.* 4 (1983) 244-260.

[13] A. Brandt, *Multi-grid techniques: 1984 guide with applications to fluid dynamics.* GMD-Studien. no 85, St. Augustin, Germany, 1984.

[14] A. Brandt, *Algebraic Multigrid Theory: The Symmetric Case. Appl. Math. Comp.* **19** (1986) 23-56.

[15] A. Brandt, *Rigorous Local Mode Analysis of Multigrid.* Lecture at the 2nd European Conference on Multigrid Methods, Cologne, Oct. 1985. Research Report, The Weizmann Institute of Science, Israel, Dec. 1987.

[16] A. Brandt and A. Lanza, *Multigrid in general relativity: I. Schwarzschild spacetime. Class. Quantum Grav.* **5** (1988) 713-732.

[17] A. Brandt and A.A. Lubrecht, *Multilevel Matrix Multiplication and Fast Solution of Integral Equations. J. Comp. Phys.* **90** (1990) 348-370.

[18] A. Brandt and J. Greenwald, *Parabolic Multigrid Revisited.* In: Hackbusch-Trottenberg [41].

[19] A. Brandt, *Multilevel computations of integral transforms and particle interactions with oscillatory kernels. Comp. Phys. Commun.* **65** (1991) 24-38.

[20] A. Brandt, D. Ron and D.J. Amit, *Multi-Level Approaches to Discrete-State and Stochastic Problems.* In: Hackbusch-Trottenberg [40].

[21] A. Brandt, *Multigrid Methods in Lattice Field Computations. Nucl. Phys.* (Proc.Suppl.) **B21** (1992) 1-45.

[22] L.M. Delves and J.L. Mohamed, *Computational Methods for integral equations.* Cambridge University Press, Cambridge, 1985.

[23] J.E. Dendy, Jr., *Black Box Multigrid. J. Comp. Phys.* **48** (1982) 366.

[24] C.C. Douglas, *Multi-Grid Algorithms with Applications to Elliptic Boundary-Value Problems. SIAM J. Numer. Anal.* **21** (1984) 236.

[25] R.P. Fedorenko, *A relaxation method for solving elliptic difference equations. USSR Computational Math. and Math. Phys.* **1** (1962) 1092.

[26] R.P. Fedorenko, *The rate of convergence of an iterative process. USSR Computational Math. and Math. Phys.* 4 (1964) 227.

[27] C.A.J. Fletcher, *Burgers' Equation: A Model for All Reasons.* In: J. Noye (ed.), *Numerical Solutions of Partial Differential Equations*, Proceedings, University of Melbourne, Aug. 1981, North-Holland, Amsterdam, 1982.

[28] C.A.J. Fletcher, *Computational Techniques for Fluid Dynamics, Vol.I.* Springer-Verlag, Heidelberg, 1988.

[29] G.F. Forsythe and W.R. Wasow, *Finite-Difference Methods for Partial Differential Equations.* John Wiley & Sons, New York, 1964.

[30] J. Gazdag, *Numerical Convective Schemes Based on Accurate Computation of Space Derivatives. J. Comp. Phys.* **13** (1973) 100.

[31] J. Goodman and A.D. Sokal, *Multigrid Monte Carlo for lattice field theories. Phys. Rev. Lett.* **56** (1986) 1015.

[32] W. Hackbusch, *A multi-grid method applied to a boundary problem with variable coefficients in a rectangle.* Report 77-17, Institut für Angewandte Mathematik, Universität Köln, 1977.

[33] W. Hackbusch, *On the computation of approximate eigenvalues and eigenfunctions of elliptic operators by means of a multi-grid method. SIAM J. Numer. Anal.* **16** (1979) 201-215.

[34] W. Hackbusch, *Convergence of Multi-Grid Iterations Applied to Difference Equations. Math. Comp.* **34** (1980) 425-440.

[35] W. Hackbusch, *Multi-Grid Convergence Theory.* In: Hackbusch-Trottenberg [39].

[36] W. Hackbusch, *Parabolic Multi-Grid Methods.* In: R. Glowinski and J.L. Lions (eds.), *Computing Methods in Applied Sciences and Engineering*, VI. Proc. of the sixth international symposium, Versailles, Dec.1983, North-Holland, Amsterdam, 1984.

[37] W. Hackbusch, *Multi-Grid Methods and Applications.* Springer-Verlag, Heidelberg, 1985.

[38] W. Hackbusch, *Multi-Grid Methods of the Second Kind.* In: D.J. Paddon and H. Holstein (eds.), *Multigrid Methods for Integral and Differential Equations.* Claredon Press, Oxford, 1985.

[39] W. Hackbusch and U. Trottenberg (eds.), *Multi-Grid Methods*, Proceedings, Köln-Porz, Nov. 1981, *Lecture Notes in Mathematics* **960**, Springer-Verlag, Berlin, 1982.

[40] W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods II*, II. Proc. of the European Conference on Multigrid Methods, Cologne, Oct. 1985, *Lecture Notes in Mathematics* **1228**, Springer-Verlag, Berlin, 1986.

[41] W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods III*, III. Proc. of the European Conference on Multigrid Methods, Bonn, Oct. 1990, Birkhäuser, Berlin, 1991.

[42] P.W. Hemker and H. Schippers, *Multiple grid methods for the solution of Fredholm integral equations of the second kind. Math. Comp. 36* (1981) 215-232.

[43] P.W. Hemker and P. Wesseling (eds.), IV. Proc. of the European Multigrid Conference, EMG'93, Amsterdam, Jul. 1993, to appear.

[44] T. Kalkreuter, *Multigrid Methods for the Computation of Propagators in Gauge Fields.* DESY 92-158.

[45] E. Katzer, *Multigrid Methods for Hyperbolic Equations.* In: Hackbusch-Trottenberg [41].

[46] T. Klassen and E. Melzer *Nucl. Phys.* **B350** (1990) 635.

[47] E. Kreyszig, *Introductory Functional Analysis with Applications.* John Wiley & Sons, New York, 1978.

[48] A. Lanza, *Multigrid in general relativity II: Kerr Space-Time. Class. Quantum Grav.* **9** (1992) 677.

[49] A. Lanza, *Self-graviting thin discs around rapidly rotating black holes. Astrophys. J.* **389** (1992) 141.

[50] A. Lanza and M.R. Dubal, *An Application of the Multi-Grid Method to the Construction of Initial Data for Brill Waves. Computers Math. Applic.* **19** (1990) 77-85.

[51] L. Lapidus and G.E. Pinder, *Numerical Solution of Partial Differential Equations in Science and Engineering.* John Wiley & Sons, New York, 1982.

[52] G. Mack and S. Meyer, *The effective action from multigrid Monte Carlo. Nucl. Phys.* (Proc. Suppl.) **17** (1990) 293.

[53] J. Mandel et al., *Copper Mountain Conference on Multigrid Methods.* IV. Proc. of the Copper Mountain Conference on Multigrid Methods, Copper Mountain, April 1989, SIAM, Philadelphia, 1989.

[54] M.J. Martins, *Phys. Rev. Lett.* **67** (1991) 419.

[55] M.J. Martins, *Phys. Lett.* **B240** (1990) 404; P. Christe, M.J. Martins, *Mod. Phys. Lett.* **A5** (1990) 2189; A. Koubek, M.J. Martins, G. Mussardo *Nucl. Phys.* **B368** (1992) 591.

[56] M.L. Mehta, *Matrix Theory, Selected Topics and Useful Results.* Hindustan Publishing Corporation, Delhi, 1989.

[57] U. Miekkala and O. Nevanlinna, *Convergence of Dynamic Iteration Methods for Initial Value Problems. SIAM J. Sci. Stat. Comput.* **8** (1987) 459-482.

[58] A.R. Mitchell and R. Wait, *The Finite Element Method in Partial Differential Equations.* John Wiley & Sons, Chichester, 1977.

[59] H.D. Mittelmann and H. Weber, *Multi-Grid Methods for Bifurcation Problems. SIAM J. Sci. Stat. Comput.* **6** (1985) 49.

[60] G. Mussardo, *Off-critical statistical models: factorized scattering theories and Bootstrap program. Phys. Rep.* **218** (1992) 215.

[61] R.A. Nicolaides, *On multiple grid and related techniques for solving discrete elliptic systems. J. Comp. Phys.* **19** (1975) 418-431.

[62] J.M. Ortega, *Numerical Analysis, A Second Course.* Academic Press, New York, 1972.

[63] B. Oskam and J.M.J. Fray, *General Relaxation Schemes in Multigrid Algorithms for Higher-Order Singularity Methods. J. Comp. Phys.* **48** (1982) 423-440.

[64] W.M. Patterson, 3rd, *Iterative Methods for the Solution of a Linear Operator Equation in Hilbert Space - A Survey. Lecture Notes in Mathematics* **394**. Springer-Verlag, Heidelberg, 1974.

[65] W.V. Petryshyn, *On the Generalized Overrelaxation Method for Operator Equations. Proc. Amer. Math. Soc.* **14** (1963) 917-924.

[66] W.V. Petryshyn, *Remarks on the Generalized Overrelaxation and the Extrapolated Jacobi Methods for Operator Equations in Hilbert Space. J. Math. Anal. Appl.* **29** (1970) 558-568.

[67] W. Pogorzelski, *Integral Equations and their Applications. Vol. I.* PWN-Polish Scientific Publishers, Warsaw, 1966.

[68] W.H. Press and S.A. Teukolsky, *Multigrid Methods for Boundary Value Problems.* I. and II.. *Computers in Physics* Sep./Oct. and Nov./Dec. 1991.

[69] H.-J. Reinhardt, *Analysis of Approximation Methods for Differential and Integral Equations.* Springer-Verlag, New York, 1985.

[70] R.D. Richtmyer and K.W. Morton, *Difference Methods for Initial-Value Problems.* John Wiley & Sons, New York, 1967.

[71] J.W. Ruge and K. Stüben, *Algebraic Multigrid.* In: S.F. McCormick, *Multigrid Methods.* SIAM, Frontiers in Applied Mathematics, Philadelphia, 1987.

[72] P.L. Sachdev, *Nonlinear diffusive waves.* Cambridge University Press, Cambridge, 1987.

[73] H. Schippers, *Application of Multigrid Methods for Integral Equations to Two Problems from Fluid Dynamics. J. Comp. Phys.* **48** (1982) 441-461.

[74] J. Schröder, U. Trottenberg and K. Witsch, *On Fast Solvers and Applications. Lecture Notes in Mathematics* **631**. Springer-Verlag, Berlin, 1976.

[75] S.F. Shandarin and Ya. B. Zel'dovich, *The large-scale structure of the universe: Turbulence, intermittency, structures in a self-graviting medium. Rev. Mod. Phys.* **61** (1989) 185-220.

[76] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis.* Springer-Verlag, New York, 1980.

[77] K. Stüben and U. Trottenberg, *Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications.* In: Hackbusch-Trottenberg [39].

[78] S. Ta'asan, *Multigrid Methods for Locating Singularities in Bifurcation Problems. SIAM J. Sci. Stat. Comput.* **11** (1990) 51-62.

[79] C.-A. Thole and U. Trottenberg, *Basic smoothing procedures for the multigrid treatment of elliptic 3D-operators. Appl. Math. Comp.* **19** (1986) 333-345.

[80] S. Vandewalle and R. Piessens, *Efficient Parallel Algorithms for Solving Initial-Boundary Value and Time-Periodic Parabolic Partial Differential Equations. SIAM J. Sci. Stat. Comput.* **13** (1992) 1330-1346.

[81] W.T. Vetterling, S.A. Teukolsky, W.H. Press and B.P. Flannery, *Numerical Recipes: The art of scientific computing.* Cambridge University Press, New York, 1985.

[82] P. Wesseling, *A survey of Fourier smoothing analysis results.* In: Hackbusch-Trottenberg [41].

[83] J.H. Wilkinson, *The Algebraic Eigenvalue Problem.* Oxford University Press, Oxford, 1965.

[84] D.M. Young, *Iterative Solution of Large Linear Systems.* Academic Press, New York, 1971.

[85] D.M. Young and R.T. Gregory, *A Survey of Numerical Mathematics, Vol.II.* Addison-Wesley Publishing Company, 1973.

[86] A.B. Zamolodchikov, *JETP Letters* **46** (1987) 160; *Int. Journ. Mod. Phys.* **A3** (1988) 743; *Advanced Studies in Pure Mathematics* **19** (1989) 641.

[87] Al.B. Zamolodchikov, *Nucl. Phys.* **B342** (1990) 695.

[88] Al.B. Zamolodchikov, *Nucl. Phys.* **B358** (1991) 497.

[89] P.M. de Zeeuw, *Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. J. Comput. Appl. Math.* **33** (1990) 1.