CWI Tracts

Managing Editors

K.R. Apt (CWI, Amsterdam)
M. Hazewinkel (CWI, Amsterdam)
J.K. Lenstra (Eindhoven University of Technology)

Editorial Board

W. Albers (Enschede)
P.C. Baayen (Amsterdam)
R.C. Backhouse (Eindhoven)
E.M. de Jager (Amsterdam)
M.A. Kaashoek (Amsterdam)
M.S. Keane (Amsterdam)
H. Kwakernaak (Enschede)
J. van Leeuwen (Utrecht)
P.W.H. Lemmens (Utrecht)
M. van der Put (Groningen)
M. Rem (Eindhoven)
H.J. Sips (Delft)
M.N. Spijker (Leiden)
H.C. Tijms (Amsterdam)

CWI
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
Telephone 31 - 20 592 9333, telex 12571 (mactr nl),
telefax 31 - 20 592 4199

CWI is the nationally funded Dutch institute for research in Mathematics and Computer Science.

Iterative methods for non-linear partial differential equations

J.M.L. Maubach

ISBN 90 6196 421 0
NUGI-code: 811

Copyright © 1994, Stichting Mathematisch Centrum, Amsterdam
Printed in the Netherlands

Table of Contents

Preface

Over the past decades, much attention has been paid to the use of *preconditioned iterative methods* for the solution of self-adjoint elliptic boundary value problems. A thorough investigation has shown that these iterative methods are eminently applicable for the solution of most frequently occurring elliptic problems. However, elliptic boundary value problems are relatively simple, and the increase of the computational capacity of the large computer mainframes nowadays creates a growing need to investigate more complex physical problems, such as those found in the oil recovery, airplane and semiconductor industry. As there is a growing demand for the investigation of these – often time-dependent – physical problems, the study of the behaviour of iterative methods for this type of problems becomes of interest. In view of this development, this thesis considers the use of such methods for the global finite element technique applied to initial value problems. This application, recently shown to be one of a growing interest, has been far less studied so far, the cause being undoubtedly the complexity entailed by non self-adjoint differential equations, which can for instance give rise to a solution with layers moving in time.

In general, the performance of iterative methods for the solution of non-linear time-dependent partial differential equations depends strongly on the applied discretization technique, since this technique determines the subsequent systems of equations to be solved. If the discretization technique is by a finite element method, the following discretization degrees of freedom can be distinguished.

- The construction and refinement of a computational grid covering the computational domain.
- The type of finite elements and associated basis representing the approximate solution on this grid.
- The non-linear solution method to linearize either the partial differential equation or the related system of equations.
- The type of – preconditioned iterative – solution method for the solution of the resulting assembled systems of linear equations.

Each of these points will influence the behaviour of the iterative solution

method since it influences the coefficient matrix of the system of equations as well as the coefficient matrix of the preconditioner. For many physical non-linear time-dependent problems the total computational solution time involved is dominated by the construction of a suitable computational grid and the assembly of the resulting linearized system of equations. Therefore, even iterative methods of optimal complexity, i.e., with a number of arithmetic operations proportional to the degrees of freedom, may not lead to effective overall solution methods. In order to overcome this problem, one may integrate the grid construction, matrix and preconditioner assembly and iterative method into a non-linear iterative solution method. Alternatively, one can try to optimize the separate discretization parameters and study the behaviour of preconditioned iterative methods for linear systems of equations separately, using iterative methods to be found in the literature.

In order to gain insight in the behaviour of the types of linear iterative solution methods for the discretization parameters to be proposed, the latter approach of separate analysis has been followed in this thesis. This will hopefully contribute to the construction of effective iterative solution methods in the future.

To solve the non-linear time-dependent equations, a continuous global time-space finite element discretization technique will be examined. This technique uses a finite element approximation in time and space simultaneously, for a relatively large time-period $(t_{j-1}, t_j]$, called *time-slab*. The technique is called *continuous* since the approximate solution on $t = t_j$ will be taken to be an initial value for the solution to be approximated on the next time-slab $(t_j, t_{j+1}]$, leading to a finite element solution in time-space which is continuous throughout the whole computational domain. This is in contrast to the so-called discontinuous global time-space finite element techniques which are said to be *discontinuous* since the solution on each next time-slab only approximately satisfies the initial value provided on the previous time-slab at $t = t_j$. This allows the approximated solution on the computational domain to be – slightly – discontinuous at the interfaces between time-slabs. The advantage of the latter methods is the possibility to adjust the computational grid at the interface of two time-slabs, but as it turns out to be fairly easy to use locally refined grids in two and three space dimensions (see chap-

ter 5 and Bänsch [12]), this becomes less important. Early work on the discontinuous methods in time-space can be found in Hulme [15] for ordinary differential equations, and in Jamet [16] for parabolic problems. More recent publications can be found in Johnson [17], [18] and Aziz and Monk [9]. For publications concerning the continuous time-slabbing technique see Axelsson and Maubach [5], [6], of which a summary can be found in chapters 1 – 4.

In order to obtain an accurate approximate solution on a given time-slab – as pointed out before, the solution of the differential equation may have layers moving in time – a coarse initial computational grid of simplices covering the time-space domain has to be provided (simplex in two and three space dimensions stands for triangle resp. tetrahedron). Considering problems with one space and one time dimension, a two-dimensional computational grid is constructed from a coarse initial computational grid with the use of local *newest vertex bisection refinement* as shown in Mitchell [19] or Sewell [20]. In the three-dimensional case, i.e., two space dimensions combined with the time dimension, analogous refinement methods exist, as is shown by [12]. Contrary to the two-dimensional regular refinement method of a triangle into four congruent children, see for instance Deuflhard and Leinen [14], this newest vertex bisection grid refinement technique has rarely been examined and/or applied before and is therefore investigated in detail in chapter 5. The approximate finite element solution on a given time-slab will be represented either on a standard nodal finite element basis, as explained in Axelsson and Barker [3], or on a hierarchical finite element basis as in Yserentant [23].

First, this bisection refinement turns out to be simple to analyze and effective, i.e, the computational time involved to track and refine along moving layers is relatively small compared to the computational time used by the linear iterative solver for the solution of the systems of equations. Further, as the bisection refinement technique is rather simple, a *sparsity pattern* analysis of the resulting matrix can be given (see section 5.5), leading to a finite element row-wise ordered *matrix storage method*, different from the classical row-wise ordered matrix storage method to be found in [3] or in Bank and Smith [11]. This enables numerical tests with matrices explicitly represented on a hierarchical

finite element basis, contrary to results in Bank et. al. in [10], where the hierarchical matrix is never assembled. Such tests exploiting classical preconditioning techniques for an hierarchical basis, to the best knowledge of the author, have never been published in the literature before.

Furthermore, related to the new matrix storage method, the finite dimensional vector representation to be used for the approximate solution is also non-standard. Contrary to classical grid point numbering strategies, where one tries to optimize the bandwidth of the resulting matrix (see [3]), in this thesis the points are numbered such that the numbering reflects the level of local refinement applied to create them, as explained in chapter 5. Since the numbering of the grid points does not change the resulting finite element basis, it does not influence the rate of convergence of unpreconditioned iterative methods as this rate is only determined by the eigenvalues of the matrix. However, if one uses *incomplete Gaussian factorization ILU* for the construction of an accelerating preconditioner, the spectrum of eigenvalues of the preconditioner will depend on this numbering due to the incomplete nature of the factorization and thus influence the rate of convergence of the accelerated iterative solution method.

After the determination of the grid and the local refinement technique, the finite element basis functions to be used for the solution on one time-slab are taken to be of the exponentially weighted type as in chapter 2 and/or the upwind Petrov-Galerkin type as in chapter 3. This choice has no influence on the matrix storage or sparsity pattern analysis referred to above. The weighing or upwind version of the standard nodal or hierarchical basis functions is used since this yields better results on coarser grids. For the relatively recent continuous global time-space finite element discretization this is shown by *discretization error estimates* for a variety of physically interesting classes of problems in the first chapters of this thesis. Although most discretization error estimates are valid for finite element basis functions of arbitrary high polynomial degree, only the linear case, and in chapter 7 the quadratic case, is considered in the numerical tests presented.

For the linearization of the partial differential equation on a certain time-slab the damped inexact Newton algorithm as presented in [13] is used.

One can follow two lines of analysis. One possibility is to use the finite element method directly for the discretization of the time-dependent non-linear differential equation, resulting in a non-linear algebraic equation, which can be solved by the damped inexact Newton algorithm. This approach has been followed in some of the numerical tests in the first chapters where the initial coarse grid was refined prior to the application of the non-linear Newton method.

Alternatively, one can use the Newton method to construct a sequence of linear partial differential equations, to be solved by a finite element method, but not necessarily on identical grids. Using this approach one can combine the Newton method with *adaptive refinement* of the grid until the desired accuracy is obtained. This approach has been used in chapter 5.

The use of a Newton-like non-linear solution algorithm requires the assembly of the *Jacobian matrix*, involving the computation of derivatives. As will be shown for all special but frequently occurring cases to be presented, this assembly is relatively easy and turns out to be not much more expensive than the computation of the *gradient*. It was often considered to be an expensive task, see e.g. [7] where the Jacobian matrix is only updated in regions where the solution varies relatively much. Chapter 7 demonstrates that for the cases to be considered in this chapter there is no reason to avoid the updating and assembly of the Jacobian matrix.

Finally, the iterative methods are presented in the last chapter. Most of them can be found in [1], [2], [3], [21] and [22]. They are well-known and have been tested thoroughly in the literature for non self-adjoint static problems. Several numerical tests in this thesis demonstrate their performance for the systems of equations emanating from the continuous global time-space finite element technique, showing their applicability also to this type of initial value problem. All tests involve an acceleration by a preconditioner obtained by an incomplete Gaussian factorization, and are therefore influenced by the grid point numbering as determined by the local grid refinement technique used. It should be noted that this type of Gaussian preconditioning is probably far from optimal. For elliptic self-adjoint boundary value problems on regular grids there exist optimal order algebraic multi-level iterative solution methods (see

e.g. [4] or [8]) which can in certain cases be extended to grids obtained by the local bisection refinement proposed, as is shown in chapter 6. The construction of optimal order iterative methods for the non self-adjoint type of problems examined in this thesis still remains an open problem.

The remainder of this thesis is organized as follows. Chapter 1 introduces the classes of partial differential equations to be considered, defines the notations to be used in the sequel and presents the continuous global time-space finite element solution method. It is shown that this method, which is related to a Petrov-Galerkin variational formulation, leads to a *global discretization error* bounded by the maximum *local discretization errors* over all time-slabs. The solution of the differential equation per slab and the applicability of the theory presented is demonstrated with the use of a simple example differential equation, for which an estimate for the local discretization errors is provided. Next, chapter 2 defines the continuous global time-space finite element discretization method for time-dependent problems in one space dimension and provides discretization error estimates in order to show that the local discretization errors are relatively easy to control. An extension of the presented theory to the multi-dimensional Petrov Galerkin streamline upwind case, is considered in chapter 3. At the end of this chapter a brief discussion concerning the applicability of global finite element techniques to more general cases can be found. Then, after the first chapters dealing with a single partial differential equation, chapter 4 introduces the continuous and discontinuous time-slabbing technique for the time-dependent Stokes system of partial differential equations. Local discretization error estimates are provided and the inf-sup condition related to the global time-space finite element solution is studied in detail.

Chapter 5 investigates the finite element basis used in all preceeding chapters in relation with the underlying grid geometry and refinement. The local grid bisection refinement is studied in detail, the sparsity patterns of the resulting – hierarchical – matrices are investigated and a solution algorithm, combining the grid refinement method and the non-linear Newton solution method, is provided. As the grid refinement method proposed leads to level structured matrices, optimal order algebraic multi-level iterative solution methods for certain locally refined grids are considered in chapter 6.

The Jacobian matrix underlies all error estimates in the first chapters and is needed for the non-linear Newton solution method. The factorization of this matrix is studied in chapter 7, investigated are the properties of the quadrature rules used for the computation of its entries. Finally, the damped inexact Newton algorithm together with the iterative methods are presented in chapter 8. Studied is the solution of the linearized systems of equations taking into account the stopping criterion, the Jacobian matrix assembly and its preconditioning. The iterative solution methods presented are the basis for all numerical tests presented in this thesis.

All chapters are related to reports which are published, to appear, submitted or in preparation. Every chapter will be accompanied by information, explaining to which reports its sections are related. Some of them contain additional unpublished sections explaining the basic principles underlying the theory presented. To avoid overlap, some – parts of – sections have been deleted, under reference to an earlier chapter, and cross references have been added. The number of numerical tests has been reduced by replacing the original – published – tests by new ones. Further, in order to enable a more uniform presentation, some notations originally used in the reports have been adjusted. However, different fields in physics prefer their own notation for the *diffusion tensor*, to be introduced in chapter 1. In order to adapt to conventional notational rules the tensor will be denoted as follows.

- Related to the Stokes and Navier-Stokes equation, the diffusion tensor, being the inverse of the Reynolds number, is denoted by ν.
- Related to potential flow theory, it will denoted by ρ, the non-linear potential flow density.
- Applied to potential electromagnetic problems, the tensor, referring to the electromagnetic reluctivity, is denoted by ν.
- It is denoted by ϵ in the case of a singular perturbed parabolic equation.

Finally, for static differential equations, the domain of definition will always be denoted by Ω, throughout all chapters. If the problem is time-dependent, then Ω will denote the space domain and Q will denote the time-space domain in order to avoid confusion. Without regard of the time-dependency, the grid is denoted by $\mathcal{Q}$ and all definitions related to

the grid and the finite element basis defined thereon are with the use of calligraphic characters.

Acknowledgements

The author wishes to thank The Netherlands Organization for Scientific Research N.W.O., which supported this thesis by grant nb. 60-10-06. In particular, he would like to thank the advisor prof. O. Axelsson for his guidance. Part of the research is related to invitations by prof. P. Deuflhard and dr. R. Roitzsch, Konrad-Zuse-Zentrum für Informations Technik, Berlin, Germany; prof. R. Lazarov and dr. S. Margenov, Center for Informatics and Computer Technology, Sofia, Bulgaria; prof. O. Axelsson and prof. M. Navon, Supercomputer Computations Research Institute, University of Florida, U.S.A.; prof. W. Layton, dr. J. Burkardt and dr. J. Welling, Department of Mathmatics, University of Pittsburgh resp. the Pittsburgh Supercomputing Center, Pennsylvania, U.S.A.; dr. P. Vassilevski and prof. D. Ewing, University of Wyoming, Laramie, U.S.A.; dr. K.-E. Karlsson and dr. A. Wolfbrandt, ABB Corporate Research, Västerås, Sweden.

References

[1] Aarden J.M. and Karlsson K.-E., *Preconditioned cg-type methods for solving the coupled system of fundamental semiconductor equations,* BIT, 29(1989), 916-937

[2] Axelsson O., *A generalized conjugate gradient, least square method,* Numerische Mathematik, 51(1987), 209-227

[3] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[4] Axelsson O. and Eijkhout V., *The nested recursive two-level factorization method for nine-point difference matrices,* internal report 8936 of the Department of Mathematics of the University of Nijmegen, The Netherlands, 1989

[5] Axelsson O. and Maubach J., *Stability and high order approximation of monotone evolution equations valid for unbounded time by continuous time slabbing methods,* internal report of the Supercomputer Computations Research Institute, Florida State University, Tallahassee, U.S.A., 1990

[6] Axelsson O. and Maubach J., *A time space finite element method for nonlinear convection diffusion problems,* **in** Notes on Numerical Fluid Mechanics, (Hackbush W. and Rannacher R. eds.) Vol. 30, 6-23, Vieweg, Braunschweig, 1990 [Proceedings of the Fifth GAMM-Seminar, Kiel, West Germany 1989]

[7] Axelsson O. and Nävert U., *On a graphical package for nonlinear partial differential equation problems,* **in** Information Processing 77, IFIP, 103-108, North-Holland , 1977

[8] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods I,* Numerische Mathematik, 56(1989), 157-177

[9] Aziz A.K. and Monk P., *Continuous finite elements in space and time for the heat equation,* Mathematics of Computation, 52(1989), 255-274

[10] Bank R.E., Dupont T.F. and Yserentant H., *The hierarchical basis multigrid method,* Preprint SC 87-1 (1987), Konrad-Zuse-Zentrum für Informationstechnik, Berlin

[11] Bank R.E. and Smith R.K., *General sparse elimination requires no permanent integer storage,* SIAM Journal on Scientific and Statistical Computing, 8(1987), 574-584

[12] Bänsch E., *Local mesh refinement in 2 and 3 dimensions,* Impact of Computing in Science and Engineering, 3(1991), 181-191

[13] Dembo R.S., Eisenstat S.C. and Steihaug T., *Inexact Newton methods,* SIAM Journal on Numerical Analysis, 19(1982), 400-408

[14] Deuflhard P., Leinen P. and Yserentant H., *Concepts of an adaptive hierarchical finite element code,* Preprint SC 88-5, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, September 1988

[15] Hulme B.L., *Discrete Galerkin methods and related one-step methods for ordinary differential equations,* Mathematics of Computation, 26(1972), 881-891

[16] Jamet P., *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain,* SIAM Journal on Numerical Analysis, 15(1978), 912-928

[17] Johnson C., *Numerical Solution of Partial Differential Equations by the Finite Element Method,* 3rd printing, Cambridge University Press, Cambridge, 1990

[18] Johnson C., *Adaptive finite element methods for diffusion and*

convection problems, Computer Methods in Applied Mechanics and Engineering, 82(1990), 301-322

[19] Mitchell W.F., *A comparison of adaptive refinement techniques for elliptic problems,* internal report no. UIUCDCS-R-1375, Department of computer science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1987

[20] Sewell E.G.,*Automatic generation of triangulations for piecewise polynomial approximation,* Ph.D. thesis, Purdue University, West Lafayette, IN, 1972

[21] Sonneveld P., *CGS, a fast Lanczos-type solver for non-symmetric linear systems,* SIAM Journal on Scientific and Statistical Computing, 10(1989), 36-52

[22] Vorst H.A. van der, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems,* internal report of the Mathematical Institute, University of Utrecht, The Netherlands, 1990

[23] Yserentant H., *On the multilevel splitting of finite element spaces,* Numerische Mathematik, 49(1986), 379-412

1 The global time-space finite element method

The abstract and all sections except the added sections
on variational formulations and finite element discretizations
are part of: Axelsson O. and Maubach J., Stability and high order
approximation of monotone evolution equations valid
for unbounded time by continuous time-slabbing methods,
Internal report of the Supercomputer Computations
Research Institute, Florida State University, Tallahassee,
U.S.A. 1990, submitted to SIAM Journal on Numerical Analysis.

Abstract

Using amazingly straightforward and short proofs it is shown that stability for unbounded time and arbitrary high order discretization errors for evolution equations with strongly monotone operators, such as arise for non-linear parabolic problems, can be achieved using time-slabbing.

Within each time-slab one for instance can use finite elements in the time-space domain. The advantage of doing so is that for problems with layers (boundary or interior) one can easily obtain a certain order of the quality of approximation, using an order of magnitude fewer degrees of freedom than is needed for classical time-stepping methods, possibly using moving grids.

Key words: Initial value problems, Finite elements, Error bounds, Grid generation and refinement, Stokes equation
AMS(MOS) subject classifications: 65L05, 65M15, 65M50, 65M60

1.1 Introduction

When solving semidiscretized evolution equations for non-linear partial differential equations using standard time-stepping methods, it turns out to be difficult to achieve a high order of approximation in both time and space. The reason for this is that because of the stiffness of the problems a severe reduction of the order of approximation, obtainable for non-stiff problems, can occur (for details, see [17] and [28], for instance). Furthermore, classical theories for ordinary differential equations are not applicable because the order of the systems and the stiffness of the problem for semidiscretized evolution problems increases with some power of h^{-1} where h is the stepsize parameter in space.

For some results for lower (second) order time-stepping methods, see [3], [6] and [24]. In [6] and [24] it was shown that using the so called Θ-method with proper values of Θ, error estimates of up to second order of approximations are valid for unbounded time and infinitely stiff problems, if the operator satisfies the strong monotonicity condition to be presented in section 1.2. Also see [12] and [13], where it is shown that one can obtain higher than second order B-convergence for certain classes of problems.

In recent years much attention has been paid to moving finite element methods. In these, the grid is adjusted in space at every time-step to resolve steep gradients in the space variables of the solutions better, but the time-step itself is taken constant for the whole space domain. This means that the time-step must be chosen as small as the steepest gradients in time require for their solutions but this time-step may be far too small for the part of the domain where the solution is smooth. As has been indicated in [25], there is also a danger of overlooking long range effects of small eddies (circular currents), corresponding to high frequency components in such methods.

It is shown that by a time-slabbing method, i.e., a method where one recursively solves the evolution equation in a time-space domain (called slab) with fixed size, using finite element methods in time-space, one can resolve the solution where it has layers, i.e., steep gradients, more efficiently, namely with an order of magnitude fewer degrees of freedom than for the classical time-stepping method.

In addition, it is shown that this method is stable and that the error

estimates remain bounded for all times for strongly monotone evolution equations – which can be infinitely stiff – and depend only on the local errors at each time-slab. Using finite elements of arbitrarily high order, for instance a p-method, one also gets discretization error estimates in time-space of arbitrary high order. The disadvantage with the time-space finite element method is that one needs elements in a space of one dimension more than for the space variable of the evolution problem. However, as has recently been shown in [2] and [11], there now exist solution methods using a multi-level structure of the matrix for which the computational complexity is optimal – or suboptimal by a factor $\log h$ – that is, the number of required arithmetic operations is proportional, or nearly proportional, to the degrees of freedom. This means that method to be presented, which is of the implicit type, acts essentially as an explicit method as far as the computational complexity per time-step is concerned.

The method to be presented steps forward from one time-slab to the next in a continuous manner, i.e., simply by using the values at a time t_j computed for the time-slab $(t_{j-1}, t_j]$, as a Dirichlet boundary condition for the next time-slab $(t_j, t_{j+1}]$.

Methods which use time-slabbing techniques but based on a discontinuous stepping method, i.e., which allow for – small – discontinuities of the function values at the interface between two time-slabs, have been considered in [21] and [23]. The advantage of the continuous time-slabbing method presented here over these earlier methods is, that it enables the use of standard finite element packages for convection diffusion equations and in addition is simpler to implement and to analyze. Global error control is provided for, simply by controlling the local errors for each time-slab. This can for instance be done using an adaptive grid refinement method, with hierarchical basis functions. By creating hierarchically defined growing finite element spaces for each time-slab one can solve problems with various scales of physical details. The continuous method was first presented for a special application in [9] and [10]. In the present chapter more general proofs will be presented and applied for evolution equations.

The remainder of the chapter is organized as follows. In section 1.2, the type of evolution problems to be considered is introduced, after which in section 1.3 the variational formulation of the time-slabbing

method is presented. In section 1.4 the stability of the method is shown and, after section 1.5 on finite element basis functions, discretization error estimates are derived in section 1.6. In the last section some conclusions are drawn.

1.2 A class of stable evolution problems

There are two general classes of evolution problems – see (1.2.4) – that are potential candidates for the efficient application of the time-slabbing method. In order to introduce them, let $\Omega \subset \mathbb{R}^n$ be an open bounded connected domain, defining the time-space domain $Q = \Omega \times (0, \infty)$ as is shown in fig. 1.2. Then, let for an open and bounded set $X \subset \mathbb{R}^m$, $m \geq 1$, $\overline{X}$ stand for its topological closure, $C^p(X)$ denote the set of p times continuously partial differentiable functions $u: X \mapsto \mathbb{R}$, and $C^p(\overline{X}) \subset C^p(X)$ be the subset of those functions for which all the partial derivatives can be extended continuously to the boundary ∂X of X. Further, let $C^0(X)$ denote the set of continuous functions on a set X and let the partial derivatives of a vectorial function $\boldsymbol{u}$ on Ω be defined by $\underline{\nabla}\boldsymbol{u} = [\underline{\nabla}u_1, \ldots, \underline{\nabla}u_n]^T$ with

$$\underline{\nabla}u_i = [\frac{\partial}{\partial x_1}u_i, \ldots, \frac{\partial}{\partial x_n}u_i]^T \quad \text{and} \quad |\underline{\nabla}\boldsymbol{u}| = \left(\sum_{i,j=1}^{n} \left|\frac{\partial}{\partial x_i}u_j\right|^2\right)^{\frac{1}{2}}.$$

Throughout this section $\underline{\nabla}$ will denote the gradient in the space directions only. In the case of possible confusion, this gradient will be denoted by $\underline{\nabla}_{\boldsymbol{x}}$.

Related to the evolution equations, for given $0 \leq t_{j-1} < t_j$ boundary conditions $\boldsymbol{u}_c$ will be prescribed on the *cylinder surface* $\Gamma_c = \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1}: \boldsymbol{x} \in \partial\Omega \wedge t \in [t_{j-1}, t_j]\}$ and an initial value $\boldsymbol{u}_0$ will be given at the bottom boundary $\Gamma_1 = \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1}: \boldsymbol{x} \in \Omega \wedge t = t_{j-1}\}$. Here $\partial\Omega$ stands for the boundary of the space domain, which is assumed to be smooth (see [1] or [27]). Finally, let $\Gamma_D = \Gamma_1 \cup \Gamma_c$ and define for $0 < t_j < \infty$ the top boundary $\Gamma_3 = \{(\boldsymbol{x}, t) \in Q: \boldsymbol{x} \in \Omega \wedge t = t_j\}$. Boundary conditions on Γ_1 will be denoted by $\boldsymbol{u}_0$, on the cylinder surface by $\boldsymbol{u}_c$, and on the union of the latter two by $\gamma = (\boldsymbol{u}_0, \boldsymbol{u}_c)$.

In the sequel, unlike usual, $C^p(\Omega) \times C^q((t_{j-1}, t_j])$ denotes the set of functions on the time-space domain $\Omega \times (t_{j-1}, t_j]$ which are p times continuously partial differentiable in the directions of the domain Ω and q times in the time direction.

The first class of problems can be described by the nonstationary *Ladyzhenskaya model* for incompressible viscous flow

$$\begin{aligned} \frac{\partial}{\partial t}\boldsymbol{u} - \sum_{k=1}^{n} \frac{\partial}{\partial x_k}\left(\epsilon(\boldsymbol{u})\frac{\partial}{\partial x_k}\boldsymbol{u}\right) + \sum_{k=1}^{n} u_k \frac{\partial}{\partial x_k}\boldsymbol{u} + \underline{\nabla} p &= \boldsymbol{f} \\ \nabla \cdot \boldsymbol{u} &= 0 \end{aligned} \qquad (1.2.1)$$

in Q, and

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}, t) &= \boldsymbol{u}_c(t) \\ \boldsymbol{u}(\boldsymbol{x}, 0) &= \boldsymbol{u}_0(\boldsymbol{x}) \end{aligned}$$

at Γ_c and Γ_1. Here

- the solution components $\boldsymbol{u}$ and p are such that $\boldsymbol{u} = [u_1, \ldots, u_n]^T$, with $u_1, \ldots, u_n \in C^0(\overline{Q}) \cap \{C^2(\Omega) \times C^1((0, \infty))\}$ and $p \in C^0(\overline{Q}) \cap \{C^1(\Omega) \times C^0((0, \infty))\}$.
- the initial value function $\boldsymbol{u}_0$, Dirichlet boundary conditions $\boldsymbol{u}_c$ and source function $\boldsymbol{f}$ are n-dimensional vectorial functions.
- the *diffusion tensor* ϵ is a function of u.

$$\epsilon(\boldsymbol{u}) = \epsilon_0 + \epsilon_1 |\underline{\nabla} \boldsymbol{u}|^{q-2}, \qquad (1.2.2)$$

where $2 < q \leq 4$, $\epsilon_0 > 0$ and $\epsilon_1 \geq 0$.

The nonstationary Ladyzhenskaya model for incompressible viscous flow reduces for $\epsilon_1 = 0$ to the *Navier-Stokes equation*, and if in addition the non-linear convection term is neglected, the equation reduces to the *Stokes equation*. A finite element approximation method for this problem (see section 1.5) was recently studied in [16].

The second class is the class of non-linear *convection-diffusion problems* described by the equation

$$\begin{aligned} \frac{\partial}{\partial t}u - \sum_{k=1}^{n} \frac{\partial}{\partial x_k}\left(\epsilon(u)\frac{\partial}{\partial x_k}u\right) + \boldsymbol{b}^T \underline{\nabla} u &= f && \text{in } Q \\ u(\boldsymbol{x}, t) &= u_c(t) && \text{at } \Gamma_c \\ u(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}) && \text{at } \Gamma_1 . \end{aligned} \qquad (1.2.3)$$

In this case

- the scalar solution $u \in C^0(\overline{Q}) \cap \{C^2(\Omega) \times C^1((0,\infty))\}$.
- the diffusion tensor $\epsilon(u)$, which is given by

$$0 < \epsilon(u) = \epsilon(x, t, |\underline{\nabla} u|^2) < \infty ,$$

  can be non-linear and time-dependent.
- the absolute value of the *flow field* $\boldsymbol{b} = \boldsymbol{b}(x, t)$ is bounded above in Q.

The classes of differential equations defined above have a space-domain which does not vary with time. However, the theory to be presented is not restricted to such cases. It allows for space-domains varying with time, but in order to simplify notations it will be assumed that the boundary of the domain does not change with time. Also, without loss of generality, in order to simplify the presentation of the global time-space finite element method, only problems with Dirichlet boundary conditions will be studied. In the presence of Neumann boundary conditions at – parts of – the cylinder Γ_c, the theory can be presented similarly.

In order to show that the solutions of the problems above are stable let H be a reflexive Banach space and consider an *evolution equation* of the form

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{u} + G(\boldsymbol{u}, t) = 0 \quad t > 0 \\ \boldsymbol{u}(0) = \boldsymbol{u}_0 \end{aligned} \tag{1.2.4}$$

where $\boldsymbol{u}_0 \in H$, and the restriction to time t of the solution $\boldsymbol{u}$, $\boldsymbol{u}(t) \in H$ for all $t > 0$. Here, by definition, the functionals $\frac{\mathrm{d}}{\mathrm{d}t}\cdot$, $G(\cdot, t)$ are mappings $H \mapsto H'$, where H' denotes the dual space of H. As H is reflexive, there exists a Hilbert space L such that H is a continuous injection of L which is a continuous injection of H'. Let in L $(\cdot, \cdot)$ and $\|\cdot\|$ denote the inner product resp. associated norm, and let $\langle \cdot, \cdot \rangle$ be the duality pairing on $L' \times L$. Clearly, this formulation includes the special case where G in (1.2.4) is a complex vectorial function of ordinary differential equations.

In order to show the stability of the solutions of G, the following definitions are introduced. First, the function $G(\cdot, t)$ is called an *unbounded*

functional if

$$\|G(\boldsymbol{u},t)\| \to \infty \quad \text{for} \quad \|\boldsymbol{u}\| \to \infty$$

for every time $t > 0$. Further, the functional G is said to be *monotone* if there exists a nonnegative continuous scalar function $\rho\colon [0,\infty) \mapsto \mathbb{R}$ such that

$$\langle G(\boldsymbol{u},t) - G(\boldsymbol{v},t), \boldsymbol{u}-\boldsymbol{v}\rangle \geq \rho(t)\|\boldsymbol{u}-\boldsymbol{v}\|^2 \quad \forall_{\boldsymbol{u},\boldsymbol{v}\in H}\forall_{t>0}. \tag{1.2.5}$$

and G is called *strongly monotone* or *dissipative* in the case where in addition

$$\rho(t) \geq \rho_0 > 0 \quad \forall_{t>0}.$$

Finally, G is called *conservative* (such as is the case for first order hyperbolic systems) if

$$\langle G(\boldsymbol{u},t) - G(\boldsymbol{v},t), \boldsymbol{u}-\boldsymbol{v}\rangle = 0 \quad \forall_{\boldsymbol{u},\boldsymbol{v}\in H}\forall_{t>0}. \tag{1.2.6}$$

It can be seen (see [16]) that Ladyzhenskayas equation (1.2.1) under the appropriate assumptions corresponds to an evolution equation with an unbounded monotone operator. The same is valid for (1.2.3) under the appropriate conditions on the diffusion tensor and flow field.

Now consider the stability properties of the functional G. If $\boldsymbol{u}$ and $\boldsymbol{v}$ are two solutions of (1.2.4) for different initial values, then

$$\tfrac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{u}-\boldsymbol{v},\boldsymbol{u}-\boldsymbol{v}) + \langle G(\boldsymbol{u},t) - G(\boldsymbol{v},t), \boldsymbol{u}-\boldsymbol{v}\rangle = 0 \quad \forall_{t>0},$$

that is, by (1.2.5)

$$\frac{\mathrm{d}}{\mathrm{d}t}(\|\boldsymbol{u}-\boldsymbol{v}\|^2) + 2\rho(t)\|\boldsymbol{u}-\boldsymbol{v}\|^2 \leq 0 \quad \forall_{t>0}$$

which leads in turn to

$$\|\boldsymbol{u}(t)-\boldsymbol{v}(t)\| \leq \exp\left(-\int_0^t \rho(s)\,ds\right)\|\boldsymbol{u}(0)-\boldsymbol{v}(0)\| \quad \forall_{t>0} \tag{1.2.7}$$

since the solution of $\dot{x}_t + 2\rho(t)x = c(t)$ for $x(0) \in [0, \infty)$ and $c(t) \leq 0$ is given by

$$0 < x(t) = \left[x(0) + \int_0^t c(s)e^{\alpha(s)}\, ds\right] e^{-\alpha(t)}$$
$$\leq x(0)e^{-\alpha(t)}$$

where $\alpha(s) = 2\int_0^s \rho(\varsigma)d\varsigma$. Hence, (1.2.4) is stable for monotone operators. If F is strongly monotone then this equation reduces to

$$\|u(t) - v(t)\| \leq e^{-\rho_0 t}\|u(0) - v(0)\| \quad \forall_{t>0}\,, \tag{1.2.8}$$

so $\|(u-v)(t)\| \mapsto 0$ as $t \mapsto \infty$, which means that (1.2.4) is *asymptotically stable*. Finally, if G is conservative, then

$$\|u(t) - v(t)\| = \|u(0) - v(0)\| \quad \forall_{t>0}$$

according to equation (1.2.7).

1.3 A variational formulation

In this section the variational formulation principle will be introduced. To this end some basic Sobolev space theory is presented and the gradient functional and its directional derivatives are defined. After an example of a strictly monotone functional G the global time-space variational formulation for this example is presented.

To simplify the introduction of the variational formulation, assume that the problem under investigation concerns a scalar problem, i.e., $u = u$. Let for an open, bounded and connected set $\Omega \subset \mathbb{R}^n$, $L^2(\Omega)$ be the space of Lebesgue square integrable functions over Ω with associated inner product resp. norm given by

$$(u, v) = \int_\Omega uv\, dx \quad \text{and} \quad \|u\| = (u, u)^{1/2},$$

using the *Lebesgue integration*. Let the Sobolev space $H^p(\Omega)$ be the closure in $\|\cdot\|$ norm of the set of $C^p(\overline{\Omega})$ (see e.g. [27]). As $H^p(\Omega)$ is the closure of a set of functions which are continuously differentiable,

associated to each function $u \in \mathrm{H}^p(\Omega)$ exists a so-called *trace function*, the restriction of u to the boundary of the set $\partial\Omega$.

Most partial differential equations involve *Dirichlet boundary conditions*, prescribing the solutions trace at – part of – the boundary of the domain. Therefore, let the Dirichlet boundary be denoted by $\Gamma_D \subset \partial\Omega$ and define the associated set of functions

$$\mathrm{H}^p_f(\Omega) = \{u \in \mathrm{H}^p(\Omega): u = f \text{ at } \Gamma_D\} \tag{1.3.1}$$

for a given trace f. By definition let $\Gamma_N = \partial\Omega - \Gamma_D$ be that part of the boundary where *Neumann boundary conditions*, involving the first derivates of the solution, are prescribed.

The classical variational formulation solution method to obtain a solution of a differential equation can be seen as a method to find a minimum of an *energy functional* g: $\mathrm{H}^1_\gamma(\Omega) \mapsto \mathbb{R}$ for a given trace γ. As an example look at the energy functional

$$g(u) = \tfrac{1}{2} \int_\Omega |\underline{\nabla} u|^2 \, d\boldsymbol{x}$$

defined for all $u \in \mathrm{H}^1_\gamma(\Omega)$. In order to find a minimum $u \in \mathrm{H}^1_\gamma(\Omega)$ of g look at the *Gateaux directional derivative* for all $v \in \mathrm{H}^1_0(\Omega)$ defined by

$$\begin{aligned} \frac{\partial}{\partial v} g(u) &= \lim_{\varsigma \to 0} \frac{1}{\varsigma} \{g(u + \varsigma v) - g(u)\} \\ &= \langle g'(u), v \rangle \end{aligned} \tag{1.3.2}$$

where the derivative $g'(u)$ is by definition required to be a linear functional on the space $\mathrm{H}^1_0(\Omega)$ for all u, and where the derivative value of $g'(u)$ in the direction of v is denoted with the use of the *duality pairing* $\langle \cdot, \cdot \rangle$: $[\mathrm{H}^1(\Omega)]' \times \mathrm{H}^1(\Omega) \mapsto \mathbb{R}$. If the limit in (1.3.2) exists uniformly in v then the functional g is said to be *Fréchet differentiable*.

Clearly, the functional g attains a minimum in $u \in \mathrm{H}^1_\gamma(\Omega)$ if the directional derivatives in all directions $v \in \mathrm{H}^1_0(\Omega)$ are equal to zero, where with the use of (1.3.2)

$$\langle G(u), v \rangle = \int_\Omega \underline{\nabla} u^T \underline{\nabla} v \, d\boldsymbol{x} \tag{1.3.3}$$

for all $v \in \mathrm{H}_0^1(\Omega)$, where $G = g'$ is called the *gradient* of g. Analogous to the definition of the directional derivative of g, the derivative of G is defined by

$$\begin{aligned}\langle \frac{\partial}{\partial w} G(u), v\rangle &= \lim_{\varsigma \to 0} \frac{1}{\varsigma} \langle G(u + \varsigma w) - G(u), v\rangle \\ &= \langle G'(u)w, v\rangle \,.\end{aligned} \tag{1.3.4}$$

In this case, where G' is the second directional derivative of the energy functional g, it is called the *Hessian matrix*. However, one can prove that there does not exist an energy functional corresponding to a 'gradient' functional $G(u)$ if its derivative $\langle G'(u)w, v\rangle$ is not symmetric in the arguments v and w, an example of nonsymmetry being

$$\langle G'(u)w, v\rangle = \int_\Omega \underline{\nabla} w^T \underline{\nabla} v + \boldsymbol{b}^T \underline{\nabla} w v \; d\boldsymbol{x}$$

for a non-zero flow field $\boldsymbol{b}$. In this case the functional G' is called the *Jacobian matrix*, i.e., the first derivative of G, but in the sequel G will be called gradient for the sake of simplicity, even if there exists no related energy function.

Now finding the minimum of g is related to finding the solution of: Find a function $u \in \mathrm{H}_\gamma^1(\Omega)$ such that

$$\langle G(u), v\rangle = 0 \quad \forall_{v \in \mathrm{H}_0^1(\Omega)} \,. \tag{1.3.5}$$

A solution $u \in \mathrm{H}_\gamma^1(\Omega) \cap \mathrm{C}^2(\overline{\Omega})$ of this problem is also the solution of the related partial differential equation

$$\begin{aligned} -\triangle u &= 0 \text{ in } \Omega \\ u &= \gamma \text{ at } \Gamma_D = \partial\Omega \end{aligned} \tag{1.3.6}$$

since, using a *Green-Stokes* partial integration lemma (see e.g. [7]), one has

$$\int_\Omega \underline{\nabla} u^T \underline{\nabla} v \; d\boldsymbol{x} = \int_\Omega -v \triangle u \; d\boldsymbol{x}$$

for all $v \in H_0^1(\Omega)$. A classical solution $u \in C^2(\overline{\Omega})$ of equation (1.3.6) is also a solution of (1.3.5), which is called the *Galerkin variational formulation* of (1.3.6). Note that there is less regularity required of a solution satisfying (1.3.5) than for a solution of (1.3.6).

Now recall equation (1.2.4), and let $H = H^1(\Omega)$, $L = L^2(\Omega)$ where the associated norm, inner product and duality pairing are those related to $L^2(\Omega)$. Before introducing the global time-space variational formulation consider the example where $\Omega \subset \mathbb{R}^n$ and the Laplace functional $G(\cdot, t)$: $H \mapsto H'$ in (1.2.4) for all functions $u \in H$ is given by

$$\langle G(u,t), v\rangle = \int_\Omega \underline{\nabla} u^T \underline{\nabla} v \, dx \quad \forall_{v\in H} \quad \forall_{t>0}.$$

This functional is strongly monotone since for solutions u and v of equation (1.2.4) and for $w = u - v$, with the use of a Green-Stokes partial integration lemma

$$\begin{aligned}\langle G(u,t) - G(v,t), u - v\rangle &= \int_\Omega \underline{\nabla} w^T \underline{\nabla} w \, dx \\ &= \int_\Omega |\underline{\nabla} w|^2 \, dx - \oint_\Gamma w \underline{\nabla} w^T \boldsymbol{n} \, ds \\ &= \int_\Omega |\underline{\nabla} w|^2 \, dx + \oint_\Gamma w^2 \, ds \\ &\geq c \int_\Omega w^2 \, dx \quad \forall_{t>0}\end{aligned}$$

for some positive scalar c, independent of u, v and $t > 0$. Here, boundary integrals can be deleted and added since for solutions u, v clearly $u - v = 0$ at $\partial\Omega$. The inequality above holds in general, for all functions $w \in H$, see e.g. [27], page 20. Since this inequality will be used quite frequently in the sequel, in order to understand it, consider

Lemma 1.3.1 *For a domain $\Omega \subset \mathbb{R}^n$, $n \geq 2$, with smooth enough boundary,*

$$\int_\Omega u^2 \, dx \leq \oint_{\partial\Omega} u^2 x^T \boldsymbol{n} \, ds + c^2 \int_\Omega |\underline{\nabla}_x u|^2 \, dx$$

for all functions $u \in H^1(\Omega)$, *where* $c^2 = \sup\{|x|^2 : x \in \Omega\}/(n-1)$.
Proof. For a given function $u \in C^2(\overline{\Omega})$ by partial integration one gets

$$\begin{aligned}\int_\Omega u^2\, dx &= \oint_{\partial\Omega} u^2 x \cdot n_x\, ds - \int_\Omega x(u^2)_x\, dx \\ &= \oint_{\partial\Omega} u^2 x \cdot n_x\, ds - \int_\Omega 2xuu_x\, dx \\ &\le \oint_{\partial\Omega} u^2 x \cdot n_x\, ds + \int_\Omega (ux/c)^2 + (cu_x)^2\, dx\end{aligned}$$

for arbitrary positive scalar c, where $n_x \in \mathbb{R}$ denotes the one-dimensional x-component of the *unit outward normal* vector. This leads to

$$\int_\Omega u^2(1-(x/c)^2)\, dx \le \oint_{\partial\Omega} u^2 x \cdot n_x\, ds + c^2 \int_\Omega (u_x)^2\, dx$$

whence the application of this procedure to every space dimension n yields

$$\int_\Omega u^2(n-(|x|/c)^2)\, dx \le \oint_{\partial\Omega} u^2 x^T n\, ds + c^2 \int_\Omega |\nabla_x u|^2\, dx\,.$$

Setting $c^2 = \sup\{|x|^2 : x \in \Omega\}/(n-1)$ gives for all $x \in \Omega$

$$n - (|x|/c)^2 \ge n - \frac{|x|^2}{\sup\{|x|^2 : x \in \Omega\}} \cdot (n-1) \ge 1\,,$$

and exploiting the fact that $H^1(\Omega)$ is the closure of $C^2(\overline{\Omega})$ under the L^2 norm leads to the desired result for $n \ge 2$. □

Note that lemma 1.3.1 is only valid for $n \ge 2$; for the simple one-dimensional case see [27]. With the use of lemma 1.3.1 now G is clearly strictly monotone since $w = 0$ at the boundary $\partial\Omega$.

A *global time-space variational formulation* associated with the example functional G is now derived in the following way. Define for a given positive t_J a *computational domain* $\Omega \times (0, t_J]$, for the sake of simplicity denoted by $Q \subset \mathbb{R}^{n+1}$ for the remainder of this paragraph.

Then, for a solution $u \in C^2(\overline{\Omega}) \times C^1((0,\infty))$ of equation (1.2.4) and smooth enough function v satisfying homogeneous Dirichlet boundary conditions on $\Gamma_D = \Gamma_1 \cup \Gamma_c$, i.e., $v = 0$ at Γ_D, one has

$$
\begin{aligned}
& u_t - \triangle u = 0 \qquad \forall_{x\in\Omega, t\in(0,\infty)} \Rightarrow \\
& \int_\Omega \left[u_t(\boldsymbol{x},t) - \triangle u(\boldsymbol{x},t)\right] v \; d\boldsymbol{x} = 0 \qquad \forall_{t\in(0,\infty)} \Rightarrow \\
& \int_0^{t_J} \int_\Omega \left[u_t(\boldsymbol{x},t) - \triangle u(\boldsymbol{x},t)\right] v \; d\boldsymbol{x}dt = 0 \qquad \Rightarrow \\
& \int_Q -uv_t + \underline{\nabla} u^T \underline{\nabla} v \; d\boldsymbol{x}dt + \int_{\partial Q} v \underline{\nabla} u^T \boldsymbol{n}_x \, ds + \int_{\partial Q} uv\boldsymbol{n}_t \, ds = 0 \Rightarrow \\
& \int_Q -uv_t + \underline{\nabla} u^T \underline{\nabla} v \; d\boldsymbol{x}dt + \int_\Omega uv(\boldsymbol{x}, t_J) \; d\boldsymbol{x} = 0
\end{aligned}
$$

where $\boldsymbol{n}_x$ is the n-dimensional space component of the *unit outward normal* vector of the surface ∂Q of Q and $\boldsymbol{n}_t$ is the one-dimensional time component of this vector. Taking the derivatives into account, this latter formula is well defined for all functions $u \in C^0(\overline{Q}) \cap \{H^1_\gamma(\Omega) \times L^2((0,t_J])\}$ and all functions $v \in H^1_0(Q)$. Here the latter space is defined substituting Q for Ω in (1.3.1) and, unlike usual, $H^p_\gamma(\Omega) \times L^q((0,t_J])$ is the set of functions u on $\Omega \times (0,t_J]$ which have trace γ, are p times generalized partial differentiable in the directions of the domain Ω (see e.g. [7]), and for which the Lebesgue integral of $u^q + |\underline{\nabla}_x u|^p$ over $\Omega \times (0,t_J]$ exists. Therefore, in this case, the global time-space Petrov-Galerkin variational formulation of (1.2.4) will be given by: Find $u \in C^0(\overline{Q}) \cap \{H^1_\gamma(\Omega) \times L^2((0,t_J])\}$ such that

$$
\begin{aligned}
\langle F(u), v\rangle = & \int_Q -uv_t + \underline{\nabla} u^T \underline{\nabla} v \; d\boldsymbol{x}dt + \int_\Omega uv(\boldsymbol{x}, t_J) \; d\boldsymbol{x} \\
& = 0 \quad \forall_{v\in H^1_0(Q)} .
\end{aligned}
\tag{1.3.7}
$$

The formulation is said to be of the *Petrov Galerkin variational formulation* type as the space of possible solution functions u, called *trial functions*, is not equal to the space of approximating *test functions* v.

Note that a variational formulation requires less regularity, i.e., smoothness of derivates, of possible solutions in general. However, a classical

solution of (1.2.4) is also a solution of (1.3.7). Therefore, if one can prove that there exists only one solution to this latter equation and if one can prove that there exists solutions of (1.2.4), then the solution of (1.3.7) will be a solution in the classical sense.

1.4 The time-slabbing solution method

In order to solve equation (1.2.4) the computational domain $\Omega \times (0, t_J] \subset Q$ will be partitioned in *time-slabs*

$$Q_j = \Omega \times (t_{j-1}, t_j] \quad \text{where} \quad 0 = t_0 < t_1 < \ldots < t_J < \infty$$

assuming without loss of generality that $t_j - t_{j-1} = \triangle t$. In order to introduce the *continuous time-slabbing* technique let

- $\hat{\boldsymbol{u}}$ be the exact solution of (1.2.4) on Q for given initial value $\boldsymbol{u}_0$.
- $\hat{\boldsymbol{u}}_{j,h}$ be an approximate solution to the same equation on Q_j on the first time-slab satisfying the initial value $\boldsymbol{u}_{0,h}$, an *interpolant* or spline approximation of $\boldsymbol{u}_0$, but at each following interval its initial value satisfying $\hat{\boldsymbol{u}}_{j,h}(t_{j-1}) = \hat{\boldsymbol{u}}_{j-1,h}(t_{j-1})$.
- $\hat{\boldsymbol{u}}_j$ be the exact solution of (1.2.4) on Q_j satisfying the same initial value as the approximate solution, i.e., $\hat{\boldsymbol{u}}_j(t_{j-1}) = \hat{\boldsymbol{u}}_{j-1,h}(t_{j-1})$ for all $j = 1, 2 \ldots, J$.

Note that $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{u}}_{j,h}$ and $\hat{\boldsymbol{u}}_j$ satisfy the same boundary conditions at ∂Q_j. The functions $\hat{\boldsymbol{u}}_{j,h}$ and $\hat{\boldsymbol{u}}_j$ also satisfy the same initial value condition.

As the approximation method to obtain $\hat{\boldsymbol{u}}_{j,h}$ is of no relevance to the proof of stability, it will be specified later on. Actually, $\hat{\boldsymbol{u}}_h(t_{j-1})$ can even contain types of errors such as numerical quadrature errors and iteration errors due to the use of an iterative solution method and premature stopping, or errors due to the interpolation of the initial value $\hat{\boldsymbol{u}}_{j-1,h}(t_{j-1})$.

The stability analysis for the evolution equation (1.2.4) for strongly monotone operators now follows readily by (1.2.8). Using the definitions above one can see that on an arbitrary time-slab Q_j

$$\begin{aligned} \|\hat{\boldsymbol{u}}(t_j) - \hat{\boldsymbol{u}}_{j,h}(t_j)\| &\leq \|\hat{\boldsymbol{u}}(t_j) - \hat{\boldsymbol{u}}_j(t_j)\| + \|\hat{\boldsymbol{u}}_j(t_j) - \hat{\boldsymbol{u}}_{j,h}(t_j)\| \\ &\leq e^{-\rho_o \triangle t} \|\hat{\boldsymbol{u}}(t_{j-1}) - \hat{\boldsymbol{u}}_j(t_{j-1})\| + \|\boldsymbol{r}(t_j)\|, \end{aligned} \tag{1.4.1}$$

where the *local discretization error* at time t_j is given by $\|r(t_j)\| = \|\hat{u}_j(t_j) - \hat{u}_{j,h}(t_j)\|$. Note that for

$$\begin{cases} \|\hat{u}(t_{j-1}) - \hat{u}_j(t_{j-1})\| = \|u_0 - u_{0,h}\|, & j = 1 \\ \|\hat{u}(t_{j-1}) - \hat{u}_j(t_{j-1})\| = \|\hat{u}(t_{j-1}) - \hat{u}_{j-1,h}(t_{j-1})\|, & j \in \{2, \ldots, J\} \end{cases}$$

by definition. Repeated use of the above in combination with (1.4.1) shows that the *global discretization error* can be estimated by

$$\begin{aligned} \|\hat{u}(t_J) - \hat{u}_{J,h}(t_J)\| &\leq e^{-J\rho_o \Delta t}\|u_0 - u_{0,h}\| + \sum_{j=1}^{J} e^{-(J-j)\rho_o \Delta t}\|r(t_j)\| \\ &\leq e^{-J\rho_o \Delta t}\|u_0 - u_{0,h}\| + \frac{1 - e^{-J\rho_o \Delta t}}{1 - e^{-\rho_o \Delta t}} \max_{1 \leq j \leq J}\|r(t_j)\| \\ &\leq e^{-J\rho_o \Delta t}\|u_0 - u_{0,h}\| + \frac{1}{1 - e^{-\rho_o \Delta t}} \max_{1 \leq j \leq J}\|r(t_j)\| \end{aligned}$$

for all $\rho_o \Delta t \in \mathbb{R}$ large enough, since

$$\sum_{s=1}^{J} x^{J-s} = \frac{1 - x^J}{1 - x}$$

for all $1 \neq x \in \mathbb{R}$ and $J \in \mathbb{N}$. This leads to the following elementary but important result.

Theorem 1.4.1 *Let $\hat{u}_{j,h}$ be the approximate finite element solution of the evolution equation (1.2.4) on Q_j for initial value $\hat{u}_{j-1,h}(t_{j-1})$ and let $\hat{u}_j$ be the exact solution of this equation on Q_j for the same initial value. Then, if G in (1.2.4) is strongly monotone,*

$$\begin{aligned} \|\hat{u}(t_J) - \hat{u}_{J,h}(t_J)\| \leq\, & e^{-J\rho_o \Delta t}\|u_0 - u_{0,h}\| + \\ & (1 - e^{-\rho_o \Delta t})^{-1} \max_{1 \leq j \leq J}\|r(t_j)\| \end{aligned} \tag{1.4.2}$$

for all $J \in \mathbb{N}$, where $r(t_j) = \hat{u}_j(t_j) - \hat{u}_{j,h}(t_j)$ for all $j \in \mathbb{N}$ and $u_{0,h}$ is an interpolant or spline approximation of u_0. □

Theorem 1.4.1 shows that one has control of the error, even on an unbounded number of time intervals, and that the error at any time line or plane $t = t_j$ is bounded by the maximal local error $\|\boldsymbol{r}(t_j)\|$ times the constant $(1 - e^{-\rho_o \Delta t})^{-1}$. Since the scalar Δt can be taken fixed this constant can also be regarded fixed. (see also [19], page 160).

Similarly, for monotone and conservative operators, the error bound

$$\|\hat{\boldsymbol{u}}(t_J) - \hat{\boldsymbol{u}}_{J,h}(t_J)\| \leq \|\boldsymbol{u}_0 - \boldsymbol{u}_{0,h}\| + J \max_{1 \leq j \leq J} \|\boldsymbol{r}(t_j)\| \quad \forall_{J \in \mathbf{N}} \quad (1.4.3)$$

is readily derived. Here the errors will increase at most linearly with the number of time-slabs.

1.5 The finite element discretization

In order to introduce the finite element discretization technique let the domain Q be covered completely by a set of simplices $\mathcal{Q}$, where in two dimensions simplex stands for triangle. This set is called *grid*, and its elements have mutually empty intersection (for more details see chapter 5). This grid will be used to construct the finite element basis functions, which must be of high order if one wants to obtain a high order discretization error estimate.

First, for the sake of simplicity, consider the definition of piecewise linear basis functions. Let the n-dimensional *reference simplex* $\hat{\triangle} \subset \mathbb{R}^n$ be defined by

$$\hat{\triangle} = \{\hat{\boldsymbol{x}} \in \mathbb{R}^n : \hat{\boldsymbol{x}} > 0 \wedge \sum_{j=1}^{n} \hat{x}_j < 1\},$$

having $n + 1$ vertices $\hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{n+1} = \boldsymbol{0}, (1, 0, \ldots, 0), \ldots, (0, \ldots, 0, 1)$, where for abbreviation $\hat{\boldsymbol{x}}_r = (\hat{x}_{1,r}, \ldots, \hat{x}_{n,r})$, and $\hat{\boldsymbol{x}} > 0$ means $\hat{x}_j > 0$ for all r. It is well-known that for each simplex $\triangle \in \mathcal{Q}$ there exists an *affine transformation*

$$\boldsymbol{x} = F_{\triangle}\hat{\boldsymbol{x}} + \boldsymbol{x}_1 = \begin{bmatrix} x_{1,2} - x_{1,1} & \cdots & x_{1,n+1} - x_{1,1} \\ \vdots & \ddots & \vdots \\ x_{n,2} - x_{n,1} & \cdots & x_{n,n+1} - x_{n,1} \end{bmatrix} \hat{\boldsymbol{x}} + \boldsymbol{x}_1$$

mapping the reference simplex $\hat{\triangle}$ onto $\triangle$ such that $\hat{x}_1, \ldots, \hat{x}_{n+1}$ are mapped onto $x_1, \ldots, x_{n+1}$ (see [7]). In order to distinguish vertices of different simplices easily, the vertices x_r of a simplex are denoted by $x_r^{(\triangle)}$ when needed.

On the reference simplex, corresponding to its vertices, $n+1$ linear basis functions are defined by

$$\hat{\varphi}_1(\hat{x}) = 1 - \sum_{r=1}^{n} \hat{x}_r$$

$$\hat{\varphi}_r(\hat{x}) = \hat{x}_{r-1}\,, \quad r = 2, \ldots, n+1\,.$$

Further, let for each simplex in the grid the linear *approximation polynomial* $\varphi_r^{(\triangle)}$ corresponding to $x_r^{(\triangle)}$ be given by the relation $\varphi_r^{(\triangle)}(x) = \hat{\varphi}_r(\hat{x})$ via the affine transformation. If $\{x_i\}_1^N$ denotes the set of all vertices, then for given number i there will be several simplices $\triangle$ and $r = \{1, \ldots, n+1\}$ such that $x_r^{(\triangle)} = x_i$. Corresponding to each vertex $x = x_i$ a piecewise linear *finite element basis function* φ_i is defined elementwise as follows. If for given $\triangle$ there exists an r such that $x_r^{(\triangle)} = x_i$, then $\varphi_i \equiv \varphi_r^{(\triangle)}$ on $\triangle$, elsewise $\varphi_i = 0$ on $\triangle$.

Higher order finite element basis functions are constructed analogously to the procedure above using a set of polynomials on the n-dimensional reference simplex and the affine transformation. For the general case see [7], section 5.1, or [29] where approximation polynomials of arbitrary degree p are used. One can also see chapter 7 of this thesis, which employs finite element basis functions of order two.

Suppose that grid refinement will be used to obtain a suitable grid, as will be demonstrated in chapter 5, leading to a sequence of grids $\mathcal{Q}_j^{(0)} \subset \mathcal{Q}_j^{(1)} \subset \cdots \subset \mathcal{Q}_j^{(k)}$. As in equation (5.9.1), the span of the finite element basis functions on each subsequent grid $\mathcal{Q}_j^{(k)}$ is denoted by $\mathcal{H}(\mathcal{Q}_j^{(k)})$ and

$$\mathcal{H}_\gamma(\mathcal{Q}_j^{(k)}) = \{u \in \mathcal{H}(\mathcal{Q}_j^{(k)}) \colon u(x) = \gamma(x) \text{ at } \mathcal{V}(\mathcal{Q}_j^{(k)}) \cap \Gamma_D\}$$

where $\mathcal{V}(\mathcal{Q}_j^{(k)})$ stands for the set of vertices of all simplices in a grid $\mathcal{Q}_j^{(k)}$. A basis of finite element basis functions obtained as above is called a

standard nodal finite element basis. For future use also define the *grid size parameter* $h_j^{(k)}$, being the minimum simplex diameter of all the simplices in $\mathcal{Q}_j^{(k)}$. Depending on the circumstances, $h_j^{(k)}$ can be denoted by $h^{(k)}$ or just by h.

Let the computational domain be denoted by Q and the grid on Q by $\mathcal{Q}$ and consider the example (1.3.7). In this case for $\Gamma_D = \Gamma_1 \cup \Gamma_c$ the Galerkin *global time-space finite element variational formulation* will be given by: Find an approximate or *discrete solution* $\hat{u}_h \in \mathcal{H}_\gamma(\mathcal{Q})$ such that

$$\begin{aligned}\langle F(\hat{u}_h), v\rangle &= \int_Q -\hat{u}_h v_t + \underline{\nabla}\hat{u}_h^T \underline{\nabla} v \; dx dt + \int_\Omega uv(x, t_J)\; dx \\ &= 0 \quad \forall_{v\in\mathcal{H}_0(\mathcal{Q})}\,.\end{aligned} \tag{1.5.1}$$

Concerning the existence of such a discrete solution of a Galerkin finite element variational formulation as above, see Ciarlet [14]. Now, requiring the *continuous solution* to be equal to $\lim_{h\downarrow 0} \hat{u}_h \in \mathrm{H}^1_\gamma(Q)$, the *global time-space variational formulation* used in all chapters is: Find $\hat{u} \in \mathrm{H}^1_\gamma(Q)$ such that

$$\begin{aligned}\langle F(\hat{u}), v\rangle &= \int_Q -\hat{u} v_t + \underline{\nabla}\hat{u}^T \underline{\nabla} v \; dx dt + \int_\Omega uv(x, t_J)\; dx \\ &= 0 \quad \forall_{v\in\mathrm{H}^1_0(Q)}\,.\end{aligned} \tag{1.5.2}$$

This formulation requires more regularity of the solution than equation (1.3.7) as the solution is supposed to be generalized differentiable in time, but for many cases of physical interest one can show that existing solutions have indeed the required smoothness.

Before considering the general case in the next section, consider the local error on a time-slab $\Omega \times (t_{j-1}, t_j]$ for the sake of simplicity denoted by Q. If, for given initial value, $\hat{u}$ is defined to be the exact solution on this time-slab and if $\hat{u}_h$ is the finite element approximate solution of the variational formulation for the same initial value, then the local error is given by

$$r(t_j) = \hat{u}(t_j) - \hat{u}_h(t_j)\,.$$

In order to estimate this local error define

- $\hat{u}_I$, the interpolation of $\hat{u}$ in $\mathcal{H}(Q)$,
- $\theta := \hat{u} - \hat{u}_h \in \mathrm{H}^1(Q)$, the discretization error,
- $\eta := \hat{u} - \hat{u}_I \in \mathrm{H}^1(Q)$, the interpolation error, and
- $\varphi := \hat{u}_h - \hat{u}_I \in \mathcal{H}_0(Q)$, the interpolation minus the discretization error.

To simplify the presentation, assume that the Dirichlet boundary conditions and the initial value function are homogeneous, whence

$$\begin{aligned}
&\tfrac{1}{2}\int_\Omega \varphi^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x} + \int_Q |\underline{\nabla}\varphi|^2\, d\boldsymbol{x}dt = \\
&\langle F(\hat{u}_h) - F(\hat{u}_I), \varphi\rangle = \\
&\langle F(\hat{u}) - F(\hat{u}_I), \varphi\rangle = \\
&\int_Q -\eta\varphi_t + \underline{\nabla}\eta^T\underline{\nabla}\varphi\, d\boldsymbol{x}dt + \int_\Omega \eta\varphi(\boldsymbol{x}, t_j)\, d\boldsymbol{x} = \\
&\int_Q \varphi\eta_t + \underline{\nabla}\eta^T\underline{\nabla}\varphi\, d\boldsymbol{x}dt
\end{aligned}$$

with the use of the relation

$$\int_Q vv_t\, d\boldsymbol{x}dt = \int_\Omega v^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x} - \int_Q vv_t\, d\boldsymbol{x}dt \quad \forall_{v\in \mathrm{H}_0^1(Q)}.$$

The last term of the former equation can be estimated above by

$$\{\int_Q \eta_t^2 + |\underline{\nabla}\eta|^2\, d\boldsymbol{x}dt\}^{1/2} \cdot \{\int_Q \varphi^2 + |\underline{\nabla}\varphi|^2\, d\boldsymbol{x}dt\}^{1/2}$$

with the use of lemma 1.3.1 and the Cauchy-Schwarz relation between inner product and associated norm. At its turn, the last factor in this bound can be estimated above by

$$\{\int_\Omega \varphi^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x}\}^{1/2} + \{\int_Q |\underline{\nabla}\varphi|^2\, d\boldsymbol{x}dt\}^{1/2}.$$

This clearly implies that there exists a positive scalar c such that

$$\int_\Omega \varphi^2(x,t_j)\,dx \le \int_\Omega \varphi^2(x,t_j)\,dx + \int_Q |\underline{\nabla}\varphi|^2\,dxdt$$
$$\le c\int_Q \eta_t^2 + |\underline{\nabla}\eta|^2\,dxdt\,.$$

Hence, using the *Nečas trace inequality* (see [27], page 84), stating that there exists a positive scalar c such that

$$\oint_{\partial Q} u^2\,ds \le c\int_Q u^2 + \left|\underline{\nabla}_{x,t}u\right|^2\,dxdt \tag{1.5.3}$$

for all functions $u \in \mathrm{H}^1(Q)$ where $\underline{\nabla}_{x,t}$ denotes the gradient in space and time, leads to

$$\|r(t_j)\|^2 = \int_\Omega \theta^2(x,t_j)\,dx$$
$$\le 2\int_\Omega \varphi^2(x,t_j)\,dx + 2\int_\Omega \eta^2(x,t_j)\,dx$$
$$\le c\int_Q \eta_t^2 + |\underline{\nabla}\eta|^2\,dxdt$$

for some scalar $c > 0$. Using the *classical interpolation error estimate* (2.5.2) presented in chapter 2, the bound

$$\|r(t_j)\| \le Dh^s\|\hat{u}\|_{s+1}$$

is obtained for all $0 \le p \le s$, where p is the degree of the finite element approximation polynomials used for the discrete solution and interpolant. As this is valid for all time-slabs and the scalar D only depends on the geometry of the domains Q_j, this implies that for time-slabs of constant width $t_j - t_{j-1} = \triangle t$ the global error is bounded above in time, according to theorem 1.4.1. Note that $\|\cdot\|_{s+1}$ stands for the Sobolev $s+1$ norm over Q, including all space and time derivatives up to order $s+1$.

1.6 Newton method and local discretization error estimates

Here the local errors $r(t_j)$ for the Galerkin variational formulation are considered for the problem (1.2.3) on the domain Q_j, treated as a convection-diffusion problem. The diffusion tensor is taken to be that of equation (1.2.2) for a scalar function, i.e., $\epsilon(u) = \epsilon_0 + \epsilon_1 |\underline{\nabla} u|^{q-2}$. Let $\Gamma_D = \Gamma_1 \cup \Gamma_c$ and Γ_3 be defined as in sections 1.2 and 1.3. Then problem (1.2.3) can be reformulated as follows. Find $u \in H^1_\gamma(Q_j) \subset H^1(\Omega) \times L^2((t_{j-1}, t_j])$ such that for all $v \in H^1_0(Q_j)$

$$\int_{t_{j-1}}^{t_j} \int_\Omega \left[u_t v + \epsilon(u) \underline{\nabla} u^T \underline{\nabla} v + \boldsymbol{b}^T \underline{\nabla} u v - f v \right] d\boldsymbol{x} dt = 0, \qquad (1.6.1)$$

where $H^1_g(Q_j)$ for a given trace g is defined using equation (1.3.1). Note that on Γ_1 and on Γ_3, $\underline{\nabla}_x u^T \boldsymbol{n}_x = 0$, where $\boldsymbol{n}_x$ is the unit outward normal vector in the space directions. For the simplicity of notation let $Q = Q_j$ in the sequel.

One can now follow either of two lines of analysis. Using the finite element method directly on the non-linear problem (1.6.1) and the theory for error estimates for bounded monotone operators (see [18]), we can derive discretization error estimates, similar to the derivation for linear coercive problems. The resulting equation is a non-linear algebraic equation which must be solved by some iterative method, such as the Newton method.

Alternatively, Newton's method, or some Newton like method on (1.6.1), can be used to get a sequence of linear differential problems. Each of these problems is solved by a finite element method, but not necessarily for identical grids. This latter option opens up the possibility of combining the algebraic solution method with the discretization method or more precisely, with the adaptive grid refinement method. Therefore this second approach has definite practical advantages over the first one, where one must work on a fixed finite element grid.

Now let $F'(u,t)$ be the Gateaux directional derivative of $F(u,t)$ and consider the variational formulation of the Newton sequence $u^{(l)}$,

$$\langle F'(u^{(l)}, t)(u^{(l+1)} - u^{(l)}), v \rangle = -\tau^{(l)} \langle F(u^{(l)}, t), v \rangle \quad \forall_{v \in H^1_0(Q)} \qquad (1.6.2)$$

where, for a given functional $G(u,t)$, $F(u,t) = u_t + G(u,t)$ and the duality pairing and corresponding norm are defined by

$$\langle u, v\rangle = \int_{t_{j-1}}^{t_j} \int_\Omega uv \; dxdt, \quad |||v||| = \langle v, v\rangle^{1/2}.$$

For problem (1.6.1) one has

$$\langle F(u,t), v\rangle = \int_{t_{j-1}}^{t_j} \int_\Omega \left[u_t v + \epsilon(u)\underline{\nabla} u^T \underline{\nabla} v + \boldsymbol{b}^T \underline{\nabla} uv - fv\right] \; dxdt,$$

$$\langle F'(u,t)w, v\rangle = \int_{t_{j-1}}^{t_j} \int_\Omega \left[w_t v + \epsilon(u)\underline{\nabla} w^T \underline{\nabla} v + \boldsymbol{b}^T \underline{\nabla} wv\right] \; dxdt +$$

$$\tfrac{1}{2}\epsilon_1(q-2) \int_{t_{j-1}}^{t_j} \int_\Omega |\underline{\nabla} u|^{q-4} \underline{\nabla} u^T \underline{\nabla} w \underline{\nabla} u^T \underline{\nabla} v \; dxdt,$$

according to equation (1.3.4). The parameter $\tau^{(l)}$ in (1.6.2) is a damping parameter if $\tau^{(l)} \leq 1$. For $\tau^{(l)} = 1$, the method reduces to the standard Newton method. As has been shown in [5] and [15], the damping parameter can be chosen such that the Newton method converges for any initial approximation. In the type of problems here considered, it turns out that the method converges with $\tau^{(l)} = 1$, because already the initial function can be chosen fairly accurately. As has also been shown in [5] and [15], there is a further important improvement of the Newton method. This is based on the observation that it is inefficient to solve (1.6.2) very accurately for the first iterations, because the corresponding approximation $u^{(l+1)}$ is only an approximation of the final solution $\lim_{l\to\infty} u^{(l)}$ and usually not a very accurate one. Therefore, instead, a solution $u^{(l+1)}$ is computed such that for all $l = 0, 1, \ldots$

$$|||F'(u^{(l)}, t)(u^{(l+1)} - u^{(l)}) + \tau^{(l)} F(u^{(l)}, t)||| \leq \varepsilon^{(l)} |||F(u^{(l)}, t)|||, \quad (1.6.3)$$

where $\varepsilon^{(l)}$ is a sequence of positive numbers converging monotonically to zero, or to some predetermined accuracy ε, i.e., $\lim_{l\to\infty} \varepsilon^{(l)} = \varepsilon$.

Therefore, as one iterates further, this sequence forces the linearized equation in the Newton method to be solved increasingly more accurately. This method has been called the damped and inexact Newton method (DIN), see[5] and [15] for an analysis of its convergence.

What makes this method additionally attractive in the present context is that it can be combined in a natural way with the finite element discretization of the problem. At the initial stage one can start with a somewhat coarse grid and refine this from one Newton iteration to the next, for instance by adding grid points and hierarchical basis functions thereon. Note however, that one has to use the exact nodal values of the boundary function u_c for the new (added) points on the boundary. Also (1.6.3) needs to be modified to

$$|||F'(u_I^{(l)},t)(u^{(l+1)}-u_I^{(l)})+\tau^{(l)}F(u_I^{(l)},t)||| \leq \varepsilon^{(l)}|||F(u_I^{(l)},t)||| \quad (1.6.4)$$

for all $l = 0, 1, \ldots$, where $u_I^{(l)}$ is the interpolant of the function $u^{(l)}$ from the possibly coarser grid $\mathcal{Q}^{(l)}$ onto the space $\mathcal{Q}^{(l+1)}$. The corresponding finite element spaces are nested (see section 1.5), $\mathcal{H}(\mathcal{Q}^{(l)}) \subset \mathcal{H}(\mathcal{Q}^{(l+1)})$ and are subspaces of $H^1(Q)$. A natural choice of the forcing sequence $\varepsilon^{(l)}$ is then a power of $h^{(l)}$.

At every Newton step one needs to solve a linear convection diffusion problem. The variational finite element formulation of (1.6.4), assuming homogeneous Dirichlet boundary conditions $\gamma \equiv 0$, takes the following form. Find a correction $\chi = u^{(l+1)} - u_I^{(l)} \in \mathcal{H}_0(\mathcal{Q}^{(l+1)})$ such that

$$\langle F'(u_I^{(l)},t)\chi, v\rangle = -\langle F(u_I^{(l)},t), v\rangle \quad (1.6.5)$$

for all $v \in \mathcal{H}_0(\mathcal{Q}^{(l+1)})$. To simplify the presentation it will now be assumed that the spaces $\mathcal{H}_0(\mathcal{Q}^{(l)}) = \mathcal{H}_0$, i.e., are identical for all iterations, and that $h^{(l)} = h$. For the derivation of a discretization error estimate, first note that for all $u \in H_0^1(Q)$

$$\langle u_t + \boldsymbol{b}^T \underline{\nabla} u, u\rangle = \langle u_t - \nabla\cdot(\boldsymbol{b}u), u\rangle \text{ and}$$

$$\langle u_t + \boldsymbol{b}^T \underline{\nabla} u, u\rangle = -\tfrac{1}{2}\langle u\nabla\cdot\boldsymbol{b}, u\rangle + \tfrac{1}{2}\int_{\Gamma_3} u^2(\boldsymbol{x},t_j)\, d\boldsymbol{x}\,.$$

Analogous to the proofs in [9] and [10], using the condition $\nabla\cdot\boldsymbol{b} \leq 0$, the boundedness of $|\boldsymbol{b}|$, and exploiting the definition of $\epsilon(u)$, it is possible to

show (see chapters 2 – 4) that there exists a positive scalar c such that

$$\epsilon_0 |||\underline{\nabla}(\chi_h - \chi_I)|||^2 + \tfrac{1}{2}\int_{\Gamma_3} (\chi_h - \chi_I)^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x} \leq c|||\underline{\nabla}_{x,t}(\chi - \chi_I)|||^2 ,$$

where the Cauchy-Schwarz inequality and the arithmetic-geometric mean inequality have been used. Here χ_I denotes the interpolant of χ in $\mathcal{H}_0$. Now, assume that for some $0 < s \leq k$, where k is the degree of the *approximation polynomials* used in the finite elements, the following relation holds

$$|||\underline{\nabla}_{x,t}(\chi - \chi_I)||| \leq ch^s |||\chi|||_{s+1} .$$

Here $|||\cdot|||_{s+1}$ denotes a Sobolev norm of order $s+1$ on Q (i.e., containing $L^2(Q)$ norms of space derivatives of χ up to order $s+1$) and c a positive generic constant (in general not the same at different occurrences). Now, the last two equations show that

$$\epsilon_0 \|\underline{\nabla}(\chi_h - \chi_I)\|_{L^2(Q)} + \left\{ \tfrac{1}{2}\int_{\Gamma_3} (\chi_h - \chi_I)^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x} \right\}^{1/2} \leq ch^s |||\chi|||_{s+1} . \tag{1.6.6}$$

Analogous to the discretization error estimate at the end of section 1.5, one can demonstrate that in particular the errors at the interface $t = t_j$ given by the expression $\left\{ \int_{\Gamma_3} (\chi - \chi_h)^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x} \right\}^{1/2}$, as well as the finite element errors in the gradient in Q, are bounded by the right-hand side of (1.6.6), which has approximation order s. Hence, if u is sufficiently smooth and the finite element space of a sufficiently high order, an arbitrary order of approximation can be obtained.

Note that the parabolic type operators considered here have a smoothing property in the respect that for increasing t, the solution gets smoother. For any fixed t, one can have infinitely differentiable solutions if the source function f and the boundary conditions allow this, even if the initial function is non-smooth. However, if one wants accurate approximations in the whole domain, then the initial function needs to be smoothed prior to applying the method. At every interface between two time-slabs a spline approximation of $\hat{u}_h$ can be taken, using only the nodal values but defining proper derivatives at the node points by taking (weighted) averages of the derivatives of $\hat{u}_h$ at the node points. In this way the spline approximation at time $t = t_j$ will be (the trace of)

a function of a higher order Sobolev space, such as $H^p(\Omega)$, $p \geq 2$, and the discretization error estimates above are valid with maximal s, i.e., $s = k$.

Incidently, since the finite element method on the time-slab will only see the node values of the approximating spline (which equal the node values of $\hat{u}_h$), the approximating spline function need not be computed. It will only be needed if a different set of node points at the interface is used.

The order of the discretization errors at the interfaces can be improved. For singular perturbation type problems, where $\epsilon(u) \ll 1$ in (1.2.3), the streamline upwind finite element method of [20] can be used to this end. As has been shown in [4], [8], [22] and [26] for the case that $\epsilon_1 = 0$ in (1.6.1), this can improve the interior error estimate in the L^2 norm by half an order (so it will still be suboptimal by such an amount). Similarly the discretization error at the interfaces will be improved. Now let $\epsilon_1 = 0$ in (1.6.1) and consider instead of (1.6.5) the variational formulation

$$\delta \hat{b}(u, v) + \hat{a}(u, v) = -\langle F(u_I^{(l)}, t), v + \delta v_{\hat{b}} \rangle \tag{1.6.7}$$

where $v_{\hat{b}} = v_t + \boldsymbol{b}^T \underline{\nabla} v$ is the streamline directional derivative in time-space. Furthermore, $\hat{a}(u, v)$ is the bilinear form defined by the right-hand side of (1.6.5) and

$$\hat{b}(u, v) = \sum_{\Delta \in \mathcal{Q}} \left[\int_\Delta \nabla \cdot (-\hat{a} \underline{\nabla} u) v_{\hat{b}} \, d\boldsymbol{x}dt + \int_\Delta u_{\hat{b}} v_{\hat{b}} \, d\boldsymbol{x}dt \right] \tag{1.6.8}$$

where $\mathcal{Q}$ is the set of finite elements (triangles) in the time-space domain. In relation (1.6.7), δ is a positive parameter which needs to be chosen such that $\epsilon_0 \delta \leq O(h^2)$ in order to guarantee the *coerciveness* of $\delta \hat{b} + \hat{a}$ on the finite element space $\mathcal{V}$ (for further details, see [8] and the next two chapters). The standard Galerkin method can now be applied to (1.6.7) and it is readily seen that when $\epsilon_0 = O(h)$ this method with $\delta = O(h)$ improves the order of the approximation as compared to the case (1.6.5) where $\delta = 0$. This result is stated in the following theorem.

Theorem 1.6.1 *Consider the streamline finite element method (1.6.7) to compute a finite element correction χ in (1.6.5) on a time-slab Q.*

Then, if $\epsilon_0\delta \leq O(h^2)$, *the discretization error at the upper boundary* Γ_3 *satisfies*

$$\left\{\int_{\Gamma_3}(\chi-\chi_h)^2\,d\boldsymbol{x}\right\}^{1/2} \leq ch^s\left[\delta^{1/2}+\epsilon_0^{1/2}+\delta^{-1/2}h\right]|||\chi|||_{s+1}\,,\ s\leq k$$

where k *is the degree of the piecewise polynomial finite elements used in* $\mathcal{V}$. □

The above estimate is proved in [8] and does not use any duality argument or elliptic regularity. It shows that, if $\delta = O(h)$, implying $\epsilon_0 \leq O(h)$, then

$$\left\{\int_{\Gamma_3}(\chi-\chi_h)^2\,d\boldsymbol{x}\right\}^{1/2} \leq ch^{s+1/2}|||\chi|||_{s+1}\,,\quad s\leq k\,.$$

This is an optimal order estimate and applies to singular perturbation type problems where ϵ_0 is very small. Note that the error estimate in theorem 1.6.1 is valid even for problems where $\epsilon_0 = 0$, i.e., for first order hyperbolic problems.

The standard procedure for regular problems, where an elliptic regularity estimate for the adjoint operator $F^\star$ is valid, is to use a duality argument to prove an optimal order estimate in $L^2(Q_j)$. Then a trace inequality could be used to improve the discretization error estimate at the interfaces in (1.6.6) by half an order. However, as the problem is not of second order in the time variable, such an elliptic regularity is not valid when the problem is solved in time-slabs.

Finally, note that if an adaptive grid refinement method is used in order to control the local errors of each time-slab, one also has an automatic time-step control method. Namely, by increasing the required number of degrees of freedom within a time-slab as the current time-step decreases, and vice versa. Proceeding this way, the degrees of freedom will remain approximately constant within each time-slab. For regular, uniform elements in time (see figure 1.1), with one layer of piecewise linear finite element basis functions defined thereon, the time-slabbing method is equivalent to the Crank-Nicolson method.

As has been shown in [4], where the time-slabbing method with one or several layers of uniform elements is used (see fig. 1.1) for the interpolation error for linear problems, a cancellation effect occurs in

the term $\|\|\|(\hat{u} - \hat{u}_h)_t + \boldsymbol{b}^T \underline{\nabla}(\hat{u} - \hat{u}_h)\|\|\|_Q$ for polynomial basis functions of odd degree, with the effect that the error in $\|\hat{u} - \hat{u}_h\|_{L^2(Q)}$ and $\left\{\int_{\Gamma_3}(\hat{u} - \hat{u}_h)^2(\boldsymbol{x}, t_j)\, d\boldsymbol{x}\right\}^{1/2}$ becomes of higher order than for irregular elements. If either $\epsilon_0 \leq O(h^2)$ or $\epsilon_0 = O(1)$ it can be shown that the error is of optimal order but this estimate requires then one order higher regularity of the solution, $\hat{u} \in H^{k+2}(Q) \cap H_0^1(Q)$. Therefore, the time-slabbing method will not give lower order of errors than standard time-stepping methods, but it gives the additional freedom of using irregular elements in time and space to approximate solutions with layers better.

Finally, it can be seen that if a 'spectral' finite element method based on Legendre polynomials or certain combinations of such polynomials in the time variable is used, the time-slabbing method becomes equivalent to an implicit Runge-Kutta method for solving the corresponding semidiscrete system of ordinary differential equations.

1.7 Conclusions

It has been shown that time-slabbing is an efficient technique to get higher order approximations and is applicable for many types of problems. It does not suffer from the error reduction phenomenon which is found for certain high order time-stepping methods.

Furthermore, shocks and layers can be resolved more easily using this technique. An additional advantage not discussed in the present chapter is that the method for a single time-slab can also be used for a backward heat equation, for instance when one uses only one – big – time-slab $(0, t_J]$, assuming that the given data on the line $t = t_J$ corresponds to an essentially layer-free solution. Then the irreversibility of the process, normally showing up in an exponential increase of numerical errors, will be noticed much less than for a standard time-stepping method.

Acknowledgements

This work was supported in part by the Florida State University Supercomputer Computations Research

Institute which is partially funded by the U.S. Department of Energy through contract no. DE-FC05-85ER250000.

1.8 References

[1] Adams R.A., *Sobolev Spaces,* Academic Press, 1975

[2] Axelsson O., *An algebraic framework for multilevel methods,* internal report 8820 (October 1988), Department of Mathematics, University of Nijmegen, The Netherlands

[3] Axelsson O., *Error estimates over infinite intervals of some discretizations of evolution equations,* BIT, 24(1984), 413-424

[4] Axelsson O., *Finite element methods for convection-diffusion problems,* **in** Numerical Treatment of Differential Equations, (Strehmel K. ed.) Leipzig: Teubner 1988 (Teubner-Texte zur Mathematik; Bd. 104), 171-182 [Proceedings of the Fourth Seminar "Numdiff-4", Halle, 1987]

[5] Axelsson O., *On global convergence of iterative methods,* **in** Iterative Solution of Nonlinear Systems of Equations, 1-19 LNM#953, (Ansorge R., Meis Th. and Törnig W. eds.), Springer Verlag, 1982

[6] Axelsson O., *Stability and error estimates valid for infinite time, for strongly monotone and infinitely stiff evolution equations,* **in** Equadiff 6 (Vosmansky J. and Zlamal M. eds.), J.E. Purkyne University, Brno, 1985

[7] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[8] Axelsson O., Eijkhout V., Polman B. and Vassilevski P., *Iterative solution of singular pertubation 2^{nd} order boundary value problems by use of incomplete block-factorization methods,* BIT, 29(1989), 867-889

[9] Axelsson O. and Maubach J., *A time-space finite element discretization technique for the calculation of the electromagnetic field in ferromagnetic materials,* Journal for Numerical Methods in Engineering, 29(1989), 2085-2111

[10] Axelsson O. and Maubach J., *A time space finite element method for nonlinear convection diffusion problems,* **in** Notes on Numerical Fluid Mechanics, (Hackbush W. and Rannacher R. eds.)

Vol. 30, 6-23, Vieweg, Braunschweig, 1990 [Proceedings of the Fifth GAMM-Seminar, Kiel, West Germany 1989]

[11] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods I,* Numerische Mathematik, 56(1989), 157-177

[12] Burrage K. and Hundsdorfer W.H., *The order of B-convergence of algebraically stable Runge-Kutta methods,* BIT, 27(1987), 62-71

[13] Burrage K., Hundsdorfer W.H. and Verwer J.G, *A study of B-convergence of Runge-Kutta methods,* Computing, 36(1986), 17-34

[14] Ciarlet P.G., *The Finite Element Method for Elliptic Problems,* North-Holland Publ., Amsterdam, 1978

[15] Dembo R.S., Eisenstat S.C. and Steihaug T., *Inexact Newton methods,* SIAM Journal on Numerical Analysis, 19(1982), 400-408

[16] Du Q. and Gunzberger M.D., *Finite-element approximations of Ladyzhenskaya model for stationary incompressible viscous flow,* SIAM Journal on Numerical Analysis, 27(1990), 1-19

[17] Frank R., Schneid J. and Ueberhuber C.W., *The concept of B-convergence,* SIAM Journal on Numerical Analysis, 18(1981), 753-780

[18] Girault V. and Raviart P.A., *Finite Element Methods for Navier-Stokes Equations,* Springer Verlag, Berlin, 1986

[19] Hairer E., Nørsett S.P. and Wanner G., *Solving Ordinary Differential Equations,* Springer Verlag, 1987

[20] Hughes T.J. and Brooks A., *A multi-dimensional upwind scheme with no crosswind diffusion,* **in** AMD, 34(1979), Finite element methods for convection dominated flows (Hughes T.J. ed.), ASME, New York

[21] Hughes T.J., Mallet M. and Mizukami A., *A new finite element formulation for computational fluid dynamics, IV. A discontinuity capturing operator for multi-dimensional advective-diffusive systems,* Computer Methods in Applied Mechanics and Engineering, 58(1986), 329-336

[22] Johnson C. and Nävert U., *An analysis of some finite element methods for advection-diffusion problems,* **in** Analytical and Numerical Approaches to Asymptotic Problems in Analysis (Axels-

son O., Frank L.S. and van der Sluis A. eds.), 99-116, North--Holland , 1981

[23] Johnson C. and Szepessy A., *Shock-capturing streamline diffusion finite element methods for nonlinear conservation laws,* **in** Recent Developments in Computational Fluid Mechanics (Tezdyar T.E. and Hughes T.J. eds.) vol. 95, AMD, The Americal Society of Mechanical Engineers, 1988

[24] Kraaijevanger J., *B-convergence of the implicit midpoint rule and trapezoidal rule,* BIT, 25(1985), 652-666

[25] Marion M. and Temam R., *Nonlinear Galerkin methods,* SIAM Journal on Numerical Analysis 26(1989), 1139-1157

[26] Nävert U., *A finite element method for convection diffusion problems,* Ph.D. thesis, Department of Computer Sciences Chalmers University of Technology, Göteborg , Sweden, 1982

[27] Nečas J., *Les Méthodes Directes en Théorie des Equations Elliptiques,* Masson, Paris, 1967

[28] Prothero A. and Robinson A., *The stability and accuracy of one-step methods,* Mathematics of Computation, 28(1974), 145-162

[29] Zienkiewicz O., *The Finite Element Method in Engineering Science,* 3^{rd} edition, Mc Graw-Hill, New York, 1977

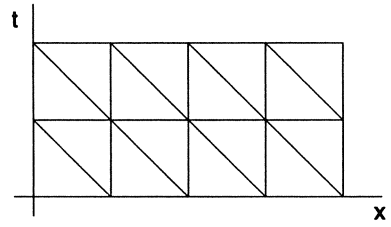


Fig. 1.1 Grid of uniform elements in space-time.

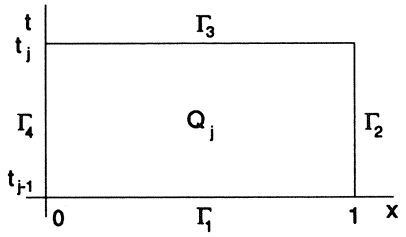


Fig. 1.2 Time-slab for space domain (0,1).

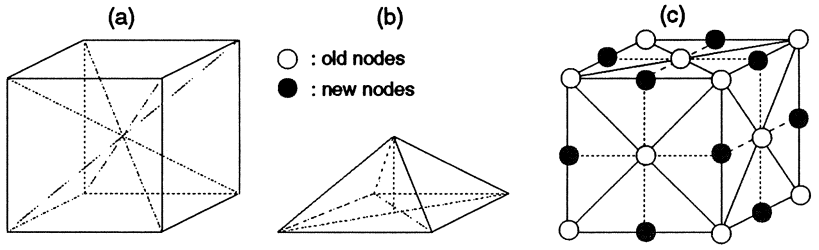


Fig. 1.3 (a) Divide each small cube into 6 pyramids from the center. (b) Divide each pyramid into 4 tetrahedrons. (c) One level of the hierarchical extension type.

2 The weighted Galerkin global finite element method

Part of: Axelsson O. and Maubach J., A time-space finite element method for non-linear convection diffusion problems, Notes on Numerical Fluid Mechanics, (Hackbush W. and Rannacher R., eds.) Vol. 30, 6-23, Vieweg 1990, [Proceedings of the Fifth GAMM-Seminar, Kiel, 1989]. The section on the variational finite element solution method contains an added discussion on the determination of a suitable weighing function. Printed with permission of the General Editor of the Vieweg Verlag Series Notes on Numerical Fluid Mechanics; part of this chapter is copyrighted by Vieweg Verlag.

Abstract

Time-stepping methods for parabolic problems require a careful choice of the stepsize for stability and accuracy. Even if a stable implicit time-stepping method is used, one might be forced to choose very small time-steps in order to get a sufficient accuracy, if the solution has steep gradients, even if these occur only in a narrow part of the domain. Therefore the solution of the corresponding algebraic systems can be expensive since many time-steps have to be taken. The same considerations are valid for explicit time-stepping methods.

A discretization technique using finite element approximations in time and space simultaneously for a relatively large time-period, called *time-slab*, is presented in this chapter. This technique may be repeatedly applied to obtain further parts of the solution in subsequent time intervals. It will be shown that, with the method proposed, the solution can be computed cheaply, even if it has steep gradients, and that stability is automatically guaranteed. For the solution of the non-linear algebraic equations on each time-slab fast iterative methods can be used.

Key words: Time-stepping, Time-space finite elements, Nonlinear parabolic differential equations, Convection diffusion, Grid refinement
AMS(MOS) subject classifications: 65F10, 65M20, 65N30, 65N50

2.1 Introduction

The method most frequently used for the numerical integration of parabolic differential equations is the method of lines, where one first uses a discretization of space derivatives by finite differences or finite elements and then uses some time-stepping method for the solution of the resulting system of ordinary differential equations. Such methods are, at least conceptually, easy to perform. However, they can be expensive if steep gradients occur in the solution, when stability must be controlled. Also the global error control can be troublesome.

This chapter considers a simultaneous discretization of space and time variables for a one-dimensional parabolic equation on a relatively long time interval, called time-slab. The discretization is repeated or adjusted for following time-slabs using continuous finite element approximations. In this method the efficiency of finite elements is utilized by choosing a finite element grid in the time-space domain such that the grid has been adjusted to steep gradients of the solution, both with respect to the space and the time variables. In this way, all the difficulties with the classical approach are solved: stability, discretization error estimates and global error control are automatically satisfied. Such a method has been discussed previously in [1] and [5]. The related boundary value techniques and global time integrations for systems of ordinary differential equations have been discussed in several papers, see [13] and the references quoted therein. In [19] a time-space method with discontinuous elements in time has been used, which is based on methods in [20], [21] and [24].

In the present chapter, an equation which describes the electromagnetic vector potential in ferromagnetic materials is taken to demonstrate the proposed discretization and solution method, but the techniques provided may also be applied to other types of parabolic equations including equations in many space variables.

The remainder of the chapter is organized as follows. In section 2.2 the necessary information about the parabolic differential equation and the parameters involved is given. Then in section 2.3 this non-linear parabolic equation is reformulated as a two-dimensional boundary value problem. After this, section 2.4 and section 2.5 consider the solution method and discretization error estimates for the problem. Section 2.6 concludes with a discussion of the method. Numerical results can be found in section 3.5.

2.2 Evolution equations

Let $\Omega \subset \mathbb{R}$ be an open interval and consider the following non-linear evolution equation defined on the time-space interval $Q := \Omega \times (0, \infty)$

$$\begin{array}{rll} -(\epsilon u_x(x,t))_x + b u_x(x,t) + \sigma u_t(x,t) & = f(x,t) & (x,t) \in Q \\ u(x,0) & = u_0(x) & x \in \Omega \\ u(0,t) & = l(t) & t \in (0,\infty) \\ u(1,t) & = r(t) & t \in (0,\infty) \end{array} \tag{2.2.1}$$

where the diffusion ϵ and the flow velocity functions b, σ satisfy $\epsilon \equiv \epsilon(u_x^2(x,t))$ resp. $b = b(x,t)$ and $\sigma = \sigma(x,t)$. Here f is a source function and u_0 some $L^2(\Omega)$ integrable function, l and r are the left- and right-hand side square integrable boundary conditions. In addition, assume that $\sigma \geq \sigma_0 > 0$, $b, \sigma \in C^1(\overline{Q})$. The latter space stands for the vector space of continuously partial differentiable functions in Q, which – including the partial derivatives – can be extended continuously to the boundary of the domain Q. Further, let for the simplicity of notation $b_x + \sigma_t \leq 0$ and define $\epsilon' \equiv \frac{\partial}{\partial \zeta}\epsilon(\zeta)$ where $\zeta = u_x^2$, and assume that $\epsilon > 0$ and $\epsilon' \geq 0$. The theory to be presented in this chapter will be valid for *inhomogeneous boundary conditions* of the Dirichlet type.

In the classical way of solving (2.2.1), one first discretizes the space-variable x, e.g., with the use of a finite element method (see [25]). Then the calculation of the solution of the system of ordinary differential equations obtained is done with the use of a time-stepping method. One of the disadvantages of this approach is that, in order to get a good approximation of the solution $\hat{u}(x,t)$ for large values of $t > 0$, many

small time-steps must be used if the solution has steep gradients, even if these occur only in a small part of the space interval. Furthermore, for explicit time-stepping methods, the stepsize must be chosen to satisfy an Euler method type stability criterion (however, as shown in [17] and [23] there exist methods with extended stability regions which can partly alleviate this difficulty). Also the local discretization errors made with the use of a time-stepping method have to be monitored closely to control the global errors made in time. Here a method is considered where a finite element grid is chosen for the time-space domain. This method has no such disadvantages.

2.3 Two-dimensional time-slab formulation

In order to compute the solution of (2.2.1) a computational domain $\Omega \times (0, t_J] \subset Q$ is partitioned into a number of *time-slabs* $Q_j = \Omega \times (t_{j-1}, t_j]$ for $0 = t_0 < t_1 < \ldots < t_J < \infty$, assuming without loss of generality $t_j - t_{j-1} = \Delta t$ for all j (see fig. 1.2). The time-slabs Q_j have lower and upper boundaries denoted by Γ_1 resp. Γ_3, and left and right boundaries Γ_4 resp. Γ_2. The number of such time-slabs is finite, independent of the choice of the grid parameter, associated with the finite elements. For the first time-slab Q_1 an initial value u_0 on Γ_1 has to be given, but for each following time-slab Q_{j+1} the solution at Γ_3 of Q_j will be taken to provide a Dirichlet boundary condition at Γ_1.

With this approach problem (2.2.1) can be rewritten into:

$$\begin{array}{rll} -(Eu_x(x,t))_x + \hat{\boldsymbol{b}}^T \underline{\nabla} u(x,t) & = f(x,t) & \text{in } Q_j \\ u(x,0) & = u_0(x) & \text{at } \Gamma_1 \\ u(0,t) & = l(t) & \text{on } \Gamma_4 \\ u(1,t) & = r(t) & \text{on } \Gamma_2 \end{array} \qquad (2.3.1)$$

on each time-slab with

- tensor $E = \epsilon(u_x^2)$ and flow field $\hat{\boldsymbol{b}} \equiv [b(x,t), \sigma(x,t)]^T$, where it is assumed that $\sigma \geq \sigma_0 > 0$ in order to preserve the parabolic nature of the equation
- square integrable functions l and r, prescribing the Dirichlet boundary conditions on the left respectively right boundary
- the divergence operator $\nabla \cdot$ and gradient operator $\underline{\nabla}$ defined on the two-dimensional (x,t) space and

- square integrable source function f at Q_j and initial value function u_0 at the boundary Γ_1.

Note that in the remainder of this chapter the gradient and divergence operators will act on the two-dimensional time-space space. Further, there is no need to impose any boundary condition at the boundary Γ_3, because for all possible trial functions u and all test functions v the corresponding boundary integral

$$\oint_{\Gamma_3} vEu_x n_x \, dx = \oint_{\Gamma_3} v\epsilon(u_x^2)u_x \cdot 0 \, dx = 0,$$

where n_x is the x component of $\boldsymbol{n}$, the *unit outward normal* of the boundary $\partial\overline{Q}_j$. At this boundary the solution $\hat{u}$ of (2.3.1) and $\hat{u}_x$ are initially unknown.

An advantage of the formulation (2.3.1) is that it permits the use of small sized elements inside layers, for an accurate time-space finite element discretization. Such layers can arise for $b > 0$ along the boundary Γ_2, for $b < 0$ along the boundary Γ_4 and in the interior along a shockwave, typically starting at the south-west corner, if $u_0(0) \neq l(0)$ and $b > 0$. In other parts of the time-space domain one can use much larger elements thus reducing the number of degrees of freedom considerably compared to a classical time-stepping method.

As will be shown, the computation of the finite element solution on each time-slab can be done efficiently. The solution $\hat{u}$ of (2.3.1) will be calculated by a non-linear iterative method, which implies that an initial solution u_0 must be provided. If there is any a priori knowledge about the solution then this information can be used to construct a proper initial grid for each time-slab.

2.4 Variational finite element solution method

Consider the variational formulation of the non-linear two-dimensional problem (2.3.1) for a certain time-slab $Q := Q_j$. Let the space $\mathrm{H}^1(Q)$ be the Sobolev space of order 1 on Q and define the boundary function γ at $\Gamma_D := \Gamma_1 \cup \Gamma_{2,4}$ by $\gamma := (u_0, r, l)$, i.e., $\gamma \equiv u_0$ at Γ_1, $\gamma \equiv r(t)$ at Γ_2 and $\gamma \equiv l(t)$ at Γ_4. To simplify the analysis, assume that there

exists an extension of γ to Q in $H^1(Q)$, which excludes the occurrence of interior layers due to discontinuous boundary data. Define the test and trial spaces by $H_0^1(Q) := \{v \in H^1(Q): v \equiv 0 \text{ at } \Gamma_D\}$ resp. $H_\gamma^1(Q) := \{u \in H^1(Q): u \equiv \gamma \text{ at } \Gamma_D\}$, both in the sense of traces.

In order to improve if possible standard Galerkin finite element discretization error estimates, consider the use of a suitable weighted Petrov-Galerkin method and therefore the determination of a suitable weighing function g (see e.g. [16], section 3.4, page 90). The weighted Petrov Galerkin variational formulation is

$$\langle F(u), v\rangle = 0 \quad \forall_{v\in H_0^1(Q)}, u \in H_\gamma^1(Q) \tag{2.4.1}$$

where the gradient F is given by

$$\langle F(u), v\rangle = \int_Q \left[-(Eu_x)_x + \hat{\boldsymbol{b}}^T \underline{\nabla} u - f\right] vg \; dx\,dt \tag{2.4.2}$$

for all $u, v \in H^1(Q)$. In order to determine g, without being restrictive, it is assumed that all boundary conditions are homogeneous Dirichlet conditions. Then, with the use of the Green-Stokes formula, (2.4.1) turns out to be equivalent to

$$\int_Q Eu_x v_x g + \boldsymbol{L}(g)\underline{\nabla} uv \; dx\,dt = \int_Q gfv \; dx\,dt \quad \forall_{v\in H_0^1(Q)} \tag{2.4.3}$$

where $\boldsymbol{L}(g) = \begin{bmatrix} Eg_x \\ 0 \end{bmatrix} + g\hat{\boldsymbol{b}}$ is a functional on $\left[H^1(Q)\right]^n$. As

$$\int_Q \boldsymbol{L}(g)^T \underline{\nabla} vv \; dx\,dt = \tfrac{1}{2} \oint_{\Gamma_3} v^2 \boldsymbol{L}(g)^T \boldsymbol{n} \; dx - \tfrac{1}{2} \int_Q v^2 \nabla\cdot \boldsymbol{L}(g) \; dx\,dt$$

for all $v \in H_0^1(Q)$, the substitution of $u = v$ into (2.4.3) leads to a left-hand side equal to

$$\int_Q gEv_x^2 - \tfrac{1}{2}\nabla\cdot \boldsymbol{L}(g)v^2 \; dx\,dt + \tfrac{1}{2} \oint_{\Gamma_3} v^2 \boldsymbol{L}(g)^T \boldsymbol{n} \; dx\,. \tag{2.4.4}$$

This left-hand side can be estimated below in terms of $\|v\|_1^2$ only if $g \geq 0$ satisfies

$$\begin{aligned} &\int_Q -\tfrac{1}{2}\nabla\cdot \boldsymbol{L}(g)v^2 \, dx\,dt \geq \int_Q cv^2 \, dx\,dt \\ &\oint_{\Gamma_3} v^2 \boldsymbol{L}(g)^T \boldsymbol{n} \, dx \geq 0 \end{aligned} \tag{2.4.5}$$

for all Lebesgue square integrable functions v and some $c \geq 0$.

Assuming that $-\nabla\cdot\hat{\boldsymbol{b}} = 0$ and $c = 0$, the first inequality (2.4.5) is satisfied, if $-\frac{1}{2}\nabla\cdot\boldsymbol{L}(g) = c$. Note that the latter equation resembles the adjoint equation of the original unweighted variational formulation (formulation (2.4.2) with $g \equiv 1$) since

$$-\nabla\cdot\boldsymbol{L}(g) = -(Eg_x)_x - \hat{\boldsymbol{b}}^T \underline{\nabla} g - g\nabla\cdot\hat{\boldsymbol{b}}\,.$$

This implies that the determination of a suitable weight function can be as difficult as solving the original variational problem.

However, for the global time-space case with tensor E and flow field $\hat{\boldsymbol{b}}$ as defined before, the functional $\boldsymbol{L}(g)$ is given by

$$\boldsymbol{L}(g) = \begin{bmatrix} \epsilon(u_x^2)g_x \\ 0 \end{bmatrix} + g\hat{\boldsymbol{b}}\,.$$

One can easily verify that the choice $g(x,t) = \mathrm{e}^{-\alpha(t-t_j)}$ satisfies (2.4.5) for a given time-slab Q_j and fixed $\alpha \geq 0$ as $\nabla\cdot\boldsymbol{L}(g) = g\nabla\cdot b \leq 0$. To simplify future proofs only this function will be used for the derivation of error estimates in section 2.5. It suffices to consider the first time-slab $(t_0, t_1) := (0, T)$, thus reducing the weight function to $t \mapsto \mathrm{e}^{-\alpha t}$.

Using the exponential weight function, one finds that, for all $u, v \in$

$\mathrm{H}^1(Q)$, $\langle F(u), v\rangle$ is equal to

$$\int_Q Eu_x(v\mathrm{e}^{-\alpha t})_x + (\hat{\boldsymbol{b}}^T \underline{\nabla} u - f)v\mathrm{e}^{-\alpha t}dx\,dt - \oint_{\partial Q} v\mathrm{e}^{-\alpha t} Eu_x \boldsymbol{n}_x\, ds =$$
$$\int_Q \epsilon u_x \frac{\partial}{\partial x}(v\mathrm{e}^{-\alpha t}) + (\hat{\boldsymbol{b}}^T \underline{\nabla} u - f)v\mathrm{e}^{-\alpha t}dx\,dt - \oint_{\Gamma_{2,4}} v\mathrm{e}^{-\alpha t} Eu_x \boldsymbol{n}_x\, ds =$$
$$\int_Q \epsilon u_x v_x \mathrm{e}^{-\alpha t} + (\hat{\boldsymbol{b}}^T \underline{\nabla} u - f)v\mathrm{e}^{-\alpha t}dx\,dt - \oint_{\Gamma_{2,4}} v\mathrm{e}^{-\alpha t} Eu_x \boldsymbol{n}_x\, ds =$$
$$\int_Q \left[Eu_x v_x + (\hat{\boldsymbol{b}}^T \underline{\nabla} u - f)v\right] \mathrm{e}^{-\alpha t}dx\,dt - \oint_{\Gamma_{2,4}} v\mathrm{e}^{-\alpha t} Eu_x \boldsymbol{n}_x\, ds$$

is valid, because $Eu_x \boldsymbol{n}_x \equiv 0$ at $\Gamma_1 \cup \Gamma_3$ and $\frac{\partial}{\partial x}(v\mathrm{e}^{-\alpha t}) = v_x \mathrm{e}^{-\alpha t}$.

Linearization of this weak formulation with the use of a damped Newton method now leads to a sequence of linear systems and solutions $u^{(k+1)} \in \mathrm{H}^1_\gamma(Q)$

$$\langle F'(u^{(k)})(u^{(k+1)} - u^{(k)}), v\rangle = -\tau^{(k)}\langle F(u^{(k)}), v\rangle \quad \forall_{v \in \mathrm{H}^1_0(Q)}. \tag{2.4.6}$$

Here the Hessian or Jacobian matrix F' is defined as in (1.3.4), by substituting F for G, whence for all $u, v, w \in \mathrm{H}^1(Q)$

$$\langle F'(u)w, v\rangle = \int_Q \left[B(u)w_x v_x + \hat{\boldsymbol{b}}^T \underline{\nabla} wv\right] \mathrm{e}^{-\alpha t}\, dx\,dt - \oint_{\Gamma_{2,4}} v\mathrm{e}^{-\alpha t} B(u)w_x \boldsymbol{n}_x\, ds \tag{2.4.7}$$

where the tensor $B(u)$ (see chapter 3 for the multi-dimensional case) is defined by

$$B(u) = \epsilon(u_x^2) + 2u_x^2 \epsilon'(u_x^2)$$

as can be seen easily using (1.3.4). Further $\tau^{(k)}$ is a positive scalar which is called damping parameter for values less than 1. This scalar can be monitored from step to step in order to achieve convergence, see for instance [3] and [12].

The fact that $u^{(k+1)} - u^{(k)} \in \mathrm{H}^1_0(Q)$, a linear vector space on which the Jacobian matrix will be positive definite (see below), implies

that the damped Newton algorithm defined by (2.4.6) will converge for properly chosen damping parameters $\tau^{(k)}$ (see e.g. [3]).

Note that, due to the convective term $\hat{\boldsymbol{b}}^T \underline{\nabla} wv$ in the integrand of (2.4.7), the Jacobian matrix $F'(u)$ is not symmetric, but because of the special structure of the tensor ϵ and the nonsymmetric term, the technique described in [10] for symmetric problems can be modified easily in order to assemble the gradient and Jacobian matrix cheaply.

Define the Hilbert space $\hat{\mathrm{H}}^1(Q) \supset \mathrm{H}^1(Q)$, the closure of $\mathrm{C}^1(\overline{Q})$ under the weighted norm

$$|||v||| := \left(\int_Q \left[v^2 + v_x^2\right] \mathrm{e}^{-\alpha t}\, dx\, dt\right)^{\frac{1}{2}} \quad \alpha \in \mathbb{R},$$

which is related to a corresponding inner product. Let the norms $\|\cdot\|_{s,\alpha}$ and $|\cdot|_{s,\alpha}$ denote the exponentially weighted Sobolev norm resp. seminorm of order s on $\mathrm{H}^1(Q)$, and let $(\cdot\,,\cdot)_{s,\alpha}$ denote the inner products corresponding to the weighted seminorms. In the case that $\alpha = 0$, the subscript '$,\alpha$' is omitted. Note that $\|\cdot\|_{s,\alpha}$ and $\|\cdot\|_s$ are equivalent norms for all $s \geq 0$ and $\alpha \geq 0$. The norm $|||\cdot|||$ can be seen as a weighted Sobolev 1 measure in space combined with a weighted L^2 measure in time on $\hat{\mathrm{H}}^1(Q)$. With the use of the set of norms introduced, and under some assumptions to be derived on the tensor ϵ and flow field $\hat{\boldsymbol{b}}$, $F'(u)$ will be seen to be uniformly positive definite on $\mathrm{H}_0^1(Q)$, i.e.,

$$\langle F'(u)v, v\rangle \geq c|||v||| > 0 \quad \forall_{v \in \mathrm{H}_0^1(Q)},$$

for some positive scalar c.

In order to see this, first note that

$$\langle F'(u)w, v\rangle = \int_Q \left[Bw_x v_x + \hat{\boldsymbol{b}}^T \underline{\nabla} wv\right] \mathrm{e}^{-\alpha t}\, dx\, dt$$

for all $v \in \mathrm{H}_0^1(Q)$ and all $u, w \in \mathrm{H}^1(Q)$. An analysis of the separate terms in this expression shows that

$$0 < \lambda_{\min} \int_Q v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt \leq \int_Q B(u) v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt \qquad (2.4.8)$$

for all $u, v \in \mathrm{H}^1(Q)$, with

$$\lambda_{\min} := \inf\{\epsilon(\zeta) + 2\zeta\epsilon'(\zeta) : \zeta = u_x^2(x,t),\ (x,t) \in Q\},$$

and that

$$\int_Q \hat{\boldsymbol{b}}^T \underline{\nabla} wv\mathrm{e}^{-\alpha t}\, dx\, dt = \oint_{\partial Q} vw\mathrm{e}^{-\alpha t}\hat{\boldsymbol{b}}^T \boldsymbol{n}\, ds - \int_Q w\nabla\cdot(\hat{\boldsymbol{b}}v\mathrm{e}^{-\alpha t})\, dx\, dt$$

is identical to

$$\begin{aligned}
&\oint_{\Gamma_3} vw\mathrm{e}^{-\alpha t}\hat{\boldsymbol{b}}^T \boldsymbol{n}\, dx - \int_Q w(\nabla\cdot\hat{\boldsymbol{b}}v\mathrm{e}^{-\alpha t} + \hat{\boldsymbol{b}}^T\underline{\nabla}(v\mathrm{e}^{-\alpha t}))\, dx\, dt \\
=&\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} vw\hat{\boldsymbol{b}}^T \boldsymbol{n}\, dx - \\
&\int_Q w\left[\nabla\cdot\hat{\boldsymbol{b}}v\mathrm{e}^{-\alpha t} + v\hat{\boldsymbol{b}}^T\underline{\nabla}\mathrm{e}^{-\alpha t} + \mathrm{e}^{-\alpha t}\hat{\boldsymbol{b}}^T\underline{\nabla}v\right] dx\, dt \\
=&\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} vw\sigma\, dx - \int_Q \hat{\boldsymbol{b}}^T\underline{\nabla}vw\mathrm{e}^{-\alpha t}\, dx\, dt + \\
&\int_Q vw(\alpha\sigma - \nabla\cdot\hat{\boldsymbol{b}})\mathrm{e}^{-\alpha t}\, dx\, dt \quad \forall_{w\in\mathrm{H}^1(Q)}\forall_{v\in\mathrm{H}_0^1(Q)}
\end{aligned}$$

because $v \equiv 0$ at $\Gamma_1 \cup \Gamma_{2,4}$, $\underline{\nabla}\mathrm{e}^{-\alpha t} = [0, -\alpha\mathrm{e}^{-\alpha t}]^T$ and $\hat{\boldsymbol{b}}^T\boldsymbol{n} = \sigma$ at Γ_3. This latter relationship leads to

$$\begin{aligned}
&\int_Q (\hat{\boldsymbol{b}}^T\underline{\nabla}v)v\mathrm{e}^{-\alpha t}\, dx\, dt = \\
&\tfrac{1}{2}\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} v^2\sigma\, dx + \tfrac{1}{2}\int_Q v^2(\alpha\sigma - \nabla\cdot\hat{\boldsymbol{b}})\mathrm{e}^{-\alpha t}\, dx\, dt \geq \qquad (2.4.9) \\
&\tfrac{1}{2}\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} v^2\sigma\, dx + b_{\min}\int_Q v^2\mathrm{e}^{-\alpha t}\, dx\, dt
\end{aligned}$$

for all $v \in \mathrm{H}_0^1(Q)$ where $b_{\min} := \inf\{\frac{1}{2}(\alpha\sigma(x,t) - \nabla\cdot\hat{\boldsymbol{b}}(x,t)) : (x,t) \in Q\}$.

Now (2.4.8) and the above show that the Jacobian matrix satisfies

$$\begin{aligned}\langle F'(u)v, v\rangle \geq & \lambda_{\min} \int_Q v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt + b_{\min} \int_Q v^2 \mathrm{e}^{-\alpha t}\, dx\, dt + \\ & \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} v^2 \sigma\, dx \\ & \geq \min\{\lambda_{\min}, b_{\min}\} \int_Q \left[v^2 + v_x^2\right] \mathrm{e}^{-\alpha t}\, dx\, dt \\ & =: c \cdot |||v||| \quad \forall_{v \in \mathrm{H}_0^1(Q)} \forall_{u \in \mathrm{H}^1(Q)}\,,\end{aligned} \tag{2.4.10}$$

i.e., is uniformly positive definite if $\lambda_{\min}$ and $b_{\min}$ both are positive. For positive $b_{\min} \leq \lambda_{\min}$ this estimate turns out to be uniform in ϵ.

In the situation where $b_{\min} = 0$ note that for piecewise continuous functions v on Q the restriction to a certain time $t \in (t_{j-1}, t_j]$, will also be piecewise continuous on Ω, in particular $v(x,t) \in \mathrm{H}^1(\Omega)$. If Γ is a nontrivial subset of Ω, then, due to a *Friedrichs inequality* (see [26], page 20, or [15]), there exists a positive scalar $\beta > 0$ such that for all functions $v \in \mathrm{H}^1(\Omega)$

$$\int_\Omega v_x^2(x,t)\, dx + \oint_\Gamma v^2(x,t)\, ds \geq \beta \int_\Omega [v^2(x,t) + v_x^2(x,t)]\, dx\,.$$

In the one-dimensional case, where Γ is equal to the set endpoints of the open interval Ω, it is by definition nontrivial. Because v is piecewise continuous on Ω and due to the fact that the space-domain does not vary within time, the expression above can be integrated with respect to the time, with the use of a weight $\mathrm{e}^{-\alpha t}$, leading to

$$\oint_{\Gamma_2 \cup \Gamma_4} v^2 \mathrm{e}^{-\alpha t}\, ds + \int_Q v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt \geq \beta \int_Q \left[v^2 + v_x^2\right] \mathrm{e}^{-\alpha t}\, dx\, dt$$

for all piecewise continuous functions v on Ω. This implies that

$$\left(\int_Q v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt\right)^{\frac{1}{2}} \text{ and } \left(\int_Q \left[v^2 + v_x^2\right] \mathrm{e}^{-\alpha t}\, dx\, dt\right)^{\frac{1}{2}}$$

are equivalent norms on the subspace of piecewise continuous functions of $v \in \mathrm{H}^1(Q)$ with $v \equiv 0$ at $\Gamma_{2,4}$, whence for $b_{\min} = 0$ and such functions

v

$$\begin{aligned}\langle F'(u)v, v\rangle \geq & \lambda_{\min} \int_Q v_x^2 \mathrm{e}^{-\alpha t}\, dx\, dt + \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} v^2 \sigma\, dx \\ \geq & \beta\lambda_{\min} \int_Q \left[v^2 + v_x^2\right] \mathrm{e}^{-\alpha t}\, dx\, dt =: c|||v|||^2\end{aligned} \tag{2.4.11}$$

for all $\alpha \geq 0$. Because $b_{\min} = 0$ this estimate is not bounded uniformly in ϵ (see the definition of $\lambda_{\min}$) but contrary to the previous estimate (2.4.10) it is also valid for $\alpha = b_{\min} = 0$. Hence (2.4.10) will be used mainly for singularly perturbed problems while (2.4.11) will be used for regular problems.

Each linear system (2.4.6) is discretized with the use of two-dimensional triangular finite elements (FE) (maximum height h) with linear basis functions (see e.g. [28], [14], [6] or [22]). Additional upwind, i.e., streamline-upwind diffusion basis functions (SUPG) (see [18] or [4]), is optional. For this latter method the linear basis functions v are replaced by $v+\delta\underline{\nabla}v$ for some scalar $\delta > 0$. As has been shown in [7], for instance, the upwind technique can be very helpful to get a more strongly positive definite system for convection dominated problems and hence increase the rate of convergence of certain generalized preconditioned conjugate gradient iterative methods.

The use of a finite dimensional subspace of $\mathrm{H}^1(Q)$ in (2.4.6) to approximate $\hat{u}$ leads to a sequence of corresponding finite dimensional linear systems of the form $F_h'(\boldsymbol{u}_h^{(k)})(\boldsymbol{u}_h^{(k+1)} - \boldsymbol{u}_h^{(k)}) = -\tau^{(k)} F_h(\boldsymbol{u}_h^{(k)})$, defined as usual in finite element methods, with $\lim_{k\to\infty} \boldsymbol{u}_h^{(k)} := \hat{\boldsymbol{u}}_h$, the discrete solution in vector representation (see e.g. [3]). These linear nonsymmetric finite dimensional systems of equations are solved by iteration with the use of preconditioned linear equation solvers. For the numerical tests GCGLS [2] or CGS [27] are used to this end. These iterative methods and the Newton method can be found in chapter 8.

2.5 Discretization error estimate

In order to study the discretization error on time-slab $Q = Q_j$, consider the introduction of a finite dimensional finite element test function

subspace $\mathcal{H}$ of $H^1(Q)$, based on an underlying (initial) finite element triangulation $\mathcal{Q} = \mathcal{Q}_j$ of the time-slab. Using $\mathcal{H}$, define $\mathcal{H}_0 := H_0^1(Q) \cap \mathcal{H}$, $\mathcal{H}_\gamma := H_\gamma^1(Q) \cap \mathcal{H}$. The function γ, which describes the Dirichlet boundary conditions, is assumed to be of such a type that $\mathcal{H}_\gamma \neq \emptyset$, e.g., if $\mathcal{H}$ is a space of piecewise linear functions, then γ has to be piecewise linear too. Also consider the following definitions.

Definitions

- $\hat{u} \in H_\gamma^1(Q)$, a solution of (2.4.1), i.e., $\langle F(\hat{u}), v\rangle = 0$ for all $v \in H_0^1(Q)$,
- $\hat{u}_h \in \mathcal{H}_\gamma$, a discrete solution satisfying $\langle F(\hat{u}_h), v\rangle = 0$ for all functions $v \in \mathcal{H}_0$,
- $\hat{u}_I$, the interpolation of $\hat{u}$ on $\mathcal{H}$,
- $\theta := \hat{u} - \hat{u}_h \in H^1(Q)$, the discretization error,
- $\eta := \hat{u} - \hat{u}_I \in H^1(Q)$, the interpolation error and
- $\varphi := \hat{u}_h - \hat{u}_I \in \mathcal{H}_0$, the interpolation minus the discretization error.

In order to estimate the discretization error note that $\varphi \equiv 0$ at Γ_D, and assume that $\epsilon, \hat{\boldsymbol{b}}$ with $\sigma \geq \sigma_0 > 0$ and $\alpha \geq 0$ are such that the following four conditions are satisfied

$$\begin{cases} \lambda_{\min} = \inf\{\epsilon(\zeta) + 2\zeta\epsilon'(\zeta) : \zeta = u_x^2(x,t),\ (x,t) \in \Omega\} > 0 \\ \lambda_{\max} = \sup\{\epsilon(\zeta) + 2\zeta\epsilon'(\zeta) : \zeta = u_x^2(x,t),\ (x,t) \in \Omega\} < \infty \\ b_{\min} = \inf\{\frac{1}{2}(\alpha\sigma(x,t) - \nabla\cdot\hat{\boldsymbol{b}}(x,t)) : (x,t) \in \Omega\} > 0 \\ b_{\max} = \sup\{\max\{\left|\hat{\boldsymbol{b}}(x,t)\right|, \sigma(x,t)\} : (x,t) \in \Omega\} < \infty \end{cases} \tag{2.5.1}$$

for all $u \in H^1(Q)$. Then for $b_{\min} > 0$ with the use of (2.4.10)

$$\begin{aligned} \langle F(\hat{u}_h) - F(\hat{u}_I), \varphi\rangle &= \langle \int_0^1 F'(\hat{u}_I + \varsigma\varphi)\varphi\, d\varsigma, \varphi\rangle \\ &= \int_0^1 \int_Q \left[B(\hat{u}_I + \varsigma\varphi)\varphi_x^2 + \hat{\boldsymbol{b}}^T \underline{\nabla}\varphi\varphi\right] e^{-\alpha t} dx\, dt\, d\varsigma \\ &\geq \lambda_{\min} \int_Q \varphi_x^2 e^{-\alpha t}\, dx\, dt + b_{\min} \int_Q \varphi^2 e^{-\alpha t} dx\, dt \end{aligned}$$

$$+\tfrac{1}{2}e^{-\alpha T}\oint_{\Gamma_3}\varphi^2\sigma\,dx$$

$$\geq \min\{\lambda_{\min}, b_{\min}\}|||\varphi|||^2 =: c_1|||\varphi|||^2$$

or for $b_{\min} = 0$ with the use of (2.4.11)

$$\langle F(\hat{u}_h)-F(\hat{u}_I),\varphi\rangle \geq \lambda_{\min}\int_Q \varphi_x^2 e^{-\alpha t}\,dx\,dt + \tfrac{1}{2}e^{-\alpha T}\oint_{\Gamma_3}\varphi^2\sigma\,dx$$

$$\geq \beta\lambda_{\min}\int_Q [\varphi^2+\varphi_x^2]\,e^{-\alpha t}\,dx\,dt =: c_1|||\varphi|||^2.$$

Further

$$\langle F(\hat{u}_h)-F(\hat{u}_I),v\rangle = \langle F(\hat{u})-F(\hat{u}_I),v\rangle \quad \forall_{v\in\mathcal{H}_0}$$

and for all $\alpha \geq 0$

$$\langle F(\hat{u})-F(\hat{u}_I),\varphi\rangle = \langle\int_0^1 F'(\hat{u}_I+\varsigma\eta)\eta\,d\varsigma,\varphi\rangle$$

$$= \int_0^1\int_Q \left[B(\hat{u}_I+\varsigma\eta)\eta_x\varphi_x + \hat{\boldsymbol{b}}^T\underline{\nabla}\eta\varphi\right]e^{-\alpha t}dx\,dt\,d\varsigma$$

$$\leq \int_Q \lambda_{\max}\left|\eta_x e^{-\frac{1}{2}\alpha t}\right|\left|\varphi_x e^{-\frac{1}{2}\alpha t}\right|\,dx\,dt$$

$$+\int_Q\left|\hat{\boldsymbol{b}}^T\underline{\nabla}\eta e^{-\frac{1}{2}\alpha t}\right|\left|\varphi e^{-\frac{1}{2}\alpha t}\right|\,dx\,dt$$

$$\leq \lambda_{\max}\left(\int_Q \eta_x^2 e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}\left(\int_Q\varphi_x^2 e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}} +$$

$$b_{\max}\left(\int_Q(\eta_x^2+\eta_t^2)e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}\left(\int_Q\varphi^2 e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}$$

$$\leq \max\{\lambda_{\max}, b_{\max}\}\left(\int_Q(\eta_x^2+\eta_t^2)e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}\cdot$$

$$\left(\left(\int_Q\varphi_x^2 e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}+\left(\int_Q\varphi^2 e^{-\alpha t}\,dx\,dt\right)^{\frac{1}{2}}\right)$$

$$\leq \sqrt{2}\max\{\lambda_{\max}, b_{\max}\}\left(\int_Q(\eta_x^2+\eta_t^2)e^{-\alpha t}\,dx\,dt|||\varphi|||\right)^{\frac{1}{2}}$$

$$\leq \sqrt{2}\max\{\lambda_{\max}, b_{\max}\}|\eta|_1|||\varphi|||$$

$$=: c_2|\eta|_1|||\varphi|||\,.$$

This relation is obtained with the use of the estimate

$$\sqrt{a}+\sqrt{b} \leq \sqrt{2}\sqrt{a+b}$$

for all positive real numbers a and b.

These relations above, in combination with

$$|||\theta||| = |||\eta - \varphi||| \leq |||\eta||| + |||\varphi||| \leq \|\eta\|_1 + |||\varphi|||\,,$$

lead to

$$|||\theta||| \leq (1+\frac{c_2}{c_1})\|\eta\|_1\,.$$

Finally for $\hat{u} \in \mathrm{H}^p(Q)$ consider the *classical interpolation error estimate*

$$\|\eta\|_r \leq Dh^{s+1-r}\|\hat{u}\|_{s+1} \quad \forall_{0\leq r\leq s\leq p}\,. \tag{2.5.2}$$

In combination with the former relations this gives the *discretization error estimate*

$$|||\hat{u}-\hat{u}_h||| \leq D(1+\frac{c_2}{c_1})h^s\|\hat{u}\|_{s+1} \quad \forall_{0\leq s\leq 1} \tag{2.5.3}$$

which is uniform in ϵ if $b_{\min} > 0$. Since the $|||\cdot|||$ norm is slightly weaker than the $\|\cdot\|_1$ norm, the discretization error estimate is not of optimal order, even if $u \in \mathrm{H}^2(Q)$, the Sobolev space of order 2.

In order to investigate the conditions in (2.5.1) consider, as an example, the electromagnetic field penetration into a half-space of ferromagnetic material. In a Carthesian space the imposed magnetic field is parallel to the z-axis, while the induced electric field is parallel to the x-axis. In this case the magnetic vector potential has only one contributing component parallel to the z-axis, and under certain additional assumptions (see for instance [23]) this enables the formulation of a one-dimensional parabolic differential equation for this component.

The following non-linear parabolic differential equation models a magnetic vector potential in a one-dimensional piece of iron with a sinusoidal magnetic potential applied on its right boundary:

$$\begin{array}{lll} -(\nu(u_x^2(x,t))u_x(x,t))_x+\sigma u_t(x,t) & =0 & 0<x<1,\ 0<t<\infty \\ u(x,0) & =u_0(x) & 0<x<1 \\ u(1,t) & =c\sin(2\pi\omega t) & 0\leq t<\infty \\ u(0,t) & =0 & 0\leq t<\infty \end{array} \tag{2.5.4}$$

for some $L^2(0,1)$ integrable function g with $g(1) = g(0) = 0$. The parameters involved are

ω the angular frequency of the magnetic vector potential on the left boundary,
c the amplitude of the magnetic vector potential applied thereon,
σ the material-dependent electric conductivity, often a constant,
μ_0 the magnetic permeability of the vacuum, $\mu_0 = 4\pi 10^{-7}\mathrm{Hm}^{-1}$,
μ_r the relative magnetic permeability inside the iron, $\mu_r \equiv \mu_r(\zeta)$,
μ the magnetic permeability inside the iron, $\mu = \mu_0 \cdot \mu_r$ and
ν the magnetic reluctivity, $\nu \equiv \mu^{-1}$.

The reluctivity is in practice a non-linear material-dependent function depending on the square of the magnetic flux density $\zeta = u_x^2(x,t)$ (see [8] for an example of a measured reluctivity). Here, in all test cases considered, it will be modeled by

$$\nu(\zeta) \equiv \nu_{\min} + \nu_{\max} \cdot \frac{\arctan(\varsigma(\zeta - \zeta_0)) + \arctan(\varsigma(\zeta_0))}{\frac{\pi}{2} + \arctan(\varsigma(\zeta_0))} \tag{2.5.5}$$

where

$\nu_{\min}$ the relative minimum of the modelled reluctivity $\nu(\zeta)$,
$\nu_{\max}$ the relative maximum of this function,
ς its steepness and
ζ_0 its turning point.

Its derivative $\frac{\mathrm{d}}{\mathrm{d}\zeta}\nu$ therefore is given by

$$\frac{\mathrm{d}}{\mathrm{d}\zeta}\nu(\zeta) = \varsigma \cdot \nu_{\max} \cdot \frac{1}{\frac{\pi}{2} + \arctan(\varsigma(\zeta_0))} \cdot \frac{1}{1 + (\varsigma(\zeta - \zeta_0))^2} \cdot$$

See [9] for an example of this function for given $\nu_{\min}$, $\nu_{\max}$, ς and ζ_0.

Note that for the time-independent formulation of (2.5.4) with a reluctivity defined as above and boundary conditions $u_0(x) = 0$ and $u(1) = c$, the exact solution $\hat{u}$ is given by $\hat{u}(x) = c \cdot x$. Hence the solution of the stationary problem contains no layers.

Denoting the magnetic reluctivity ν by ϵ, note that in this case $\epsilon' \geq 0$ and $\lambda(\zeta) := \epsilon(\zeta) + 2\zeta\epsilon'(\zeta)$ is a continuous function, bounded above and below on $[0, \infty)$ by

$$0 < \lambda_{\min} := \epsilon_{\min} \leq \lambda(\zeta) \leq \lambda(\zeta_{\max}) =: \lambda_{\max} < \infty \quad \forall_{\zeta \in [0,\infty)},$$
$$\zeta_{\max} = \frac{1}{\varsigma}((4\varsigma^2\zeta_0^2 + 3)^{\frac{1}{2}} - \varsigma\zeta_0)$$

whence the first two conditions of (2.5.1) are satisfied. Due to $\hat{\boldsymbol{b}} = [0, 1]^T$, clearly $b_{\min} = 0$ and $b_{\max} = 1 < \infty$, whence all conditions of (2.5.1) are satisfied for all weighing functions $e^{-\alpha t}$ with $\alpha \geq 0$. For numerical results we refer to [9] and section 3.5.

2.6 Conclusions

The use of finite elements in both time and space, where the time-space domain is considered as a whole in the generation of finite elements, is efficient. The method is applicable also in multi-dimensional problems, where tetrahedron elements can be used, for instance. As has been shown, the stability of time-stepping on the larger time-slabs is an immediate consequence of the positive definiteness of the Jacobian matrix. The use of ordinary continuous finite element approximations enables the use of standard finite element packages for the time-space domain. Adaptive refinement of an initial grid on each time-slab in order to locate and fit steep gradients is advisable and is presently studied by the authors.

Finally the solution of the linear systems can be performed quite cheaply. Using still more efficient preconditioners, for instance those based on incomplete factorization or domain decomposition (see [7] and [11]), one can get methods for which the computational effort is not larger than about proportional to the number of node points.

Acknowledgements

This study was suggested by dr. Arne Wolfbrandt, ABB, Corporate Research, Västerås, Sweden.

2.7 References

[1] Axelsson O., *Finite element methods for convection-diffusion problems,* **in** Numerical Treatment of Differential Equations, (Strehmel K. ed.) Leipzig: Teubner 1988 (Teubner-Texte zur Mathematik; Bd. 104), 171-182 [Proceedings of the Fourth Seminar "Numdiff-4", Halle, 1987]

[2] Axelsson O., *A generalized conjugate gradient, least square method,* Numerische Mathematik, 51(1987), 209-227

[3] Axelsson O., *On global convergence of iterative methods,* **in** Iterative Solution of Nonlinear Systems of Equations, 1-19 LNM#953, (Ansorge R., Meis Th. and Törnig W. eds.), Springer Verlag, 1982

[4] Axelsson O., *On the numerical solution of convection dominated convection diffusion problems,* **in** Mathematical Methods in Energy Research (Gross K.I. ed.), 3-21, SIAM Philadelphia 1984

[5] Axelsson O., *The numerical solution of partial differential equations,* **in** Mathematics and Computer Science II: fundamental contributions in the Netherlands since 1945 (Hazewinkel M., Lenstra J.K. and Meertens L. eds.), 1-18, North-Holland 1986

[6] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[7] Axelsson O., Eijkhout V., Polman B. and Vassilevski P., *Iterative solution of singular pertubation 2^{nd} order boundary value problems by use of incomplete block-factorization methods,* BIT, 29(1989), 867-889

[8] Axelsson O. and Maubach J., *A time-space finite element discretization technique for the calculation of the electromagnetic field in ferromagnetic materials,* Journal for Numerical Methods in Engineering, 29(1989), 2085-2111

[9] Axelsson O. and Maubach J., *A time space finite element method for nonlinear convection diffusion problems,* **in** Notes on Numerical Fluid Mechanics, (Hackbush W. and Rannacher R. eds.) Vol. 30, 6-23, Vieweg, Braunschweig, 1990 [Proceedings of the Fifth GAMM-Seminar, Kiel, West Germany 1989]

[10] Axelsson O. and Maubach J., *On the updating and assembly of the Hessian matrix in finite element methods,* Computer Methods in Applied Mechanics and Engineering, 71(1988), 41-67

[11] Axelsson O. and Polman B., *A robust preconditioner based on algebraic substructuring and two-level grids,* **in** Robust Multi-Grid Methods (Hackbusch W. ed.), Notes on Numerical Fluid Mechanics, Vol. 23, 1-26, Vieweg, BraunSchweig, 1988

[12] Axelsson O. and Steihaug T., *Some computational aspects in the numerical solution of parabolic equations,* Journal of Computational and Applied Mathematics, 4(1978), 129-142

[13] Axelsson O. and Verwer J.G., *Boundary value techniques for initial value problems in ordinary differential equations,* Mathematics of Computation, 45(1985), 153-171

[14] Ciarlet P.G., *The Finite Element Method for Elliptic Problems,* North-Holland Publ., Amsterdam, 1978

[15] Friedman A., *Partial Differential Equations,* Holt, New York, 1969

[16] Hemker P.W.,*A numerical study of stiff two-point boundary value problems,* Ph.D. thesis, S.M.C., Amsterdam, 1977

[17] Houwen V.d. P.J.,*Construction of Integration Formulas for Initial Value Problems,* North-Holland , Amsterdam 1976

[18] Hughes T.J. and Brooks A., *A multi-dimensional upwind scheme with no crosswind diffusion,* **in** AMD, 34(1979), Finite element methods for convection dominated flows (Hughes T.J. ed.), ASME, New York

[19] Hughes T.J.R. and Hulbert M.,*Space-time finite element methods for elastodynamics: formulation and error estimates,* Computer Methods in Applied Mechanics and Engineering, 66(1988), 339-363

[20] Jamet P., *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain,* SIAM Journal on Numerical Analysis, 15(1978), 912-928

[21] Johnson C. and Pitkäranta J., *An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation,* Report MAT-A215, Institute of Mathematics, Helsinki University of Technology, Helsinki, Finland, 1984

[22] Kardestuncer H. (editor in chief) and Douglas H.N. (project editor), *Finite Element Handbook,* Mc Graw Hill, 1987

[23] Karlsson K.-E. and Wolfbrandt A., *An explicit technique for calculating the electromagnetic field and power losses in ferromag-*

netic materials, internal report 721-83, department for electrical analysis methods ASEA, Västerås, Sweden, 1983

[24] Lesaint P. and Raviart P.A., *On a finite element method for solving the neutron transport equation,* **in** Mathematical Aspects of Finite Elements in Partial Differential Equations (de Boor C. ed.), 89-123, Academic Press, New York, 1974

[25] Nečas J., *Introduction to the Theory of Nonlinear Elliptic Equations,* Prague, 1982

[26] Nečas J., *Les Méthodes Directes en Théorie des Equations Elliptiques,* Masson, Paris, 1967

[27] Sonneveld P., *CGS, a fast Lanczos-type solver for non-symmetric linear systems,* SIAM Journal on Scientific and Statistical Computing, 10(1989), 36-52

[28] Zienkiewicz O., *The Finite Element Method in Engineering Science,* 3^{rd} edition, Mc Graw-Hill, New York, 1977

3 A multi-dimensional streamline upwind approach

Revised version of: Maubach J.M.L., Preconditioned iterative methods for problems discretized in time-space, in Lecture Notes of the Summer school on Preconditioned Conjugate Gradient Methods and Applications, Nijmegen, The Netherlands 1989, 274-291. The theory has been generalized to the multi-dimensional case and a section on other global finite element applications has been included. The revised numerical tests are partially related to those in: Axelsson O. and Maubach J., A time-space finite element discretization technique for the calculation of the electromagnetic field in ferromagnetic materials, International Journal for Numerical Methods in Engineering 29(1989), 2085-2111; parts of which are reprinted with the kind permission of, and copyrighted by, John Wiley & Sons, Ltd.

Abstract

Time-stepping methods for parabolic problems require a careful choice of the stepsize for stability and accuracy. Even if a stable implicit time-stepping method is used, one might be forced to choose very small time-steps in order to get sufficient accuracy, if the solution has steep gradients, even if these occur only in a narrow part of the domain. Therefore the solution of the corresponding algebraic systems can be expensive since many time-steps have to be taken. The same considerations apply to for explicit time-stepping methods. In this chapter a discretization technique is presented, which uses finite element approximations in time and space simultaneously for a relatively large time-period, called time-slab. This technique may be repeatedly applied to obtain further parts of the solution in subsequent time intervals. It will be shown that, with the method proposed, the solution can be computed cheaply, even if it has steep gradients, and that stability is automatically guaranteed.

For the solution of the non-linear algebraic equations on each time-slab fast iterative methods can be used.

Key words: Time-stepping, Time-space finite elements, Nonlinear parabolic differential equations, Convection diffusion, Grid refinement
AMS(MOS) subject classifications: 65F10, 65M20, 65N30, 65N50

3.1 Introduction

The method most frequently used for the numerical integration of parabolic differential equations is the method of lines. Here one first uses a discretization of space derivatives by finite differences or finite elements and then uses some time-stepping method for the solution of the resulting system of ordinary differential equations. Such methods are, at least conceptually, easy to perform. However, they can be expensive if steep gradients occur in the solution, stability must be controlled, and the global error control can be troublesome.

This chapter considers a simultaneous discretization of space and time variables for a one-dimensional parabolic equation on a relatively long time interval, called time-slab. The discretization is repeated or adjusted for following time-slabs using continuous finite element approximations. In such a method the efficiency of finite elements is utilized by choosing a finite element grid in the time-space domain where the finite element grid has been adjusted to steep gradients of the solution both with respect to the space and the time variables. In this way, one solves all the difficulties with the classical approach since stability, discretization error estimates and global error control are automatically satisfied. Such a method has been discussed previously in [1] and [3]. The related boundary value techniques or global time integration for systems of ordinary differential equations have been discussed in several papers, see [9] and the references quoted therein. In [11] a time-space method with discontinuous elements in time has been used, which is based on methods in [12], [13] and [15].

In the present chapter a non-linear convection diffusion problem is considered. This problem is presented and reformulated as a two-dimensional boundary value problem in section 3.2. In section 3.3 the discrete problem and a solution method is formulated, and in section 3.4

the stability and discretization error estimates for the method are considered. Finally in section 3.5, numerical tests and a discussion of the grid generation method used, is found, and after a short overview of other possible applications of the global finite element method in section 3.6 some conclusions are drawn in section 3.7.

3.2 Parabolic differential equations

Let $\Omega \subset \mathbb{R}^n$, $n \geq 1$, be an open bounded and polygonal domain and consider the following multi-dimensional non-linear parabolic partial differential equation on the time-space interval $Q := \Omega \times (0, \infty)$:

$$\begin{aligned} -\nabla_x \cdot (\epsilon \underline{\nabla}_x u(\boldsymbol{x}, t)) + \boldsymbol{b} \underline{\nabla}_x u(\boldsymbol{x}, t) + \sigma u_t(\boldsymbol{x}, t) &= f(\boldsymbol{x}, t) \qquad (\boldsymbol{x}, t) \in Q \\ u(\boldsymbol{x}, 0) &= u_0(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Omega \\ u(\boldsymbol{x}, t) &= u_c(t) \qquad \boldsymbol{x} \in \partial\Omega, t \geq 0 \end{aligned} \tag{3.2.1}$$

where the diffusion ϵ and the flow velocity functions $\boldsymbol{b}$, σ satisfy $\epsilon \equiv \epsilon(|\underline{\nabla}_x u|^2)$ resp. $\boldsymbol{b} = \boldsymbol{b}(\boldsymbol{x}, t) \in \mathbb{R}^n$ and $\sigma = \sigma(\boldsymbol{x}, t)$, f is a source function, u_0 some $L^2(\Omega)$ integrable function and u_c is a square integrable Dirichlet boundary condition. In addition, assume that $\sigma \geq \sigma_0 > 0, \boldsymbol{b}, \sigma \in C^1(\overline{Q})$ (see chapter 1). Further, let $\underline{\nabla}_x \boldsymbol{b} + \sigma_t \leq 0$ and define $\epsilon' \equiv \frac{d}{d\zeta}\epsilon(\zeta)$ for $\zeta = |\underline{\nabla}_x u|^2$ and assume that $\epsilon' \geq 0$. The parabolic problem above occurs in many applications of which one was considered in [6].

As in chapter 1, in order to compute the solution of (3.2.1), a computational domain $\Omega \times (0, t_J] \subset Q$ is partitioned into a number of equidistant time-slabs $Q_j = \Omega \times (t_{j-1}, t_j]$ for $0 = t_0 < t_1 < \ldots < t_J < \infty$, assuming without loss of generality $t_j - t_{j-1} = \Delta t$ for all j (see fig. 1.2). The time-slabs have lower and upper boundaries denoted by $\Gamma_1 = \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \Omega \wedge t = t_{j-1}\}$ resp. $\Gamma_3 = \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \Omega \wedge t = t_j\}$, and the *cylinder surface* $\Gamma_c = \{(\boldsymbol{x}, t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \partial\Omega \wedge t \in [t_{j-1}, t_j]\}$. The number of such time-slabs is finite, independent of the choice of the grid parameter, associated with the finite elements. For the first time-slab Q_1 an initial value u_0 on Γ_1 has to be given, but for each following time-slab Q_{j+1} the solution at Γ_3 of Q_j will be taken to provide a Dirichlet boundary condition at Γ_1. With this approach problem (3.2.1) can be

rewritten into:

$$\begin{aligned} -\nabla_x \cdot (E \underline{\nabla}_x u(x,t)) + \hat{b}^T \underline{\nabla} u(x,t) &= f(x,t) \quad \text{in } Q_j \\ u(x,0) &= u_0(x) \quad \text{at } \Gamma_1 \\ u(x,t) &= u_c(t) \quad \text{on } \Gamma_c \end{aligned} \tag{3.2.2}$$

on each time-slab with

- tensor $E = \mathrm{Diag}(\epsilon(|\underline{\nabla}_x u|^2), \ldots, \epsilon(|\underline{\nabla}_x u|^2))$ of order n and flow field $\hat{b} \equiv \begin{bmatrix} b(x,t) \\ \sigma(x,t) \end{bmatrix} = \begin{bmatrix} b \\ \sigma \end{bmatrix} \in \mathbb{R}^{n+1}$, where it is assumed that $\sigma \geq \sigma_0 > 0$ in order to preserve the parabolic nature of the equation
- square integrable functions u_0 and u_c, prescribing the initial value and the Dirichlet boundary conditions on the cylinder surface
- the divergence operator $\nabla \cdot$ and gradient operator $\underline{\nabla}$ defined on the $n+1$ dimensional (x,t) space and
- square integrable source function f at Ω and initial value function u_0 at the Dirichlet boundary Γ_1.

Throughout this chapter the gradient operator $\underline{\nabla} = \underline{\nabla}_{x,t}$ will denote the space plus time derivatives contrary to $\underline{\nabla}_x$ which stands for the derivatives in space only.

Note that, analogous to chapter 1, there is no need to impose any boundary condition at the boundaries Γ_1 and Γ_3, because for all possible trial functions u and all test functions v the corresponding boundary integral

$$\oint_{\Gamma_3} v(E \underline{\nabla}_x u)^T n_x \, dx = \oint_{\Gamma_3} v\epsilon(|\underline{\nabla}_x u|^2) \underline{\nabla}_x u \cdot 0 \, dx = 0,$$

where n_x is the n-dimensional space component of n, the *unit outward normal* of the boundary $\partial \overline{Q}_j$. At this boundary the solution $\hat{u}$ of (3.2.2) and $\underline{\nabla}_x \hat{u}$ are initially unknown.

3.3 Weighted streamline upwind solution method

Consider the variational formulation of the non-linear two-dimensional problem (3.2.2) for a certain time-slab $Q := Q_j$. Let $\mathrm{H}^1(Q)$ be the

Sobolev space of order 1 on Q and define the boundary function γ at $\Gamma_D := \Gamma_1 \cup \Gamma_c$ by $\gamma := (u_0, u_c)$, i.e., $\gamma(\boldsymbol{x}, t) \equiv u_0(\boldsymbol{x})$ at Γ_1, $\gamma(\boldsymbol{x}, t) \equiv u_c(t)$ at Γ_c. To simplify the analysis, assume that there exists an extension of γ to Q in $H^1(Q)$, which excludes the occurrence of interior layers due to discontinuous boundary data. Furthermore, for the sake of simplicity, assume that $\sigma(\boldsymbol{x}, t) = \sigma_0 > 0$. Define the test and trial spaces by $H_0^1(Q) := \{v \in H^1(Q): v \equiv 0 \text{ at } \Gamma_D\}$ resp. $H_\gamma^1(Q) := \{u \in H^1(Q): u \equiv \gamma \text{ at } \Gamma_D\}$, both in the sense of traces, and define the partial differential operator $L(u)$ by

$$L(u) = -\nabla_{\boldsymbol{x}} \cdot (E \underline{\nabla}_{\boldsymbol{x}} u) + \hat{\boldsymbol{b}}^T \underline{\nabla} u - f\,. \tag{3.3.1}$$

The weighted streamline upwind variational formulation now becomes

$$\langle F(u), v \rangle = 0 \quad \forall_{v \in H_0^1(Q)}, u \in H_\gamma^1(Q) \tag{3.3.2}$$

where for a given triangulation $\mathcal{Q} = \{\triangle\}$ of the domain Q, fixed $\alpha \geq 0$ and fixed $\hat{\delta} > 0$, the gradient F for all $v \in H_0^1(Q)$ is defined by

$$\begin{aligned}
\langle F(u), v \rangle &= \int_Q \left[-\hat{\delta} \nabla \cdot (\hat{\boldsymbol{b}} L(u)) + L(u) \right] v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt \\
&= \int_Q \hat{\delta} L(u) \hat{\boldsymbol{b}}^T \underline{\nabla} v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt - \\
&\quad \int_Q \hat{\delta} L(u) \alpha b_2 v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt - \\
&\quad \oint_{\partial Q} \hat{\delta} L(u) v \mathrm{e}^{g(t)} \hat{\boldsymbol{b}}^T \boldsymbol{n}\, ds + \int_Q L(u) v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt \\
&= \hat{\delta} \sum_{\triangle \in \mathcal{Q}} \int_\triangle L(u) \hat{\boldsymbol{b}}^T \underline{\nabla} v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt + \\
&\quad (1 - \alpha \hat{\delta} \sigma) \int_Q L(u) v \mathrm{e}^{g(t)}\, d\boldsymbol{x}dt
\end{aligned} \tag{3.3.3}$$

since $L(u) = 0$ on $\Gamma_1 \cup \Gamma_3$ and $v = 0$ on Γ_c. Here $t \mapsto \mathrm{e}^{g(t)}$ is a weight function controlled by a continuous differentiable function g on $[0, \infty)$. This weight function can be useful to get better estimates of the discretization errors, as will be demonstrated in section 3.4. It suffices to consider the first time-slab, where the weight function $t \mapsto \mathrm{e}^{-\alpha t}$ is used.

The positive scalar $\hat{\delta}$ is the streamline upwind parameter, which will be used in order to get a strongly positive definite system for convection dominated problems and to obtain a discretization error estimate in the $H^1(\Omega)$ norm. The streamline upwind technique will increase the rate of convergence of certain generalized preconditioned conjugate gradient iterative methods.

Note that a solution of (3.2.2) is also a solution of (3.3.2). Unless $u \in H^2(Q)$, the leading term in (3.3.3) only exists as a sum of integrals over each individual element $\triangle$. Therefore dividing (3.3.3) by $1 - \alpha\hat{\delta}\sigma$ and setting $\delta := \hat{\delta}/(1 - \alpha\hat{\delta}\sigma)$ leads to the equivalent variational formulation

$$\langle F(u), v\rangle = 0 \quad \forall_{v\in H_0^1(Q)}, u \in H_\gamma^1(Q)$$

where now F is defined by

$$\langle F(u), v\rangle = \delta \sum_{\triangle\in Q} \int_\triangle L(u)\hat{\boldsymbol{b}}^T \underline{\nabla} v e^{-\alpha t}\, d\boldsymbol{x}dt + \int_Q L(u) v e^{-\alpha t}\, d\boldsymbol{x}dt$$

whence

$$\begin{aligned}\langle F(u), v\rangle = &\delta \sum_{\triangle\in Q} \int_\triangle -\nabla_{\boldsymbol{x}}\cdot(E\underline{\nabla}_{\boldsymbol{x}}u)\hat{\boldsymbol{b}}^T\underline{\nabla} v e^{-\alpha t}\, d\boldsymbol{x}dt + \\ &\delta \int_Q \left[\hat{\boldsymbol{b}}^T\underline{\nabla} u\cdot\hat{\boldsymbol{b}}^T\underline{\nabla} v - f\hat{\boldsymbol{b}}^T\underline{\nabla} v\right] e^{-\alpha t}\, d\boldsymbol{x}dt + \\ &\int_Q \left[(E\underline{\nabla}_{\boldsymbol{x}}u)^T\underline{\nabla}_{\boldsymbol{x}} v + \hat{\boldsymbol{b}}^T\underline{\nabla} u v - f v\right] e^{-\alpha t}\, d\boldsymbol{x}dt\end{aligned} \tag{3.3.4}$$

for all $v \in H_0^1(Q)$ where $\oint_{\partial Q} v e^{-\alpha t}(E\underline{\nabla}_{\boldsymbol{x}}u)^T\boldsymbol{n}_{\boldsymbol{x}}\, ds = 0$ dropped out due to $v = 0$ on Γ_c and the fact that $(E\underline{\nabla}_{\boldsymbol{x}}u)^T\boldsymbol{n}_{\boldsymbol{x}} \equiv 0$ at $\Gamma_1 \cup \Gamma_3$. Note that for $c = \alpha\sigma$

$$\hat{\delta} = h(\delta), \quad h{:}\,\delta \mapsto \frac{\delta}{1 + c\delta}.$$

For every positive c the map h is a strictly increasing function, a bijection from $[0, \infty)$ onto $[0, 1)$ whence $\hat{\delta}$ is uniformly bounded away from infinity in δ.

Linearization of this weak formulation by a damped Newton method now leads to a sequence of linear systems and solutions $u^{(k+1)} \in H_\gamma^1(Q)$

$$\langle F'(u^{(k)})(u^{(k+1)} - u^{(k)}), v\rangle = -\tau^{(k)}\langle F(u^{(k)}), v\rangle \quad \forall_{v\in H_0^1(Q)} \tag{3.3.5}$$

where the Gateaux directional derivative of F, the *Jacobian matrix* F', is defined as in chapter 1. For linear functionals F the Jacobian matrix is simply given by $\langle F'(u)w, v\rangle = \langle F(w), v\rangle$ for all functions u, v and w; in the non-linear case the Jacobian matrix is given by

$$\begin{aligned}\langle F'(u)w, v\rangle =& \delta \sum_{\Delta\in Q}\int_{\Delta} -\nabla_x\cdot(B\underline{\nabla}_x w)\cdot \hat{\boldsymbol{b}}^T\underline{\nabla} v e^{-\alpha t}\, dx dt + \\ & \delta\int_Q \hat{\boldsymbol{b}}^T\underline{\nabla} w\cdot \hat{\boldsymbol{b}}^T\underline{\nabla} v e^{-\alpha t}\, dx dt + \\ & \int_Q \left[(B\underline{\nabla}_x w)^T\underline{\nabla}_x v + \hat{\boldsymbol{b}}^T\underline{\nabla} w v\right] e^{-\alpha t}\, dx dt\end{aligned} \tag{3.3.6}$$

for all $v \in H_0^1(Q)$ and all $u, w \in H^1(Q)$, where the tensor B is defined by

$$B = E + 2E'\underline{\nabla} u\underline{\nabla} u^T, \tag{3.3.7}$$

with $E' = \mathrm{Diag}(\epsilon'(|\underline{\nabla}_x u|^2), \ldots, \epsilon'(|\underline{\nabla}_x u|^2))$, a matrix of order n. In order to see this consider the following lemma.

Lemma 3.3.1 *Let $\Omega \subset \mathbb{R}^n$ be an open and bounded domain, let E be a diagonal matrix of order n with diagonal matrix entries $\epsilon_{ii}(x, |\underline{\nabla}_x u|^2)$, and let*

$$\langle F(u), v\rangle = \int_\Omega (E\underline{\nabla}_x u)^T\underline{\nabla}_x v\, dx,$$

then

$$\langle F'(u)w, v\rangle = \int_\Omega \left[\{E + 2E'\underline{\nabla}_x u\underline{\nabla}_x u^T\}\underline{\nabla}_x w\right]^T \underline{\nabla}_x v\, dx$$

where $E' = \mathrm{Diag}(\epsilon'_{11}(x, |\underline{\nabla}_x u|^2), \ldots, \epsilon'_{nn}(x, |\underline{\nabla}_x u|^2))$. Further, if all diagonal elements of E are equal to $\epsilon = \epsilon(x, |\underline{\nabla}_x u|^2)$, then for $B := E + 2E'\underline{\nabla} u\underline{\nabla} u^T$

$$\sigma(B) = \{\epsilon, \epsilon + 2|\underline{\nabla}_x u|^2\epsilon'\},$$

where the first and second eigenvalues have multiplicity $n-1$ *respectively* 1.

Proof. Using the chain-rule for differentiation in a Banach-space, first note that

$$\begin{aligned}\langle F'(u)w, v\rangle &= \int_\Omega \left[E\underline{\nabla}_x w + 2(\underline{\nabla}_x u^T \underline{\nabla}_x w)\cdot E'\underline{\nabla}_x u\right]^T \underline{\nabla}_x v \; d\boldsymbol{x} \\ &= \int_\Omega (E\underline{\nabla}_x w)^T \underline{\nabla}_x v \; d\boldsymbol{x} + \\ &\quad 2\int_\Omega (\underline{\nabla}_x u^T \underline{\nabla}_x v)\cdot(E'\underline{\nabla}_x u^T \underline{\nabla}_x w)\; d\boldsymbol{x}\end{aligned}$$

This, in combination with the fact that

$$\begin{aligned}(\underline{\nabla}_x u^T \underline{\nabla}_x v, E'\underline{\nabla}_x u^T \underline{\nabla}_x w)_e &= (\underline{\nabla}_x v, \underline{\nabla}_x u E'\underline{\nabla}_x u^T \underline{\nabla}_x w)_e \\ &= (\underline{\nabla}_x v, E'\underline{\nabla}_x u \underline{\nabla}_x u^T \underline{\nabla}_x w)_e \\ &= (E'\underline{\nabla}_x u \underline{\nabla}_x u^T \underline{\nabla}_x w, \underline{\nabla}_x v)_e\end{aligned}$$

yields the desired result. Here, $(\cdot,\cdot)_e$ stands for the *Euclidian inner product.* The eigenvalues of B and the multiplicity thereof follow easily from the definition of B, exploiting that $\underline{\nabla}_x u \underline{\nabla}_x u^T$ is a matrix of rank 1. □

As an example consider the tensor matrices $E = \mathrm{Diag}(\epsilon(u_x^2))$ respectively $E = \mathrm{Diag}(\epsilon(u_x^2+u_y^2), \epsilon(u_x^2+u_y^2))$ for which the lemma above leads to

$$B = \mathrm{Diag}(\epsilon + 2u_x^2\epsilon') \text{ and } B = \begin{bmatrix} \epsilon + 2u_x^2\epsilon' & 2u_x u_y \epsilon' \\ 2u_x u_y \epsilon' & \epsilon + 2u_y^2\epsilon' \end{bmatrix}$$

Note that the first tensor corresponds to a time-slabbing problem with space-dimension 1, whereas the second tensor originates from the case of two space dimensions. This tensor also arises in the case of a static two-dimensional partial differential equation, since the lemma is not restricted to time-dependent problems. Now combining lemma 3.3.1 with integration in time leads to (3.3.6).

In order to study the Jacobian matrix on time-slab $Q = Q_j$ in detail, introduce the finite element test function spaces $\mathcal{H}$, $\mathcal{H}_0$ and $\mathcal{H}_\gamma$ on Q_j

as in section 2.5. Finally, let the norms $\|\cdot\|_{s,\alpha}$, $|\cdot|_{s,\alpha}$ and corresponding inner product on $\mathrm{H}^1(Q)$ be defined as in section 2.4. Norms $\|\cdot\|_{s,\alpha,\triangle}$ with additional domain subscript (here $\triangle$) denote weighted Sobolev norms of order s over this domain. With the set of norms introduced and under some assumptions to be derived on $\mathcal{H}$, and with the tensor ϵ and flow field $\hat{\boldsymbol{b}}$, $F'(u)$ will be seen to be uniformly positive definite on $\mathcal{H}_0$, i.e.,

$$\begin{aligned}\langle F'(u)v, v\rangle \geq & \int_Q \left[\tfrac{1}{2}\lambda_{\min}|\underline{\nabla}_x v|^2 + \tfrac{1}{2}\delta(\hat{\boldsymbol{b}}^T\underline{\nabla} v)^2\right] \mathrm{e}^{-\alpha t}\,dx\,dt + \\ & \int_Q \left[\tfrac{1}{2}\lambda_{\min} b_{\min} v^2\right] \mathrm{e}^{-\alpha t}\,dx\,dt + \\ & \tfrac{1}{2}\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} v^2\sigma\,dx \\ =: & |||v|||^2_{1,\alpha,\lambda,\delta} \geq \tfrac{1}{2}\chi\|v\|_1^2 \quad \forall_{v\in\mathcal{H}_0}\,,\end{aligned} \tag{3.3.8}$$

for some positive scalars $\lambda_{\min}$, $b_{\min}$ and χ. The subscripts λ, δ in $|||\cdot|||_{1,\alpha,\lambda,\delta}$ henceforth are omitted for simplicity. The difference of subsequent approximate discrete solutions $u_h^{(k+1)} - u_h^{(k)}$ is an element of $\mathcal{H}_0$, whence (3.3.8) implies that the damped Newton algorithm given by (3.3.5) will converge for properly chosen damping parameters $\tau^{(k)}$ (see e.g. [2]). The relation (3.3.8) implies that there can be at most one solution of equation (3.3.5) as, using a standard inequality,

$$0=\langle F(u)-F(v), u-v\rangle = \int_0^1 \langle F'(u+\varsigma(v-u))(u-v), u-v\rangle\, d\varsigma \geq c\|u-v\|_1^2$$

for all solutions $u, v \in \mathrm{H}^1_\gamma(Q)$ implying $u-v \in \mathrm{H}^1_0(Q)$. Under appropriate assumptions on F the nonsymmetric Galerkin type equation (3.3.5) has a solution $u^{(k+1)} \in \mathrm{H}^1_\gamma(Q)$ according to [10].

In order to show that the Jacobian matrix is coercive, i.e., to show that (3.3.8) is satisfied, let $\sigma(A)$ denote the spectrum of a matrix A and assume that

- There exist bounds $\lambda_{\min}$ and $\lambda_{\max}$ such that for all functions $u \in \mathcal{H}$ and all $(\boldsymbol{x}, t) \in Q$

$$0 < \lambda_{\min} \leq \{\lambda \in \mathbb{R}: \lambda \in \sigma(B)\} \leq \lambda_{\max} < \infty$$

where $B = B(u(\boldsymbol{x}, t))$.

- The scalars $\lambda_{\min}$ and $\lambda_{\max}$ satisfy

$$\lambda_{\max}^2 \delta \leq (\frac{h}{C_0})^2 \lambda_{\min} .$$

- There exists a positive scalar C_0 such that the following *inverse inequality* holds

$$|\triangle v|_{0,\alpha,\triangle} \leq C_0 h^{-1} |\underline{\nabla}_x v|_{0,\alpha,\triangle} \quad \forall_{v \in \mathcal{H}} \tag{3.3.9}$$

(see e.g. [10], page 140, for arbitrary high order of finite element basis functions). Note that this is trivially true if $\mathcal{H}$ is the space of piecewise linear finite element basis functions on the triangulation $\mathcal{Q}$.

- On each $\triangle \in \mathcal{Q}$ the tensor B satisfies $|\nabla_x \cdot (B \underline{\nabla}_x v)|_0 \leq c |\triangle_x v|_0$ for some scalar c, which is set to 1 for ease of notation. Note that this condition is satisfied for all differentiable functions ϵ if piecewise linear or constant finite element basis functions are used. Also, in the cases where the diffusion changes only in time, i.e., where $\epsilon = \epsilon(t)$, or where ϵ is elementwise constant, this condition is satisfied.

Now consider the terms in (3.3.6) separately. Exploiting the above assumptions, the first term in (3.3.6) can be estimated below because it is bounded above by

$$\begin{aligned}
&\delta |\nabla_x \cdot (B \underline{\nabla}_x v)|_{0,\alpha,\triangle} \cdot |\hat{\boldsymbol{b}}^T \underline{\nabla} v|_{0,\alpha,\triangle} \\
\leq &\tfrac{1}{2} \lambda_{\max}^2 \delta |\triangle_x v|_{0,\alpha,\triangle}^2 + \tfrac{1}{2} \delta |\hat{\boldsymbol{b}}^T \underline{\nabla} v|_{0,\alpha,\triangle}^2 \\
\leq &\tfrac{1}{2} \lambda_{\max}^2 \delta C_0^2 h^{-2} |\underline{\nabla}_x v|_{0,\alpha,\triangle}^2 + \tfrac{1}{2} \delta |\hat{\boldsymbol{b}}^T \underline{\nabla} v|_{0,\alpha,\triangle}^2 \\
\leq &\tfrac{1}{2} \lambda_{\min} |\underline{\nabla}_x v|_{0,\alpha,\triangle}^2 + \tfrac{1}{2} \delta |\hat{\boldsymbol{b}}^T \underline{\nabla} v|_{0,\alpha,\triangle}^2
\end{aligned} \tag{3.3.10}$$

for all $v \in \mathrm{H}_h^1(Q)$, since $|(v, w)_{s,\alpha}| \leq |v|_{s,\alpha} |w|_{s,\alpha}$ for all $v, w \in \mathrm{H}^s(\Omega)$ and $|ab| \leq \frac{1}{2}(a^2 + b^2)$ for all positive a, b. An analysis of the separate sub-terms in the third expression of (3.3.6) shows that

$$\lambda_{\min} \int_Q |\underline{\nabla}_x v|^2 \mathrm{e}^{-\alpha t} \, dx dt \leq \int_Q (B(u) \underline{\nabla} v)^T \underline{\nabla} v \mathrm{e}^{-\alpha t} \, dx dt \tag{3.3.11}$$

for all $u, v \in \mathrm{H}^1(Q)$, and that, analogous to the derivation in (2.4.9),

$$\int_Q (\hat{\boldsymbol{b}}^T \underline{\nabla} v) v \mathrm{e}^{-\alpha t}\, d\boldsymbol{x}dt \geq \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} v^2 \sigma\, d\boldsymbol{x} + b_{\min} \int_Q v^2 \mathrm{e}^{-\alpha t}\, d\boldsymbol{x}dt$$

for all $v \in \mathrm{H}_0^1(Q)$, where $b_{\min} := \inf\{\tfrac{1}{2}(\alpha\sigma(\boldsymbol{x},t) - \nabla\cdot\hat{\boldsymbol{b}}(\boldsymbol{x},t)) : (\boldsymbol{x},t) \in Q\}$. Now (3.3.10), (3.3.11) and the above show that the Jacobian matrix satisfies estimate (3.3.8)

$$\begin{aligned} \langle F'(u)\imath, v\rangle \geq & \int_Q \left[\tfrac{1}{2}\lambda_{\min}|\underline{\nabla}_{\boldsymbol{x}} v|^2 + \tfrac{1}{2}\delta(\hat{\boldsymbol{b}}^T \underline{\nabla} v)^2\right] \mathrm{e}^{-\alpha t}\, d\boldsymbol{x}dt + \\ & \int_Q \left[b_{\min} v^2\right] \mathrm{e}^{-\alpha t}\, d\boldsymbol{x}dt + \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} v^2 \sigma\, d\boldsymbol{x} \end{aligned} \tag{3.3.12}$$

for all $v \in \mathcal{H}_0$ because of (3.3.9).

In order to get a coercivity estimate in the Sobolev 1 norm note that one has

$$\lambda_{\min}|\underline{\nabla}_{\boldsymbol{x}} v|^2 + \delta(\hat{\boldsymbol{b}}^T \underline{\nabla} v)^2 =: (E_{n+1}\underline{\nabla} v)^T \underline{\nabla} v$$

where

$$E_{n+1} = \lambda_{\min} \begin{bmatrix} I_n & \emptyset \\ \emptyset & 0 \end{bmatrix} + \delta \hat{\boldsymbol{b}}\hat{\boldsymbol{b}}^T \tag{3.3.13}$$

with I_n the identity matrix of order n. Omitting the subscript 'min' for $\lambda_{\min}$ to simplify the notations, elementary computations show that for $n \geq 1$

$$\begin{aligned} \mathrm{Det}(E_n - zI_n) =& (\lambda - z)^{n-1} \cdot (z^2 - (\delta|\hat{\boldsymbol{b}}|^2 + \lambda)z + \lambda\delta\sigma^2) \\ =& (\lambda - z)^{n-1} \cdot p_\lambda(z)\,. \end{aligned} \tag{3.3.14}$$

Under the substitution $|\hat{\boldsymbol{b}}|^2 = (1+\varsigma)\sigma^2$ for $\varsigma \geq 0$, the discriminant of the factor $p_\lambda(z)$ is equal to the following quadratic polynomial in λ

$$\begin{aligned} d(\lambda) :=& (\delta|\hat{\boldsymbol{b}}|^2 + \lambda)^2 - 4\lambda\delta\sigma^2 \\ =& \lambda^2 + 2\delta(|\hat{\boldsymbol{b}}|^2 - 2\sigma^2)\lambda + \delta^2|\hat{\boldsymbol{b}}|^4 \\ =& \lambda^2 + 2(\varsigma - 1)\delta\sigma^2\lambda + (1+\varsigma)^2\delta^2\sigma^4\,. \end{aligned}$$

The discriminant of $d(\lambda)$ is equal to

$$4(1-\varsigma)^2\delta^2\sigma^4 - 4(1+\varsigma)^2\delta^2\sigma^4 = -16\varsigma\delta^2\sigma^4 \leq 0$$

leading to $d(\lambda) \geq 0$ for all values of λ. This inequality in turn guarantees that the factor p_λ in (3.3.14) has at least one positive real root. As the product of p_λ's roots is equal to its last term $\lambda\delta\sigma^2$ this ensures the existence of two positive roots. Therefore, all eigenvalues of E_n are positive for all possible combinations of $\lambda_{\min} > 0$ and $\hat{\boldsymbol{b}}$, for all $n \geq 1$. Note that for $\epsilon \ll 1$ the roots of $p_\lambda(z)$ are of order δ and of order $\lambda_{\min}$, implying

$$\sigma(E_n) = \{\lambda_{\min}, O(\lambda_{\min}), O(\delta)\} \subset (0, \infty).$$

Denoting the smallest eigenvalue of E_n with χ, this leads to

$$\tfrac{1}{2}\lambda_{\min}|\underline{\nabla}_x v|^2 + \tfrac{1}{2}\delta(\hat{\boldsymbol{b}}^T\underline{\nabla} v)^2 \geq \tfrac{1}{2}\chi(|\underline{\nabla}_x v|^2 + v_t^2) \qquad (3.3.15)$$

whence for $\chi \leq 1$ the Jacobian matrix

$$\langle F'(u)v, v\rangle \geq \tfrac{1}{2}\chi\|v\|_1^2 + \tfrac{1}{2}\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} v^2\sigma \, d\boldsymbol{x} \quad \forall_{v\in\mathcal{H}_0}\forall_{u\in \mathrm{H}^1(Q)} \qquad (3.3.16)$$

is uniformly positive definite for positive $b_{\min}$. The condition $O(\lambda\delta) = O(h^2)$, following from the relation below (3.3.9), implies $\chi = O(h)$ for $\lambda = O(h)$ and $\delta = O(h)$, and $\chi = O(h^2)$ if one of these two scalars is $O(1)$ and the other is $O(h^2)$.

In the situation where $b_{\min} = 0$, the restriction to a certain time $t \in (t_{j-1}, t_j]$ of piecewise polynomial functions v on Q, will also be piecewise polynomial on Ω, in particular $v(\boldsymbol{x}, t) \in \mathrm{H}^1(\Omega)$. As in chapter 2, due to a *Friedrichs inequality*, there exists a positive scalar $\beta > 0$, not depending on v, such that,

$$\oint_{\partial\Omega} v^2(\boldsymbol{x}, t)\, ds + \int_\Omega |\underline{\nabla}_x v(\boldsymbol{x}, t)|^2 \, d\boldsymbol{x} \geq \beta \cdot \int_\Omega v^2(\boldsymbol{x}, t) + |\underline{\nabla}_x v(\boldsymbol{x}, t)|^2 \, d\boldsymbol{x}.$$

Since v is piecewise polynomial on Ω and the boundary of the space-domain does not vary with time, integration of the expression above with respect to the time shows that

$$\Big(\int_Q |\underline{\nabla}_x v|^2 \mathrm{e}^{-\alpha t} d\boldsymbol{x}\, dt\Big)^{\frac{1}{2}} \text{ and } \Big(\int_Q \Big[v^2 + |\underline{\nabla}_x v|^2\Big] \mathrm{e}^{-\alpha t} d\boldsymbol{x}\, dt\Big)^{\frac{1}{2}} \qquad (3.3.17)$$

are equivalent norms on $\mathcal{H}_0 \subset \mathrm{H}^1(Q)$. Hence, for $b_{\min} = 0$

$$\begin{aligned}\langle F'(u)v, v\rangle \geq & \int_Q \left[\tfrac{1}{2}\lambda_{\min}\beta|\underline{\nabla}_x v|^2 + \tfrac{1}{2}\delta(\hat{\boldsymbol{b}}^T\underline{\nabla}v)^2\right] \mathrm{e}^{-\alpha t}dx\,dt + \\ & \int_Q \left[\tfrac{1}{2}\beta v^2\right] \mathrm{e}^{-\alpha t}dx\,dt + \tfrac{1}{2}\mathrm{e}^{-\alpha T}\oint_{\Gamma_3} v^2\sigma\, dx\end{aligned} \tag{3.3.18}$$

for all $v \in \mathcal{H}_0$. This estimate is not bounded uniformly in ϵ, but contrary to the previous estimate (3.3.8) it is also valid for $\alpha = b_{\min} = 0$. Hence (3.3.8) will be used mainly for singularly perturbed problems while relation (3.3.17) will be used for regular problems. The coercivity constant estimate in the Sobolev 1 norm can now be obtained from (3.3.15).

In order to give a discretization error estimate in section 3.4 it is necessary to show that the Jacobian matrix is a bounded functional. To this end, note that

- For all $a, b, \gamma \in \mathbb{R}$, $\gamma > 0$

$$|ab| \leq \frac{\gamma}{2}a^2 + \frac{1}{2\gamma}b^2\,. \tag{3.3.19}$$

- The following relationship is valid for an arbitrary flow field $\hat{\boldsymbol{b}}$

$$\begin{aligned}(\hat{\boldsymbol{b}}^T\underline{\nabla}w)^2 \leq & \|\hat{\boldsymbol{b}}\|_e^2\|\underline{\nabla}w\|_e^2 = (|\boldsymbol{b}|^2 + \sigma^2)\|\underline{\nabla}w\|_e^2 \\ = & b_{\max}^2 \cdot \|\underline{\nabla}w\|_e^2\end{aligned} \tag{3.3.20}$$

for all $w \in \mathrm{H}^1(Q)$.

Here it is assumed that $b_{\max} < \infty$ and $\|\cdot\|_e$ stands for the Euclidian norm.

Using (3.3.6) and (3.3.10) it is easy to derive that $|\langle F'(u)w, v\rangle|$ is bounded above by

$$\begin{aligned}&\frac{1}{2\gamma}M(w) + \frac{\gamma}{2}\delta|\hat{\boldsymbol{b}}^T\underline{\nabla}v|_{0,\alpha}^2 + \frac{1}{2\gamma}\delta|\hat{\boldsymbol{b}}^T\underline{\nabla}w|_{0,\alpha}^2 + \frac{\gamma}{2}\delta|\hat{\boldsymbol{b}}^T\underline{\nabla}v|_{0,\alpha}^2 + \\ &\frac{1}{2\gamma}\lambda_{\max}|\underline{\nabla}_x w|_{0,\alpha}^2 + \frac{\gamma}{2}\lambda_{\max}|\underline{\nabla}_x v|_{0,\alpha}^2 + \frac{1}{2\gamma}|\hat{\boldsymbol{b}}^T\underline{\nabla}w|_{0,\alpha}^2 + \frac{\gamma}{2}|v|_{0,\alpha}^2\end{aligned}$$

which is at its turn bounded above by

$$\begin{aligned} &\leq \frac{1}{2\gamma}\left[M(w) + (1+\delta)b_{\max}^2|w|_{1,\alpha}^2 + \lambda_{\max}|\underline{\nabla}_x w|_{0,\alpha}^2\right] + \\ &\frac{\gamma}{2}\int_Q \left[v^2 + \lambda_{\max}|\underline{\nabla}_x v|^2 + 2\delta(\hat{\boldsymbol{b}}^T\underline{\nabla} v)^2\right] \mathrm{e}^{-\alpha t}\, dx dt \end{aligned} \tag{3.3.21}$$

for arbitrary $\gamma > 0$ where $M(w)$ is defined by

$$M(w) := \lambda_{\max}^2 \sum_{\Delta \in \mathcal{Q}} \delta |\Delta_x w|_{0,\alpha,\Delta}^2$$

for all functions w piecewise in $\mathrm{H}^2(Q)$.

3.4 Discretization error estimate

In order to estimate the discretization error, consider the exact solution $\hat{u} \in \mathrm{H}^1_\gamma(Q)$, the discrete solution $\hat{u}_h \in \mathcal{H}_\gamma$ and its interpolant $\hat{u}_I \in \mathcal{H}_\gamma$ as introduced in section 2.5. In addition, define the discretization error $\theta := \hat{u} - \hat{u}_h \in \mathrm{H}^1(Q)$, the interpolation error $\eta := \hat{u} - \hat{u}_I \in \mathrm{H}^1(Q)$ and the interpolation minus the discretization error $\varphi := \hat{u}_h - \hat{u}_I \in \mathcal{H}_0$. Note that $\varphi \equiv 0$ at Γ_D and that in fact $\hat{u}_I \in \mathcal{H}_\gamma$ and $\theta, \eta \in \mathrm{H}^1_0(Q)$ due to the fact that $\mathcal{H}_\gamma \neq \emptyset$.

Now assume that ϵ, $\hat{\boldsymbol{b}}$ with positive σ and α are such that for all $u \in \mathrm{H}^1(Q)$ the following conditions are satisfied

$$\begin{cases} \lambda_{\min} = \inf\{\lambda \in \mathbb{R}: \lambda \in \sigma(B)\} > 0 \\ \lambda_{\max} = \sup\{\lambda \in \mathbb{R}: \lambda \in \sigma(B)\} < \infty \\ b_{\min} = \inf\{\frac{1}{2}(\alpha\sigma(\boldsymbol{x},t) - \nabla\cdot\hat{\boldsymbol{b}}(\boldsymbol{x},t)): (\boldsymbol{x},t) \in Q\} > 0 \\ b_{\max} = \sup\{\max\{\left|\hat{\boldsymbol{b}}(\boldsymbol{x},t)\right|\}: (\boldsymbol{x},t) \in Q\} < \infty \\ \lambda_{\max}^2\delta \leq \max\{(\frac{h}{C_0})^2, (\frac{h}{D})^2\}\lambda_{\min} \end{cases} \tag{3.4.1}$$

for some positive scalar D, to be specified below. Let γ be a positive constant small enough such that

$$\min\{\frac{\lambda_{\min}}{\lambda_{\max}} - \gamma, 1 - 2\gamma, b_{\min} - \tfrac{1}{2}\gamma, 1\} > 0$$

independent of the upwind scalar δ. Then for $b_{\min} > 0$ with the use of (3.3.12)

$$\begin{aligned}\langle F(\hat{u}_h) - F(\hat{u}_I), \varphi\rangle &= \langle \int_0^1 F'(\hat{u}_I + \varsigma\varphi)\varphi \, d\varsigma, \varphi\rangle \\ &\geq \int_0^1 |||\varphi|||_{1,\alpha}^2 \, d\varsigma = |||\varphi|||_{1,\alpha}^2\end{aligned} \tag{3.4.2}$$

Further

$$\langle F(\hat{u}_h) - F(\hat{u}_I), v\rangle = \langle F(\hat{u}) - F(\hat{u}_I), v\rangle \quad \forall_{v \in H_h^1(\Omega)} \tag{3.4.3}$$

and according to (3.3.21) for all $\alpha \geq 0$

$$\begin{aligned}\langle F(\hat{u}) - F(\hat{u}_I), \varphi\rangle &= \langle \int_0^1 F'(\hat{u}_I + \varsigma\eta)\eta \, d\varsigma, \varphi\rangle \\ &\leq \frac{1}{2\gamma}\left[M(\eta) + (1+\delta)b_{\max}^2 |\eta|_{1,\alpha}^2 + \lambda_{\max} |\underline{\nabla}_x \eta|_{0,\alpha}^2\right] + \\ &\quad \frac{\gamma}{2}\int_Q \left[\varphi^2 + \lambda_{\max}|\underline{\nabla}_x \varphi|^2 + 2\delta(\hat{\boldsymbol{b}}^T \underline{\nabla}\varphi)^2\right] \mathrm{e}^{-\alpha t} dx\, dt\,.\end{aligned}$$

A combination of these three relations leads for γ small enough to the existence of a positive scalar c_0, independent of λ and δ, such that

$$c_0 |||\varphi|||_{1,\alpha}^2 \leq \frac{1}{2\gamma}\left[M(\eta) + c_1 \|\eta\|_{1,\alpha}^2\right] \quad \forall_{\alpha \geq 0} \tag{3.4.4}$$

where $c_1 := (1+\delta)b_{\max}^2 + \lambda_{\max}$. Now suppose the finite element subspace of $H^1(Q)$ under consideration is the space spanned by piecewise polynomials of degree k on the triangulation $\mathcal{Q}$. Let u_I denote the corresponding Lagrangian or Hermitian interpolant of a function $u \in H^{s+1}(Q)$. With the use of

$$\sum_i a_i^2 \leq (\sum_i |a_i|)^2 \quad \forall_{a_i \in \mathbf{R}}$$

in combination with the classical interpolation error estimate (2.5.2)

$$\sum_{\Delta \in \mathcal{Q}} \|\hat{u} - \hat{u}_I\|_{r,\alpha,\Delta} \leq D h^{s+1-r} \|\hat{u}\|_{s+1} \quad \forall_{0 \leq r \leq s \leq k}$$

(see [4], theorem 5.6) this leads to

$$\begin{aligned}
|||\varphi|||_1^2 &\leq \frac{1}{2c_0\gamma}\left[M(\eta) + c_1\|\eta\|_{1,\alpha}^2\right] \\
&\leq \frac{1}{2c_0\gamma}\left[\lambda_{\max}^2\delta\left(\sum_{\Delta\in\mathcal{Q}}\|\triangle_x\eta\|_{0,\alpha,\Delta}\right)^2 + c_1\|\eta\|_{1,\alpha}^2\right] \\
&\leq \frac{1}{2c_0\gamma}\left[\lambda_{\max}^2\delta D^2 h^{2s-2}\|\hat{u}\|_{s+1}^2 + c_1 D^2 h^{2s}\|\hat{u}\|_{s+1}^2\right] \\
&\leq \frac{1}{2c_0\gamma}(\lambda_{\min} + c_1 D^2)h^{2s}\|\hat{u}\|_{s+1}^2 \\
&=: c_2 h^{2s}\|\hat{u}\|_{s+1}^2
\end{aligned}$$

for all $0 \leq 2 \leq s \leq k$. In the case of piecewise linear Lagrangian basis functions note that for $s = k = 1$

$$\|\eta_{xx}\|_{0,\alpha,\Delta} = \|(\hat{u} - \hat{u}_I)_{xx}\|_{0,\alpha,\Delta} = \|\hat{u}_{xx}\|_{0,\alpha,\Delta} \leq h^{s-1}\|\hat{u}\|_{s+1}\,.$$

Finally, in combination with the triangle inequality $\|\theta\|_1 = \|\eta - \varphi\|_1 \leq \|\eta\|_1 + \|\varphi\|_1$, this leads to a *discretization error estimate* satisfying

$$\begin{aligned}
|||\hat{u} - \hat{u}_h|||_1 &\leq C \cdot h^s\|\hat{u}\|_{s+1} \\
\|\hat{u} - \hat{u}_h\|_1 &\leq C\chi^{-\frac{1}{2}} \cdot h^s\|\hat{u}\|_{s+1}
\end{aligned} \tag{3.4.5}$$

for all s and k as above and χ as defined by (3.3.15). Note that the error estimate is of optimal order for $\hat{u} \in \mathrm{H}^2(\Omega)$ and that the error constant is $O(1)$ using the $|||\cdot|||_1$ norm.

To analyze the boundedness in the space and time $\mathrm{H}^1(Q_j)$ norm of the discrete solution $\hat{u}_h$ consider (3.3.4). For the sake of simplicity assume that the Dirichlet boundary condition u_c on the cylinder surface Γ_c is a *homogeneous boundary condition*, i.e., $u_c = 0$. Then the discrete solution $\hat{u}_h$ on time-slab Q_j with triangulation $\mathcal{Q}_j$ is bounded by the global data ϵ, $\hat{b}$, f and initial data u_0 on that time-slab because under the appropriate assumptions posed in the beginning of this section

$\langle F(\hat{u}_h), \hat{u}_h \rangle = 0$ implies

$$
\begin{aligned}
&\|\hat{u}_h\|^2_{1,\alpha,\delta} + \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} \hat{u}_h^2 \sigma \, ds - \tfrac{1}{2} \oint_{\Gamma_1} \hat{u}_h^2 \sigma \, ds \\
\leq & \delta \sum_{\Delta \in \mathcal{Q}} \int_{\Delta} \nabla_x \cdot (E \underline{\nabla}_x \hat{u}_h) \cdot \hat{\boldsymbol{b}}^T \underline{\nabla} \hat{u}_h \mathrm{e}^{-\alpha t} \, d\boldsymbol{x} dt + \\
& \delta \int_Q (\hat{\boldsymbol{b}}^T \underline{\nabla} \hat{u}_h)^2 \mathrm{e}^{-\alpha t} \, d\boldsymbol{x} dt + \\
& \int_Q \left[(E \underline{\nabla}_x \hat{u}_h)^T \underline{\nabla}_x \hat{u}_h \cdot \hat{\boldsymbol{b}}^T \underline{\nabla} \hat{u}_h \hat{u}_h \right] \mathrm{e}^{-\alpha t} \, d\boldsymbol{x} dt \\
= & \int_Q (f + \hat{\boldsymbol{b}}^T \nabla \cdot f) \hat{u}_h \mathrm{e}^{-\alpha t} \, d\boldsymbol{x} dt \\
\leq & \|f + \hat{\boldsymbol{b}}^T \nabla \cdot f\|_{1,\alpha} \cdot \|\hat{u}_h\|_{1,\alpha}
\end{aligned}
\tag{3.4.6}
$$

for a some positive constant c, implying that $\|\hat{u}_h\|_{1,\alpha}$ is bounded. This is equivalent to

$$
\begin{aligned}
c\|\hat{u}_h\|^2_{1,\alpha} - \|f + \hat{\boldsymbol{b}}^T \nabla \cdot f\|_{1,\alpha} \cdot \|\hat{u}_h\|_{1,\alpha} \leq & \tfrac{1}{2} \int_\Omega \sigma(\boldsymbol{x}, 0) u_0(\boldsymbol{x})^2 \, d\boldsymbol{x} - \\
& \tfrac{1}{2}\mathrm{e}^{-\alpha T} \oint_{\Gamma_3} \hat{u}_h^2 \sigma \, d\boldsymbol{x}
\end{aligned}
$$

for some positive constant c depending on the data ϵ, δ and α. Denoting the right-hand side of the latter equation by c_b and $\|f + \hat{\boldsymbol{b}}^T \nabla \cdot f\|_{1,\alpha}$ by s_b this leads to

$$
\begin{cases}
\|\hat{u}_h\|_{1,\alpha} & \leq \dfrac{1}{c} \|f + \hat{\boldsymbol{b}}^T \nabla \cdot f\|_{1,\alpha} \qquad \text{if } s_b > 0 \text{ and } c_b \leq 0 \\
\displaystyle\oint_{\Gamma_3} \hat{u}_h^2 \sigma \, ds & \leq \mathrm{e}^{\alpha T} \displaystyle\int_\Omega u_0^2(\boldsymbol{x}) \sigma(\boldsymbol{x}, 0) \, d\boldsymbol{x} \qquad \text{if } s_b = 0 .
\end{cases}
$$

For many important equations of the type (3.2.2) the weighing scalar α can be taken zero, leading to a discrete solution bounded in L^2-norm in time. Note that $u_0 = 0$ leads to $c_b \leq 0$ and therefore to the boundedness of the discrete solution by the source function f and flow field $\hat{\boldsymbol{b}}$.

3.5 Grid generation and numerical results

In order to study the performance of iterative solution methods for the global time-space finite element discretization technique proposed, three test problems in one-dimensional space are considered (see fig. 1.2). Depending on the choice of the diffusion function ϵ, there may appear a parabolic layers along Γ_2 and Γ_4. If the grid would not be refined here, oscillations would arise with the finite elemenent discretization method used, even in the case of a standard streamline upwind method, because no artificial diffusion perpendicular to the streamlines is used. However, the use of a fine grid along this layer makes artificial diffusion unnecessary, and in addition provides an accurate resolution of the layers.

Table 3.5.1 Grid generation details.

Test	Grid	T	N	Fig.
1	$Q_1^{(0)}$	12	12	
	$Q_1^{(10)}$	3218	855	3.1
	$Q_5^{(10)}$	3216	856	3.3
2	$Q_1^{(0)}$	6	8	
	$Q_1^{(10)}$	5290	1377	3.5
	$Q_5^{(10)}$	4882	1281	3.8
3	$Q_1^{(0)}$	80	54	
	$Q_1^{(12)}$	38004	9605	3.11

The subsequently approximated parts of the solution are piecewise linear. The old grid points at Γ_3 of Q_j must therefore be used as grid points for the boundary Γ_1 of the new time-slab domain Q_{j+1}, because otherwise the restriction of the discrete solution on Γ_3 (Q_j) will not be exactly represented by the finite element functions on the new subdomain. However, more grid points may be added, where the Dirichlet boundary conditions are determined by linear interpolation, to represent a boundary layer better. Also, one can adjust the grid such that one ends up with fewer nodes on Γ_3 than on Γ_1, which is convenient if the solution gets smoother with increasing time.

As an initial solution for the non-linear iterations on each time-slab, the initial solution for the first time-slab is used, imposing the Dirichlet boundary condition on Γ_c. In practice, it would have been better to use the numerical solution of the previous time-slab contrary to the solution of the first time-slab.

Table 3.5.1 gives a survey of the grids to be used for the numerical tests. For each grid $\mathcal{Q}_j^{(k)}$ (see sections 1.5 and 5.4) the space domain as well as the number of triangles T and the number of grid points N, depending on the time-slab, are given. The grids are constructed using an adaptive refinement procedure, as described in section 5.9.

Table 3.5.2 The test cases.

Test case	No. 1	No. 2	No. 3
Ω	(0.00,0.03)	(0.00,0.06)	(0,1)
(t_0, t_1)	(0.00,0.02)	(0.00,0.02)	$(0,\frac{5}{8})$
$\nu_{\min}$	10^{-4}	10^{-4}	10^{-6}
$\nu_{\max}$	0	$4.6 \cdot 10^{-3}$	0
ζ_0	–	4.0	–
ς	–	3.0	–
$\hat{\boldsymbol{b}}$	$[0,1]^T$	$[0,1]^T$	$[\frac{9}{10}(x+2),1]^T$
f	0	0	0
$l(t)$	0	0	1
$r(t)$	$h(t)$	$h(t)$	0
$u_0(\boldsymbol{x})$	0	0	0
α	0	0	1
$\boldsymbol{u}^0(\boldsymbol{x},t)$	$\frac{x}{0.03} \cdot h(t)$	$\frac{x}{0.06} \cdot h(t)$	$\begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{elsewise} \end{cases}$
$\tau^{(k)}$	1.0	1.0	1.0
$\varepsilon_{\text{nonlinear}}$	10^{-10}	10^{-10}	10^{-10}
$\varepsilon_{\text{linear}}$	10^{-11}	$\rho \cdot \|F(\boldsymbol{u}^{(k)})\|_e < 10^{-10}$	10^{-12}
Remarks	Linear	Nonlinear	Linear

The test problems are defined using table 3.5.2, where the abbreviation

$h(t) := 440\sin(2\pi 50t)$ is used. Reference material for the first two test cases, which are based on the electromagnetic equation at the end of section 2.5, can be found in [14]. The first problem is a linear problem with a parabolic boundary layer for which an exact analytical solution exists. For a figure of the electromagnetic reluctivity ν, see [7]. Following [14], the magnetic vector potential applied in (2.3.1) has a frequency ω of 50Hz and an amplitude of $c = 440\text{Wbm}^{-1}$. The electric conductivity σ is taken to be a field independent constant, $5 \cdot 10^6\text{Sm}^{-1}$. The data for the reluctivity in test case 2 are determined with the use of the data in [14] and normalized for a conductivity equal to 1.

Example 3 has a shock moving in time. It was chosen to demonstrate that large time-slabs – with moving shocks – can be handled efficiently. One of the differences between examples 1 and 3 is that for $\nu \equiv \nu_{\min} \downarrow 0$ there will appear a parabolic boundary layer along Γ_2 in the former case, whereas there is a shock inside the domain in the latter example. Another difference is that problem 3 has a flow field $\hat{\boldsymbol{b}}$ such that $\nabla\cdot\hat{\boldsymbol{b}} > 0$. Problems 1 and 2 have a flow field where $\nabla\cdot\hat{\boldsymbol{b}} = 0$. Therefore, only for problem 3 additional exponential weighting is used. Note that this problem is not covered by the provided theory, because the boundary conditions γ can not be extended to a function in $\mathrm{H}^1(Q)$

For a given problem, grid, and linear solver, table 3.5.3 shows the total number of non-linear iterations, with the number of linear iterations specified for each non-linear step. The Euclidian norms of the residuals of the initial solutions $\boldsymbol{u}^0$ on the first fime-slab are 0.25, 8.27 and 0.55 for tests 1 – 3. The linear solvers are accelerated with an ILU preconditioner, see for instance [4]. More information on these solvers can be found in chapter 8. The number of iterations for the third test is reasonable since there is very little diffusion. For $\alpha \downarrow$ it has been observed that there is no convergence of the iterative solvers. For $\alpha \to \infty$ however, the number of iterations decreases rapidly for increasing α. This is caused by the scaling with $\mathrm{e}^{-\alpha t}$.

The '*' in table 3.5.3 indicates that the streamline upwind finite element basis functions have been used, contrary to the standard nodal ones, which are supposed to be the default choice. Figure 3.1 shows the grid $Q_1^{(10)}$ used for the first case. Because there is a parabolic boundary

Table 3.5.3 Numerical results.

Prob.	Grid	Lin. Solver	# Iterations
1*	$Q_1^{(10)}$	GCGLS	1:39
	$Q_5^{(10)}$	GCGLS	1:39
	$Q_1^{(10)}$	CGS	1:29
	$Q_5^{(10)}$	CGS	1:29
	$Q_1^{(10)}$	CGSTAB	1:27
	$Q_5^{(10)}$	CGSTAB	1:27
1	$Q_1^{(10)}$	GCGLS	1:112
	$Q_1^{(10)}$	CGS	1: 89
	$Q_1^{(10)}$	CGSTAB	1: 83
2*	$Q_1^{(10)}$	CGS	4:4,18,31,42
	$Q_5^{(10)}$	CGS	4:4,29,31,36
3*	$Q_1^{(12)}$	CGS	1:276

layer along Γ_2, the grid is only fine in a small area along this boundary. The equidistant levels of the SUPG solution as well as the SUPG solution itself are shown in fig. 3.2 and 3.4, for the first time-slab and fifth time-slab respectively. Note that this solution almost behaves oscillatory on the latter time-slab.

Figures 3.5-3.10 show the refined grid and the SUPG solution for the second test case, in which there is no layer involved. Note that the grid is partially refined over a larger area because the electromagnetic field penetrates further into the material.

For problem 3, the grid $Q_1^{(12)}$ on the first time-slab and the equidistant levels of the computed solution are shown in figures 3.11 and 3.12. Magnifications of this grid and the computed solution thereon can be found in figs. 3.13 resp. 3.14. Analogous to grid generation for the problems 1 and 2, the grids on each time-slab have been constructed following the adaptive refinement procedure given in section 5.9, where other numerical examples can be found.

3.6 Other global finite element applications

The time-slabbing technique may also be used for the solution of *delay differential equations* , i.e., equations of the type

$$\begin{aligned} -\nabla_{\boldsymbol{x}}\cdot(E\underline{\nabla}_{\boldsymbol{x}}u(\boldsymbol{x},t)) + \hat{\boldsymbol{b}}^T\underline{\nabla}u(\boldsymbol{x},t) + u(\boldsymbol{x},t-\triangle t) &= f(\boldsymbol{x},t) \text{ in } Q \\ u(\boldsymbol{x},t) &= u_0(\boldsymbol{x},t) \text{ in } Q_0 \\ u(\boldsymbol{x},t) &= u_c(t) \text{ at } \Gamma_c \end{aligned} \tag{3.6.1}$$

with $Q_0 = \Omega \times (-\triangle t, 0)$ analogous to (3.2.2). In this case the Jacobian matrix of the corresponding variational formulation in (3.3.6) will have an additional term on every time-slab (arguments $(\boldsymbol{x}, t)$ are omitted where possible)

$$\int_Q w(\boldsymbol{x}, t-\triangle t)(v + \delta\hat{\boldsymbol{b}}^T\underline{\nabla}v)\, d\boldsymbol{x}dt \quad \forall_{v,w\in H^1(Q)} \tag{3.6.2}$$

which vanishes on the finite element subspace $\mathcal{H}$ if the triangulation is such that all triangles have longest edge less than $\triangle t$.

The global finite element techniques can be applied as well to partial differential equations which depend on a *single parameter.* As an example consider the equation

$$\begin{aligned} -(\epsilon(1+u_x^2(x,\lambda))^{-1/2}u(x,\lambda)_x)_x + \lambda u(x,\lambda) &= 0 \text{ in } Q \\ u(x,\lambda) &= g \text{ at } \Gamma \end{aligned} \tag{3.6.3}$$

for a small positive scalar ϵ. Here streamline upwind techniques analogous to the time-slabbing techniques introduced earlier can be applied to the two-dimensional (x, λ) space. Note that this approach differs completely from the *path following* solution technique, which is a semi-discrete solution method.

Another manner to obtain possibly better discretization error estimates is to transform the differential equation (3.2.1) by a variable transformation, where e.g., the solution u is substituted by $u \cdot g$, for some weighing function g.

3.7 Conclusions

The efficiency of using finite elements in both time and space, where the time-space domain is considered as a whole regarding the use of finite elements, has been demonstrated. The method is applicable also in multi-dimensional problems where one can for instance use tetrahedron elements. As could be seen, the stability of time-stepping on the larger time-slabs is an immediate consequence of the positive definiteness of the Jacobian matrix. The use of ordinary continuous finite element approximations enables the use of standard finite element packages for the time-space domain. Adaptive refinement of an initial grid on each time-slab in order to locate and fit steep gradients is advisable and is presently studied by one of the authors.

The solution of the linear systems can be performed quite cheaply. Using still more efficient preconditioners, for instance those based on incomplete factorization or domain decomposition (see [5] and [8]), one can get methods for which the computational effort is not larger than about proportional to the number of grid points. This means that one can get savings in the computational effort of orders of magnitude compared to standard time-stepping methods, even if moving grid strategies are used, when problems with local layers are solved.

Finally, note that one could alternatively have used higher order elements instead of piecewise linear finite elements with obvious minor modifications done in the above presentation. This would have led to a faster rate of convergence of the discrete approximations for the solution of the partial differential equation.

3.8 References

[1] Axelsson O., *Finite element methods for convection-diffusion problems,* **in** Numerical Treatment of Differential Equations, (Strehmel K. ed.) Leipzig: Teubner 1988 (Teubner-Texte zur Mathematik; Bd. 104), 171-182 [Proceedings of the Fourth Seminar "Numdiff-4", Halle, 1987]

[2] Axelsson O., *On global convergence of iterative methods,* **in** Iterative Solution of Nonlinear Systems of Equations, 1-19 LNM#953, (Ansorge R., Meis Th. and Törnig W. eds.), Springer Verlag, 1982

[3] Axelsson O., *The numerical solution of partial differential equations,* **in** Mathematics and Computer Science II: fundamental contributions in the Netherlands since 1945 (Hazewinkel M., Lenstra J.K. and Meertens L. eds.), 1-18, North-Holland 1986

[4] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[5] Axelsson O., Eijkhout V., Polman B. and Vassilevski P., *Iterative solution of singular pertubation* 2^{nd} *order boundary value problems by use of incomplete block-factorization methods,* BIT, 29(1989), 867-889

[6] Axelsson O. and Maubach J., *A time-space finite element discretization technique for the calculation of the electromagnetic field in ferromagnetic materials,* Journal for Numerical Methods in Engineering, 29(1989), 2085-2111

[7] Axelsson O. and Maubach J., *A time space finite element method for nonlinear convection diffusion problems,* **in** Notes on Numerical Fluid Mechanics, (Hackbush W. and Rannacher R. eds.) Vol. 30, 6-23, Vieweg, Braunschweig, 1990 [Proceedings of the Fifth GAMM-Seminar, Kiel, West Germany 1989]

[8] Axelsson O. and Polman B., *A robust preconditioner based on algebraic substructuring and two-level grids,* **in** Robust Multi-Grid Methods (Hackbusch W. ed.), Notes on Numerical Fluid Mechanics, Vol. 23, 1-26, Vieweg, BraunSchweig, 1988

[9] Axelsson O. and Verwer J.G., *Boundary value techniques for initial value problems in ordinary differential equations,* Mathematics of Computation, 45(1985), 153-171

[10] Ciarlet P.G., *The Finite Element Method for Elliptic Problems,* North-Holland Publ., Amsterdam, 1978

[11] Hughes T.J.R. and Hulbert M., *Space-time finite element methods for elastodynamics: formulation and error estimates,* Computer Methods in Applied Mechanics and Engineering, 66(1988), 339-363

[12] Jamet P., *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain,* SIAM Journal on Numerical Analysis, 15(1978), 912-928

[13] Johnson C. and Pitkäranta J., *An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation,* Report MAT-

A215, Institute of Mathematics, Helsinki University of Technology, Helsinki, Finland, 1984

[14] Karlsson K.-E. and Wolfbrandt A., *An explicit technique for calculating the electromagnetic field and power losses in ferromagnetic materials,* internal report 721-83, department for electrical analysis methods ASEA, Västerås, Sweden, 1983

[15] Lesaint P. and Raviart P.A., *On a finite element method for solving the neutron transport equation,* **in** Mathematical Aspects of Finite Elements in Partial Differential Equations (de Boor C. ed.), 89-123, Academic Press, New York, 1974

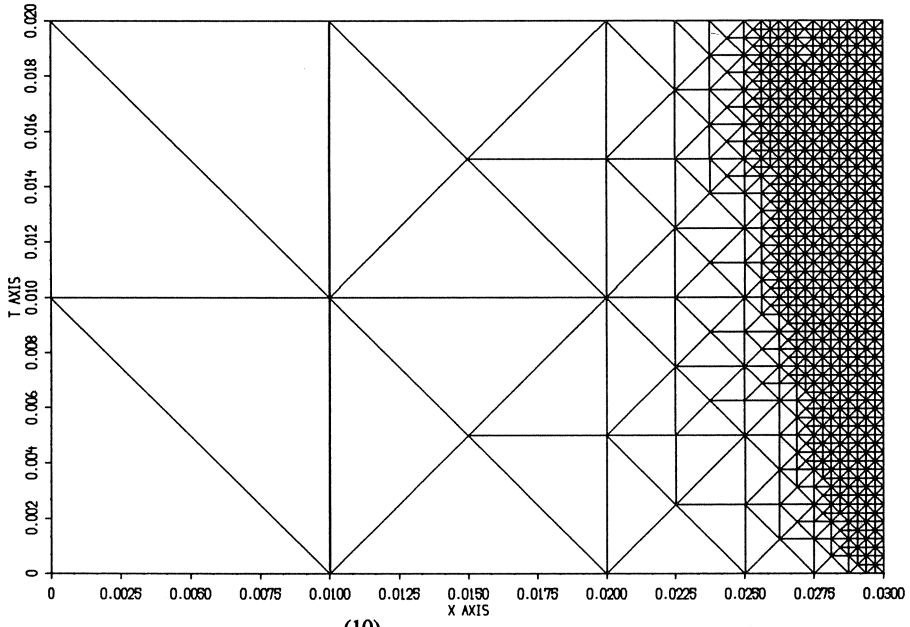


Fig. 3.1 Test 1. The grid $\mathcal{Q}_1^{(10)}$ on the first time-slab.

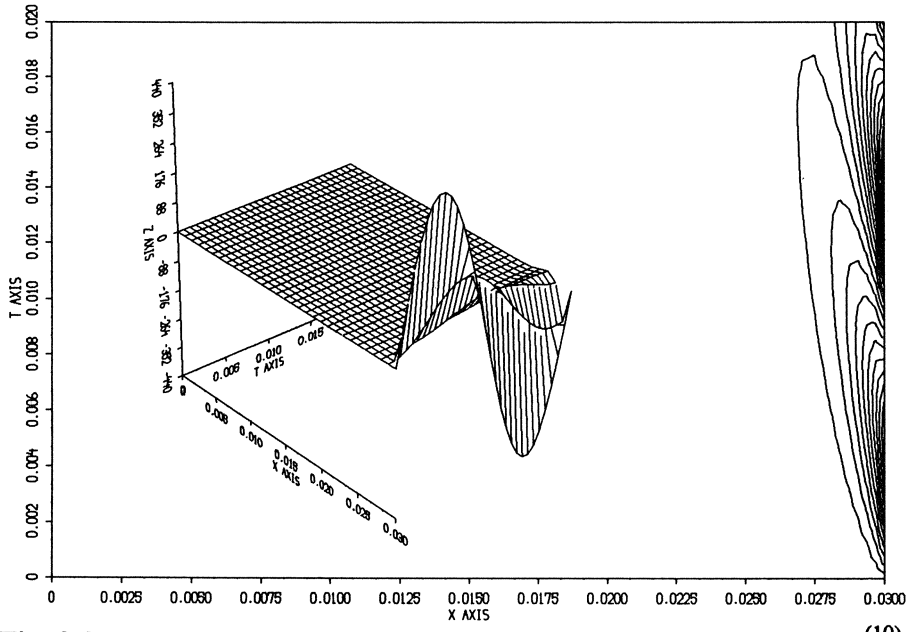


Fig. 3.2 Test 1. The isoclines of and the SUPG solution on the grid $\mathcal{Q}_1^{(10)}$.

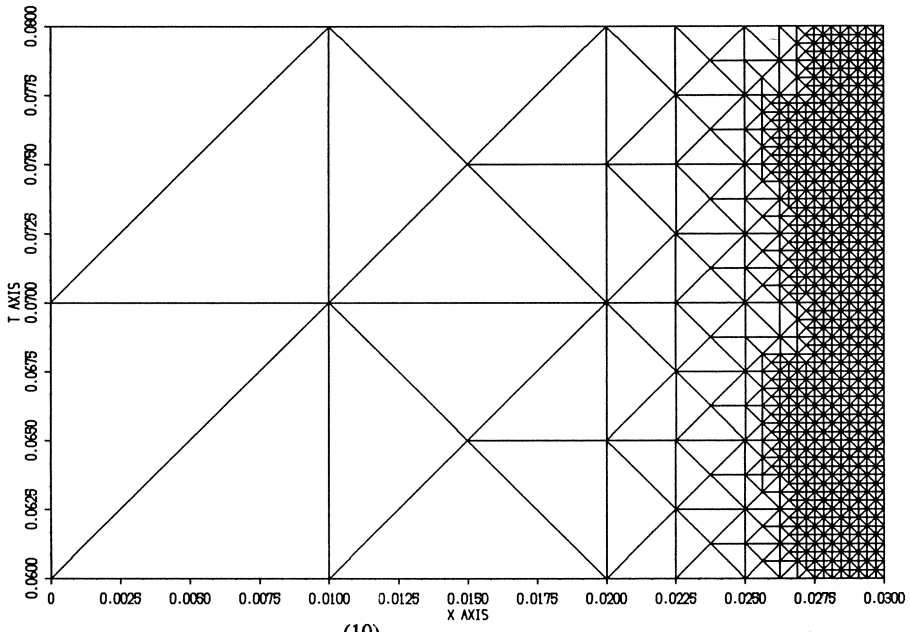


Fig. 3.3 Test 1. The grid $\mathcal{Q}_5^{(10)}$ on the fifth time-slab.

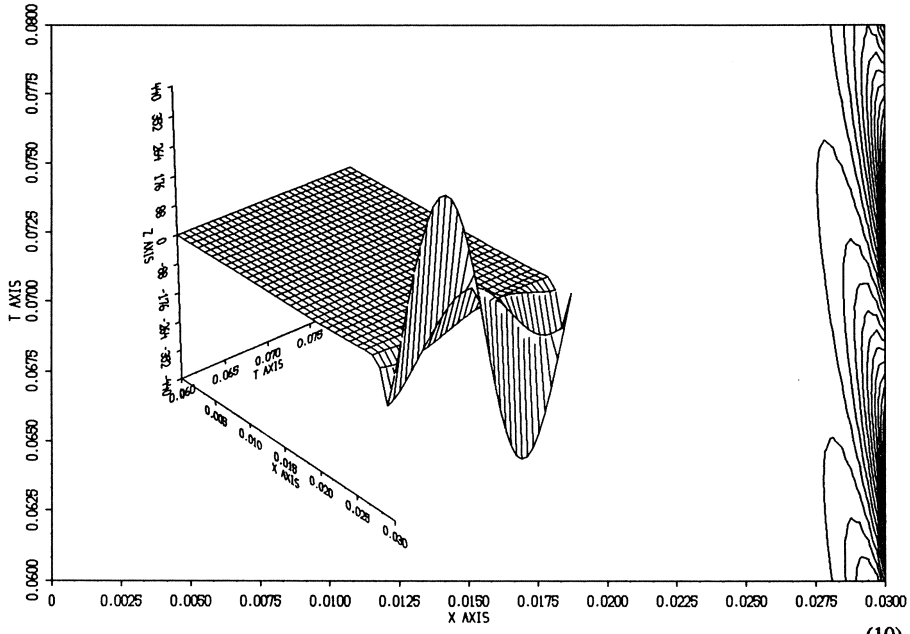


Fig. 3.4 Test 1. The isoclines of and the SUPG solution on the grid $\mathcal{Q}_5^{(10)}$.

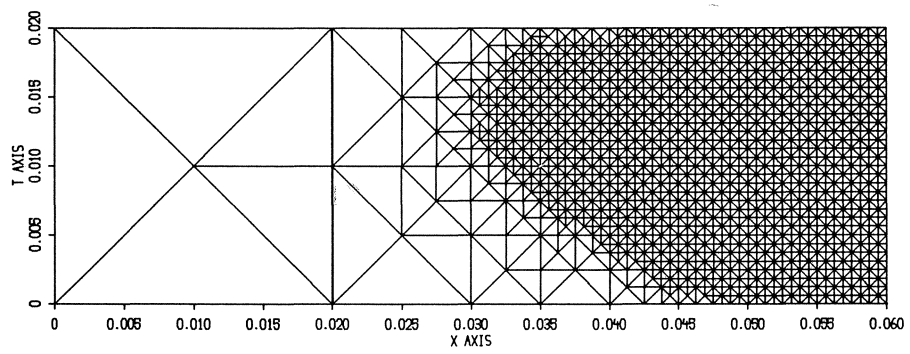


Fig. 3.5 Test 2. The grid $\mathcal{Q}_1^{(10)}$ on the first time-slab.

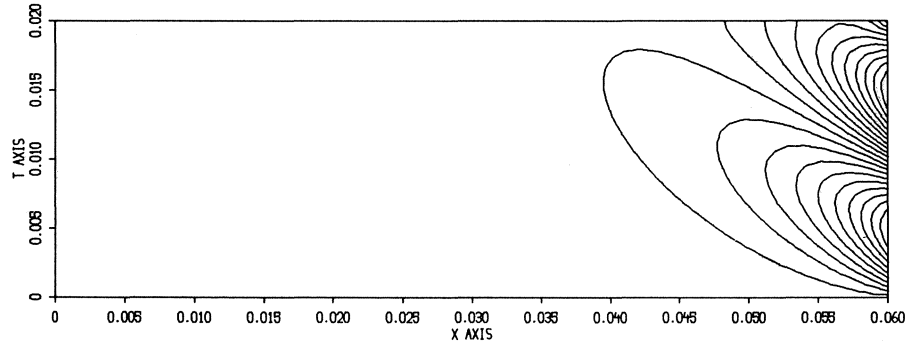


Fig. 3.6 Test 2. The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(10)}$.

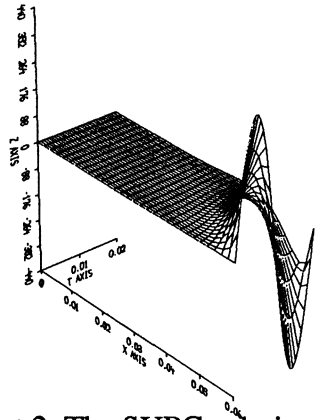


Fig. 3.7 Test 2. The SUPG solution on the grid $\mathcal{Q}_1^{(10)}$.

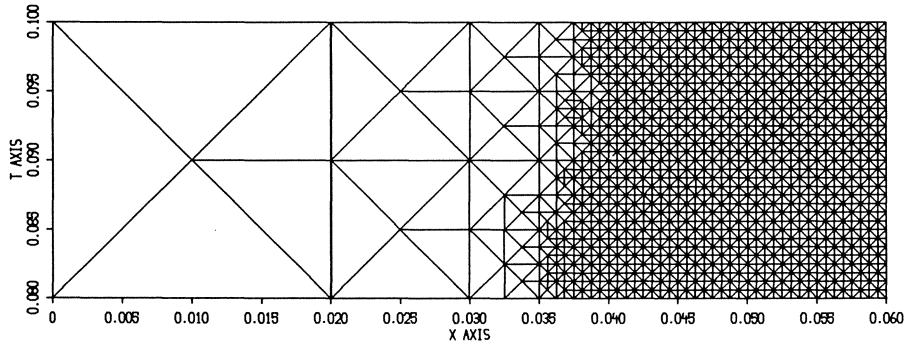


Fig. 3.8 Test 2. The grid $\mathcal{Q}_5^{(10)}$ on the fifth time-slab.

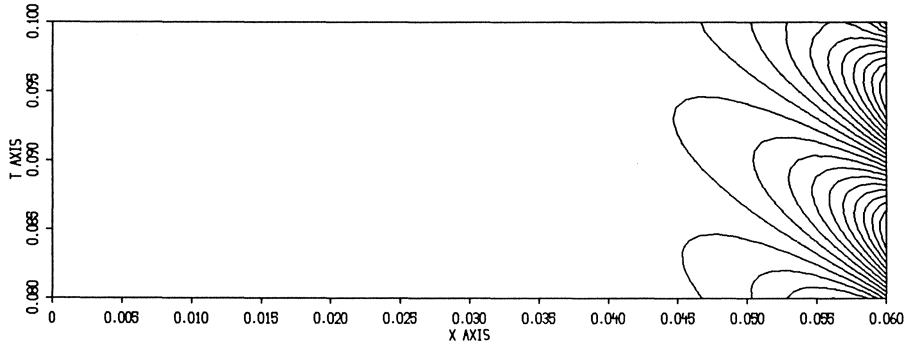


Fig. 3.9 Test 2. The isoclines of the SUPG solution on the grid $\mathcal{Q}_5^{(10)}$.

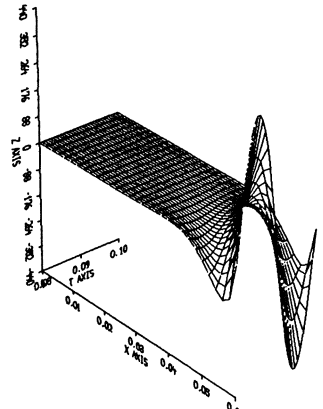


Fig. 3.10 Test 2. The SUPG solution on the grid $\mathcal{Q}_5^{(10)}$.

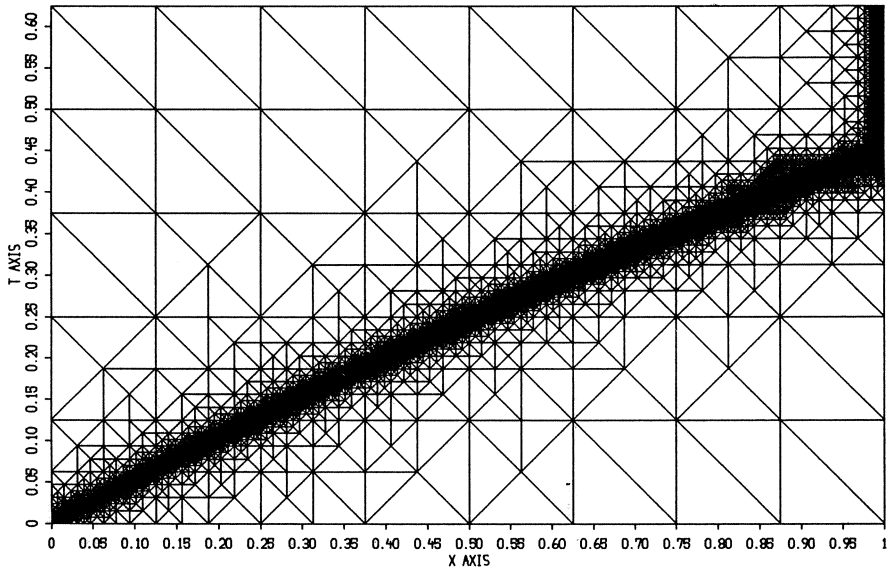


Fig. 3.11 Test 3. The grid $\mathcal{Q}_1^{(12)}$ on the first time-slab.

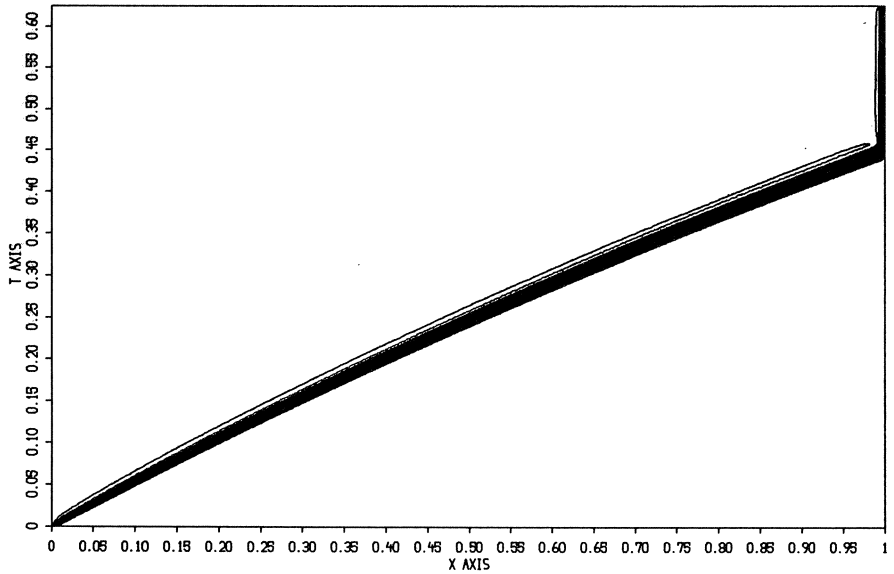


Fig. 3.12 Test 3. The isoclines of the exponentially weighted SUPG solution on $\mathcal{Q}_1^{(12)}$.

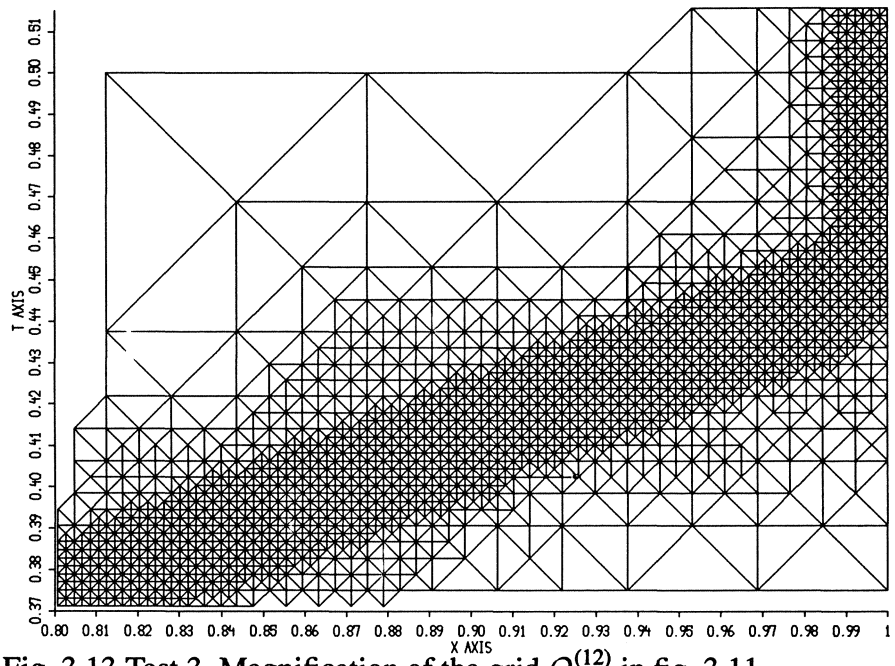


Fig. 3.13 Test 3. Magnification of the grid $\mathcal{Q}_1^{(12)}$ in fig. 3.11.

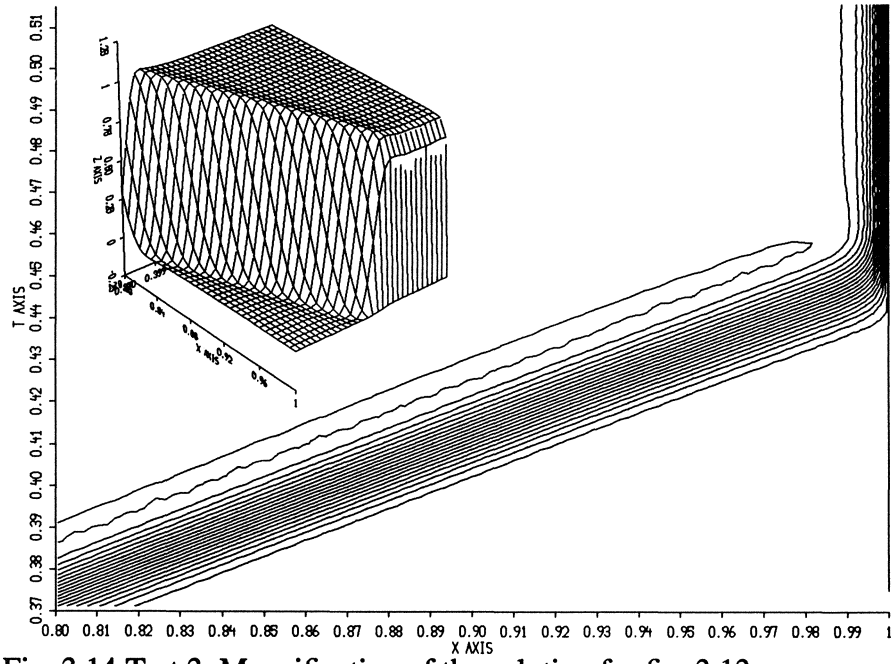


Fig. 3.14 Test 3. Magnification of the solution for fig. 3.13.

4 The Stokes system of differential equations

The section on the continuous time-slabbing technique for the Stokes problem and the section on the two-level hierarchical bases are part of the report: Axelsson O. and Maubach, J., Stability and high order approximation of monotone evolution equations valid for unbounded time by continuous time slabbing methods, Internal report of the Supercomputer Computations Research Institute, Florida State University, Tallahassee, U.S.A., submitted to SIAM Journal on Numerical Analysis. All other sections including the abstract are from: Layton W. and Maubach J., Space-time finite element methods for fluid flow problems I. The basic theory for discontinuous Galerkin methods, Preliminary report of the Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA. 15260, U.S.A. 1990.

Abstract

A time-space finite element procedure for the solution of the linear incompressible Stokes equations is studied. This procedure uses the so-called time-slabbing methodology as well as discontinuous Galerkin ideas for moving from one time-slab to the following one. An analysis of the error in the method for both the flow-field and the pressure is given. For the latter, an inf-sup condition of the time-space finite element spaces is required.

Key words: Time-space finite elements, Error bounds, Nonlinear parabolic differential equations
AMS(MOS) subject classifications: 65M05, 65M60, 65M15

4.1 Introduction

In this chapter the numerical solution via time-space finite element methods of the time-dependent Stokes problem is considered. The algorithm to be considered is the, so-called, discontinuous Galerkin method introduced for parabolic problems in Jamet [12], [13], who also considers moving boundaries. The ultimate goal is to solve Navier-Stokes problems using unstructured and adaptive time-space grids which incorporate a priori and/or a posteriori knowledge of the solution behavior. This point of view involves an evolution of discontinuous Galerkin methods into discontinuous time-slabbing procedures, of which the continuous variant is extensively used in Maubach [18] and Axelsson and Maubach [1], [2]. In fact, the succes of this approach for parabolic problems when coupled with appropriate data structures and fast solvers on each time-slab, suggests that the approach can be extended to fluid flow problems.

In section 4.2 the continuous global time-space finite element method is considered. After that, in section 4.3, the discontinuous procedure for the linear Stokes problem is studied. Stability and an error estimate for the flow field are proved in propositions 4.3.1 and 4.3.2. Next, two reinterpretations of the celebrated inf-sup condition are examined, a local and a global one. An error bound for the pressure valid under both is given. At this early stage in the development of time-space Galerkin methods for Navier-Stokes problems, the inf-sup condition which occurs in them, is not as completely understood as the one which occurs for the stationary problem. There are some interesting complications which arise from both the time-space formulation and the discontinuous Galerkin formulation. In section 4.4, some tensor product spaces are presented which satisfy the inf-sup condition arising for the *continuous time-slabbing* Galerkin methods, since in this case the condition is more standard and relatively easy to be satisfied. The construction of spaces which are based on unstructured grids and which satisfy the inf-sup conditions (4.4.4) and (4.5.2) is still an open problem. The related time-space inf-sup condition arising for the *discontinuous time-slabbing* Galerkin methods is analyzed in section 4.5. In this section also examples of tensor product spaces, which satisfy the conditions are provided. Finally, in section 4.6 the use of a finite element hierarchical basis for the continuous time-space formulation is considered.

For early work on Galerkin methods in time for initial value problems, see Hulme [10], [11] for ordinary differential equations, and Jamet [12], [13] for parabolic problems. There has been quite a bit of work on this topic recently, especially for parabolic problems. For a representative selection, see Johnson [14], French [6], Aziz and Monk [3] as well as the previously cited references. Apparently, there has been little effort in the direction of validating these methods for fluid flow problems – a challenge undertaken here. Interesting extensions of this work include coupled non-isothermal flows, see e.g. Boland and Layton [4] or [5], flow problems with moving boundaries, time-space adaptivity and resolution of issues related to the time-space inf-sup condition.

4.2 The Stokes problem

This section studies the numerical solution of the time-dependent Stokes problem by the use of global time-space finite element methods. The goal is to solve the problem on a fixed time-slab using adaptive hierarchically refined grids. Stability and error estimates for the flow field are proved first, and after considering the inf-sup condition on a time-slab an error bound valid for the pressure is obtained. Thereafter the use of a two-level hierarchical finite element basis is considered for the computation of the flow field. It will be shown in section 4.6 that each time the computational grid is refined, one can obtain a cheap initial approximation for the flow field by splitting this into two parts.

In order to present the global time-space formulation for problems with constraints, consider the Stokes problem (the Navier-Stokes equations can be handled analogously). Consider $\Omega \in \mathbb{R}^2$, an open bounded connected polygonal domain which defines the time-slab $Q = \Omega \times (0, t_J]$ and boundary $\partial\Omega$. Assume that Q has a smooth boundary. The solution $\boldsymbol{u}, p$ of the Stokes problem satisfies

$$\begin{aligned}
\boldsymbol{u}_t - \nu\triangle\boldsymbol{u} + \underline{\nabla} p &= \boldsymbol{f} && \text{in } Q \\
\nabla\cdot\boldsymbol{u} &= 0 && \text{in } Q \\
\boldsymbol{u} &= 0 && \text{at } \Gamma_c \\
\boldsymbol{u}(\boldsymbol{x}, 0) &= \boldsymbol{u}_0(\boldsymbol{x}) && \text{at } \Gamma_1 \\
\textstyle\int_\Omega p(\boldsymbol{x}, t)\, d\boldsymbol{x} &= 0 && \text{for } t \in (0, t_J] \text{ a.e.,}
\end{aligned} \tag{4.2.1}$$

where the two-dimensional vectorial function $\boldsymbol{u} = [u_1, u_2]^T$ is such that $u_1, u_2 \in \mathrm{C}^0(\overline{Q}) \cap \{\mathrm{C}^2(\Omega) \times \mathrm{C}^1((0,t_J])\}$ and $p \in \mathrm{C}^0(\overline{Q}) \cap \{\mathrm{C}^1(\Omega) \times \mathrm{C}^0((0,t_J])\}$, if $\boldsymbol{f}$ and $\boldsymbol{u}_0$ are sufficiently smooth functions. Here $\Gamma_c = \{(\boldsymbol{x},t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \partial\Omega \wedge t \in [0,t_J]\}$, and $\Gamma_1 = \{(\boldsymbol{x},t) \in \mathbb{R}^{n+1} : \boldsymbol{x} \in \Omega \wedge t = 0\}$. This equation has a unique solution (see e.g. [15] and [17]). In order to obtain an approximation of the solution $\boldsymbol{u}, p$ above define $\Gamma_D = \Gamma_1 \cup \Gamma_c$ and a new set of Hilbert spaces

$$V = \{\boldsymbol{v} \in \left[\mathrm{H}^1(\Omega) \times \mathrm{L}^2((0,t_J])\right]^2 : \boldsymbol{v}(\boldsymbol{x},t) = 0 \text{ at } \Gamma_D\}$$

$$M = \{q \in \mathrm{L}^2(\Omega) \times \mathrm{L}^2((0,t_J]) : \int_Q q \, d\boldsymbol{x}dt = 0\},$$

which will be used here and in the remainder of this section. Now the variational formulation of (4.2.1) is: find $\hat{\boldsymbol{u}} \in V$ and $p \in M$ such that

$$\begin{aligned} &\int_Q \left[\hat{\boldsymbol{u}}_t \boldsymbol{v} + \nu \underline{\nabla} \hat{\boldsymbol{u}} : \underline{\nabla} \boldsymbol{v} + p \nabla \cdot \boldsymbol{v} - \boldsymbol{f}\boldsymbol{v}\right] d\boldsymbol{x}dt = 0 \quad \forall_{\boldsymbol{v} \in V} \\ &\int_Q q \nabla \cdot \hat{\boldsymbol{u}} \, d\boldsymbol{x}dt = 0 \quad \forall_{q \in M} \end{aligned} \tag{4.2.2}$$

since the homogeneous Dirichlet boundary conditions in (4.2.1) cause the solution $\hat{\boldsymbol{u}}$ to be an element of V. Here, $\underline{\nabla}\boldsymbol{u} : \underline{\nabla}\boldsymbol{v} = \underline{\nabla}u_1^T \underline{\nabla} v_1 + \underline{\nabla}u_2^T \underline{\nabla} v_2$ is defined as usual for all $u = [u_1, u_2]^T \in V$ and all $v = [v_1, v_2]^T \in V$. Further, for the construction of finite element spaces assume that $Q \subset \mathbb{R}^3$ has been divided into tetrahedral elements. Then, choose finite element subspaces $\mathcal{V} \subset \left[\mathrm{H}^1(Q)\right]^2 \subset V$ and $\mathcal{M} \subset M$ such that

$$\boldsymbol{v} \in \mathcal{V} \Rightarrow \int_Q q \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x}dt = 0 \quad \forall_{q \in \mathcal{M}} \tag{4.2.3}$$

Several possible choices of finite element spaces which satisfing this relationship exist. Consider as an example $\mathcal{V}$ the set of piecewise linear basis functions defined at the vertices of the tetrahedrons, and $\mathcal{M}$, the space spanned by the basis functions that are piecewise constant per tetrahedron, and that satisfy the constraint $\int_Q q \, d\boldsymbol{x}dt = 0$. For other choices, see [8]. The discrete Galerkin global time-space solution $\hat{\boldsymbol{u}}_h, p_h \in \mathcal{V} \times \mathcal{M}$ is now a solution of

$$\int_Q \left[(\hat{\boldsymbol{u}}_h)_t \boldsymbol{v} + \nu \underline{\nabla} \hat{\boldsymbol{u}}_h : \underline{\nabla} \boldsymbol{v} - \boldsymbol{f}\boldsymbol{v}\right] d\boldsymbol{x}dt = 0 \quad \forall_{\boldsymbol{v} \in \mathcal{V}} \tag{4.2.4}$$

since condition (4.2.3) implies that $\int_Q q\nabla\cdot\hat{\boldsymbol{u}}_h \, d\boldsymbol{x}dt = 0$ for all $q \in \mathcal{M}$ automatically.

In order to show that the discrete solution is bounded $L^2(\Omega)$ and in time, the following norms will be needed. For $\boldsymbol{u} \in V$ define

$$|||\boldsymbol{u}|||_0 = \left\{ \int_Q |\boldsymbol{u}(\boldsymbol{x},t)|^2 \, d\boldsymbol{x}dt \right\}^{1/2}, \quad |||\boldsymbol{u}|||_1 = |||\underline{\nabla}\boldsymbol{u}|||_0$$

and

$$|||\boldsymbol{u}|||_Q = \left\{ \int_\Omega \boldsymbol{u}(\boldsymbol{x},t_J)^2 \, d\boldsymbol{x} + \nu|||\boldsymbol{u}|||_1^2 \right\}^{1/2}.$$

On the set M, $|||\cdot|||_0$ will be used as a norm. Note that the gradient $\underline{\nabla}$ always denotes the space derivatives, it does not include the time derivatives. The time-space gradient will be denoted by $\underline{\nabla}_{x,t}$ and to avoid confusion sometimes $\underline{\nabla}$ is denoted by $\underline{\nabla}_x$.

First it is shown that $|||\hat{\boldsymbol{u}}_h|||_Q$ of the flow field $\hat{\boldsymbol{u}}_h$ is bounded in time-space. Since there are non-slip homogeneous Dirichlet boundary conditions one may substitute $\boldsymbol{v} = \hat{\boldsymbol{u}}_h \in \mathcal{V}$ in (4.2.4) leading to

$$\int_Q \left[(\hat{\boldsymbol{u}}_h)_t\hat{\boldsymbol{u}}_h + \nu\underline{\nabla}\hat{\boldsymbol{u}}_h : \underline{\nabla}\hat{\boldsymbol{u}}_h - \boldsymbol{f}\hat{\boldsymbol{u}}_h \right] d\boldsymbol{x}dt = 0 \Rightarrow$$

$$\tfrac{1}{2}\int_\Omega \hat{\boldsymbol{u}}_h^2(\boldsymbol{x},t_J)\, d\boldsymbol{x} - \tfrac{1}{2}\int_\Omega \hat{\boldsymbol{u}}_h^2(\boldsymbol{x},0)\, d\boldsymbol{x} + \nu|||\hat{\boldsymbol{u}}_h|||_1^2 = \int_Q \boldsymbol{f}\hat{\boldsymbol{u}}_h \, d\boldsymbol{x}dt$$

implying that

$$|||\hat{\boldsymbol{u}}_h|||_Q^2 \le \frac{c}{\nu}|||\boldsymbol{f}|||_0^2 + \int_\Omega \hat{\boldsymbol{u}}_h^2(\boldsymbol{x},0)\, d\boldsymbol{x}, \tag{4.2.5}$$

as there exists a scalar $c > 0$ such that $|||\boldsymbol{v}|||_0^2 \le c|||\boldsymbol{v}|||_1^2$ for all $\boldsymbol{v} \in V$. Hence, if there exists a positive scalar c such that

$$\operatorname*{ess\,sup}_{x\in\Omega} \left\{ \int_0^\infty |\boldsymbol{f}(\boldsymbol{x},t)|^2 \, dt \right\} < c$$

then the finite element solution $\hat{\boldsymbol{u}}_h$ is bounded for all time $t > 0$. In view of (1.4.2), time-slabbing for subsequent time-slabs will lead to a good approximation of the solution $\hat{\boldsymbol{u}}$ of (4.2.2) if the discretization error $|||\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}_h|||_Q$ is bounded as in (1.6.6).

In order to show this, let $\boldsymbol{\zeta} \in \mathcal{V}$ be chosen arbitrary and set $\boldsymbol{\eta} = \hat{\boldsymbol{u}} - \boldsymbol{\zeta}$, $\boldsymbol{\varphi} = \hat{\boldsymbol{u}}_h - \boldsymbol{\zeta}$ and $\boldsymbol{\theta} = \boldsymbol{\eta} - \boldsymbol{\varphi}$. Now subtracting the continuous (4.2.2) and discrete (4.2.4) formulations for an arbitrary $q \in \mathcal{M}$ leads to the following error equation

$$\int_Q \left[\boldsymbol{\varphi}_t \boldsymbol{v} + \nu \underline{\nabla}\boldsymbol{\varphi} : \underline{\nabla}\boldsymbol{v}\right] \, d\boldsymbol{x}dt = \int_Q \left[\boldsymbol{\eta}_t \boldsymbol{v} + \nu \underline{\nabla}\boldsymbol{\eta} : \underline{\nabla}\boldsymbol{v} + (p-q)\nabla\cdot\boldsymbol{v}\right] \, d\boldsymbol{x}dt$$

for arbitrary $\boldsymbol{v} \in \mathcal{V}$. Setting $\boldsymbol{v} = \boldsymbol{\varphi} \in \mathcal{V}$ leads to

$$\begin{aligned} &\tfrac{1}{2}\int_\Omega \boldsymbol{\varphi}^2(\boldsymbol{x}, t_J) \, d\boldsymbol{x} + \nu |||\boldsymbol{\varphi}|||_1^2 \\ &= \int_Q \left[\boldsymbol{\eta}_t \boldsymbol{\varphi} + \nu \underline{\nabla}\boldsymbol{\eta} : \underline{\nabla}\boldsymbol{\varphi} + (p-q)\nabla\cdot\boldsymbol{\varphi}\right] \, d\boldsymbol{x}dt + \tfrac{1}{2}\int_\Omega \boldsymbol{\varphi}^2(\boldsymbol{x}, 0) \, d\boldsymbol{x} \,. \end{aligned}$$

Under the assumption that $\hat{\boldsymbol{u}}_h(\boldsymbol{x}, 0) = \hat{\boldsymbol{u}}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x})$, the terms in the right-hand side of the latter equation can be estimated above by

$$\int_\Omega \boldsymbol{\varphi}^2(\boldsymbol{x}, 0) \, d\boldsymbol{x} = \int_\Omega (\hat{\boldsymbol{u}}_h - \boldsymbol{\zeta})^2(\boldsymbol{x}, 0) \, d\boldsymbol{x} = \int_\Omega \boldsymbol{u}_0^2(\boldsymbol{x}) \, d\boldsymbol{x}$$

and

$$\begin{aligned} &\int_Q \left[\boldsymbol{\eta}_t \boldsymbol{\varphi} + \nu \underline{\nabla}\boldsymbol{\eta} : \underline{\nabla}\boldsymbol{\varphi} + (p-q)\nabla\cdot\boldsymbol{\varphi}\right] \, d\boldsymbol{x}dt \leq \\ &\frac{1}{c_1\nu}|||\boldsymbol{\eta}_t|||_0^2 + \frac{c_1\nu}{4}|||\boldsymbol{\varphi}|||_0^2 + \nu|||\boldsymbol{\eta}|||_1^2 + \frac{\nu}{4}|||\boldsymbol{\varphi}|||_1^2 + c_2|||p-q|||_0^2 \leq \\ &\frac{1}{c_1\nu}|||\boldsymbol{\eta}_t|||_0^2 + \nu|||\boldsymbol{\eta}|||_1^2 + \frac{(1+c_1)\nu}{4}|||\boldsymbol{\varphi}|||_1^2 + c_2|||p-q|||_0^2 \end{aligned}$$

for some positive scalars c_1 and c_2 (for instance, on $\mathcal{V}$ $c_1|||\cdot|||_0^2 \leq |||\cdot|||_1^2$). This leads to

$$\tfrac{1}{2}|||\boldsymbol{\varphi}|||_Q^2 \leq \frac{1}{c_1\nu}|||\boldsymbol{\eta}_t|||_0^2 + \nu|||\boldsymbol{\eta}|||_1^2 + c_2|||p-q|||_0^2 + \tfrac{1}{2}\int_\Omega \boldsymbol{u}_0^2(\boldsymbol{x}) \, d\boldsymbol{x} \quad (4.2.6)$$

and via relations $\sqrt{a^2+b^2} \leq a+b$ and $a^2 \leq b^2+ad \Rightarrow a \leq b+d$ for positive $a, b, d \in \mathbb{R}$, one gets

$$|||\varphi|||_Q \leq c\left\{\sqrt{\nu^{-1}}|||\eta_t|||_0^2 + \sqrt{\nu}|||\eta|||_1^2 + |||p-q|||_0^2 + \int_\Omega u_0^2(x)\,dx\right\},$$

for some generic positive scalar c. This finally leads to the relation

$$\begin{aligned}|||\theta|||_Q^2 \leq & 2|||\eta|||_Q^2 + 2|||\varphi|||_Q^2 \\ \leq & c\left\{\nu|||\eta|||_1^2 + \frac{1}{\nu}|||\eta_t|||_0^2 + |||p-q|||_0^2 + \right. \\ & \left. \int_\Omega \eta^2(x,t_J)\,dx + \int_\Omega u_0^2(x)\,dx\right\}\end{aligned}$$

for all $\zeta = \hat{u} - \eta \in \mathcal{V}$ and all $q \in \mathcal{M}$ whence

$$\begin{aligned}|||\hat{u}-\hat{u}_h|||_Q^2 \leq & c \inf_{v\in\mathcal{V}}\left\{\int_\Omega (\hat{u}-v)^2(x,t_J)\,dx + \nu|||\hat{u}-v|||_1^2\right\} + \\ & c \inf_{v\in\mathcal{V}}\left\{\frac{1}{\nu}|||\hat{u}_t - v_t|||_0^2\right\} + \\ & c \inf_{q\in\mathcal{M}}\left\{|||p-q|||_0^2\right\} + c\int_\Omega u_0^2(x)\,dx\,.\end{aligned}$$

Using the *Nečas trace inequality* (see [19], page 84) one can estimate $\int_\Omega (\hat{u}-v)^2(x,t_J)\,dx$ above by $|||\nabla_{x,t}(\hat{u}-v)|||_0$, leading to

$$\begin{aligned}|||\hat{u}-\hat{u}_h|||_Q^2 \leq & c \inf_{v\in\mathcal{V}}\left\{|||\hat{u}-v|||_1^2 + |||\hat{u}_t - v_t|||_0^2\right\} + \\ & c \inf_{q\in\mathcal{M}}\left\{|||p-q|||_0^2\right\} + c\int_\Omega u_0^2(x)\,dx\,.\end{aligned} \tag{4.2.7}$$

Using for example piecewise linear basis functions for $\mathcal{V}$ on a grid of tetrahedrons with maximal diameter h leads to

$$|||\hat{u}-\hat{u}_h|||_Q^2 \leq ch + \inf_{q\in\mathcal{M}}\left\{|||p-q|||_0^2\right\} \tag{4.2.8}$$

if $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{u}}_t$ are sufficiently regular. However, it should be noted that the combination of elementwise linear functions for the flow and elementwise constant functions for the pressure is unstable. Note that one may get similar error estimates if other finite element spaces are used, for example tensor product finite element spaces. For the pressures p and p_h one can obtain a similar error bound, provided that a non-standard inf-sup condition holds for $\mathcal{X}$, $\mathcal{M}$

$$|||p - p_h|||_0^2 \leq c \left\{ \inf_{q \in \mathcal{M}} \left\{ |||p - q|||_0^2 \right\} + \nu |||\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}_h|||_1^2 + |||(\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}_h)_t|||_0^2 \right\}$$

where c is again a positive scalar. Note that the error bound for the pressure variable involves the time derivative of the flow field error. The error bound (4.2.5) does not provide an estimate in this case, but the use of streamline upwind or duality techniques could provide a bound including this term, analogous to (4.2.5). The bound is obtained using an *inf-sup* condition, claiming that there exist a constant β such that

$$\inf_{q \in \mathcal{M}} \sup_{v \in \mathcal{X}} \left\{ \frac{\int_Q q \nabla \cdot \boldsymbol{v} \, d\boldsymbol{x}dt}{|||\boldsymbol{v}|||_Q \cdot |||q|||_0} \right\} \geq \beta > 0 .$$

See theorem 4.4.1, for the derivation of such a bound for the continuous global time-space time-slabbing case. Summarizing one can conclude that it is possible to use time-slabbing for the solution of the Stokes equations. Section 4.6 will comment on some of the advantages of this technique and indicate how it can be combined with local refinement and hierarchically defined finite element basis in order to save computational effort.

4.3 The discontinuous approach for the Stokes problem

This section studies the discontinuous time-slabbing numerical solution of the Stokes problem by using time-space finite element methods (see e.g. [9]). Here $\Omega \in \mathbb{R}^n$, $n = 2, 3$ is an open, bounded, connected and polygonal domain which defines $Q = \Omega \times (0, t_J]$ and the boundary $\partial\Omega$. For the simplicity of exposition, assume that $n = 2$ in the sequel, and that

the solution $\boldsymbol{u}, p$ of the Stokes problem (4.2.1) is sufficiently smooth. In order to give a discontinuous time-space Galerkin variational formulation for this equation, some definitions have to be introduced. Let $0 = 0 < t_1 < \ldots < t_J$ be a partitioning of $(0, t_J]$ used to partition the computational domain Q into time-slabs by $Q_j := \Omega \times (t_{j-1}, t_j]$. Further define as usual for $v \in C^0(Q)$

$$\boldsymbol{v}^+(\boldsymbol{x}, t) = \lim_{\varepsilon \downarrow 0} \boldsymbol{v}(\boldsymbol{x}, t+\varepsilon), \quad \boldsymbol{v}^-(\boldsymbol{x}, t) = \lim_{\varepsilon \downarrow 0} \boldsymbol{v}(\boldsymbol{x}, t-\varepsilon), \quad \boldsymbol{v}^-(\boldsymbol{x}, 0) = \boldsymbol{v}(\boldsymbol{x}, 0)$$

for all $\boldsymbol{x} \in \Omega$ and all time $t \in (0, t_J]$. For each function defined on Q let P_j denote its restriction onto Q_j and define the sets

$$\hat{X} = \{\boldsymbol{v}: Q \mapsto \mathbb{R}^n : P_j \boldsymbol{v} \in \left[C^2(\overline{Q}_j)\right]^n \wedge \boldsymbol{v} = 0 \text{ at } \partial\Omega \times (t_{j-1}, t_j]\}$$

$$\hat{M} = \{q: Q \mapsto \mathbb{R} : P_j q \in C^1(\overline{Q}_j) \wedge \int_\Omega q(\boldsymbol{x}, t)\, d\boldsymbol{x} = 0 \text{ for } t_{j-1} < t < t_j\}$$

where for a domain Q, $C^p(\overline{Q})$ is the subset of $C^p(Q)$ of those functions for which all the partial derivatives can be extended continuously to the boundary of Q. Note that elements of $\hat{X}$ and $\hat{M}$ may be discontinuous across time levels t_j for all $j = 1, \ldots, J$. Further define for $\boldsymbol{v} = [v_1, v_2]^T$ the norms

$$\begin{cases} \|\boldsymbol{v}\|^2_{L^2(Q)} = \displaystyle\int_Q |\boldsymbol{v}|^2 \, d\boldsymbol{x}dt = \int_Q v_1^2 + v_2^2 \, d\boldsymbol{x}dt \\ \|\boldsymbol{v}(\boldsymbol{x}, t)\|^2_{L^2(\Omega)} = \displaystyle\int_\Omega |\boldsymbol{v}(\boldsymbol{x}, t)|^2 \, d\boldsymbol{x} \\ |||\boldsymbol{v}|||^2_{j,-} = \nu \|\underline{\nabla} \boldsymbol{v}\|^2_{L^2(Q_j)} + \|\boldsymbol{v}^-(\boldsymbol{x}, t_j)\|^2_{L^2(\Omega)} \\ |||\boldsymbol{v}|||^2_{-} = \displaystyle\sum_{j=1}^{J} \nu \|\underline{\nabla} \boldsymbol{v}\|^2_{L^2(Q_j)} + \max_{j=1,\ldots,J} \{\|\boldsymbol{v}^-(\boldsymbol{x}, t_j)\|^2_{L^2(\Omega)}\} \end{cases}$$

for all $j = 1, \ldots, J$, where $\underline{\nabla} \boldsymbol{v} = [\underline{\nabla} v_1, \underline{\nabla} v_2]^T$. Note that the norm $|||\cdot|||_-$ is almost a Sobolev 1 norm $H^1(Q)$ on Q, it only lacks the time derivative and has an additional L^2 contribution.

Now let X and M be the closure of $\hat{X}$ and $\hat{M}$ under the norms $|||\cdot|||_-$ resp. $\|\cdot\|_{L^2(Q)}$, let $X_j = P_j X$ and $M_j = P_j M$ and consider the variational formulation of the Stokes problem. First note that the continuous

solution $\boldsymbol{u}, p$ of (4.2.1) is also a solution of

$$\begin{aligned} &\int_Q \boldsymbol{u}_t \boldsymbol{v} + \nu \underline{\nabla} \boldsymbol{u} : \underline{\nabla} \boldsymbol{v} + p \nabla \cdot \boldsymbol{v} - \boldsymbol{f} \boldsymbol{v} \, d\boldsymbol{x} dt = 0 \quad \forall_{\boldsymbol{v} \in X} \\ &\int_\Omega q \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) \, d\boldsymbol{x} = 0 \quad \forall_{q \in M}, \text{ for a.e. } 0 < t \leq t_J \end{aligned} \tag{4.3.1}$$

under the given initial and boundary conditions as in (4.2.1) where as usual $\underline{\nabla} \boldsymbol{u} : \underline{\nabla} \boldsymbol{v} = \underline{\nabla} u_1^T \underline{\nabla} v_1 + \underline{\nabla} u_2^T \underline{\nabla} v_2$. Since the test functions $\boldsymbol{v} \in X$ and $q \in M$ are allowed to be discontinuous across time-slabs and since $\boldsymbol{u}^+(\boldsymbol{x}, t) = \boldsymbol{u}^-(\boldsymbol{x}, t)$ for all $(\boldsymbol{x}, t) \in Q$ for smooth enough data (see [21]), the classical solution $\boldsymbol{u}, p$ is also a solution to

$$\begin{aligned} &\int_{Q_j} \boldsymbol{u}_t \boldsymbol{v} + \nu \underline{\nabla} \boldsymbol{u} : \underline{\nabla} \boldsymbol{v} + p \nabla \cdot \boldsymbol{v} - \boldsymbol{f} \boldsymbol{v} \, d\boldsymbol{x} dt \\ &+ \int_\Omega (\boldsymbol{u}^+ - \boldsymbol{u}^-) \boldsymbol{v}^+(\boldsymbol{x}, t_{j-1}) \, d\boldsymbol{x} = 0 \quad \forall_{\boldsymbol{v} \in X_j} \\ &\int_\Omega q \nabla \cdot \boldsymbol{u}(\boldsymbol{x}, t) \, d\boldsymbol{x} = 0 \quad \forall_{q \in M_j}, \text{ for a.e. } t_{j-1} < t \leq t_j \end{aligned} \tag{4.3.2}$$

for all $j = 1, \ldots, J$. In this chapter $\hat{\boldsymbol{u}}, p$ will denote a solution of (4.2.1) in the weak sense of (4.3.2).

Now, suppose that $\hat{\boldsymbol{u}}_h, p_h$ are the discrete solutions of the same equations where X_j and M_j are substituted by finite element subspaces $\mathcal{X}_j = P_j \mathcal{X}$ and $\mathcal{M}_j = P_j \mathcal{M}$, for $j = 1, \ldots, J$. In order to decouple the system of equations for the flow-field $\hat{\boldsymbol{u}}$ and the pressure p above, consider the introduction of the subspace $V \subset X$ of divergence free functions, defined by

$$V = \{\boldsymbol{v} \in X : \int_\Omega q \nabla \cdot \boldsymbol{v}(\boldsymbol{x}, t) \, d\boldsymbol{x} = 0 \quad \forall_{q \in M}, \text{ for a.e. } 0 < t \leq t_J\}$$

and a discrete subset $\mathcal{V} \subset \mathcal{X}$ by

$$\mathcal{V} = \{\boldsymbol{v} \in \mathcal{X} : \int_\Omega q \nabla \cdot \boldsymbol{v}(\boldsymbol{x}, t) \, d\boldsymbol{x} = 0 \quad \forall_{q \in \mathcal{M}}, \text{ for a.e. } 0 < t \leq t_J\}.$$

This space $\mathcal{V}$ is only approximate divergence free, relative to the space $\mathcal{M}$. Note that $0 \in \mathcal{V}$, but without further knowledge it is not clear if $\mathcal{V}$ is any larger.

Now setting $\hat{u}_h, p_h$ in $\mathcal{V}$ resp. $\mathcal{M}$ to be a solution of equation (4.3.2), this system of equations reduces to

$$\int_{Q_j} (\hat{u}_h)_t v + \nu \underline{\nabla} \hat{u}_h : \underline{\nabla} v - f v \, dx dt = - \int_\Omega (\hat{u}_h^+ - \hat{u}_h^-) v^+(x, t_{j-1}) \, dx \quad (4.3.3)$$

for all $v \in \mathcal{V}_j$ and all time-slabs Q_j.

For the derivation of the error estimates in the sequel, the following *dual norms* are useful

$$\|f\|_{\star,h} = \nu^{-\frac{1}{2}} \sup_{v \in \mathcal{V}} \frac{\int_Q f v \, dx dt}{\|\underline{\nabla} v\|_{L^2(Q)}} \text{ and } \|f\|_{\star,h,j} = \nu^{-\frac{1}{2}} \sup_{v \in \mathcal{V}_j} \frac{\int_{Q_j} f v \, dx dt}{\|\underline{\nabla} v\|_{L^2(Q_j)}} .$$

Note that since $\mathcal{V} \not\subset V$, $\|\cdot\|_{\star,h}$ is an exterior approximation of the $V^\star$ norm. Exploiting these norms one can derive

Lemma 4.3.1 *For all $q \in L^2(Q)$ and all $f \in [L^2(Q)]^2$ there exist positive scalars c such that*

$$\begin{cases} \|u\|^2_{L^2(Q)} \le c \|\underline{\nabla} u\|^2_{L^2(Q)} \\ \|u\|^2_{L^2(Q)} \le c \nu^{-1} |||u|||^2_- \\ \|f\|^2_{\star,h} \le c \nu^{-1} \|f\|^2_{L^2(Q)} \\ \left| \int_Q q \nabla \cdot v \, dx dt \right| \le c \|q\|_{L^2(Q)} \|\underline{\nabla} v\|_{L^2(Q)} . \end{cases}$$

for all $u \in X$.

Proof. Consider the first two statements. Let $u \in \hat{X}$, whence $u(x,t) \in [H^1(\Omega)]^2$ for almost every time $t \in (0, t_J]$. From a Friedrichs inequality it follows that

$$\int_\Omega |u(x,t)|^2 \, dx \le c \int_\Omega |\underline{\nabla} u(x,t)|^2 \, dx$$

holds at almost every time t since $\boldsymbol{u} = 0$ at $\partial\Omega$. Here c is positive and independent of the time since Ω is not varying with time, whence integration with respect to the time variable leads to

$$\|\boldsymbol{u}\|^2_{\mathrm{L}^2(Q)} \leq c\|\underline{\nabla}\boldsymbol{u}\|^2_{\mathrm{L}^2(Q)}$$

and consequently to

$$\|\boldsymbol{u}\|^2_{\mathrm{L}^2(Q)} \leq c\nu^{-1}|||\boldsymbol{u}|||^2_{-} .$$

The third statement can be verified using

$$\begin{aligned}\|\boldsymbol{f}\|_{\star,h} &= \nu^{-\frac{1}{2}} \sup_{\boldsymbol{v}\in\mathcal{V}}\{\frac{\int_Q \boldsymbol{f}\boldsymbol{v}\, d\boldsymbol{x}dt}{\|\underline{\nabla}\boldsymbol{v}\|_{\mathrm{L}^2(Q)}}\} \leq \nu^{-\frac{1}{2}} \sup_{\boldsymbol{v}\in\mathcal{V}}\{\frac{\|\boldsymbol{f}\|_{\mathrm{L}^2(Q)}\|\boldsymbol{v}\|_{\mathrm{L}^2(Q)}}{\|\underline{\nabla}\boldsymbol{v}\|_{\mathrm{L}^2(Q)}}\} \\ &= c\nu^{-\frac{1}{2}}\|\boldsymbol{f}\|_{\mathrm{L}^2(Q)} .\end{aligned}$$

Note that by construction one has got

$$\int_Q \boldsymbol{f}\boldsymbol{u}\, d\boldsymbol{x}dt \leq \|\boldsymbol{f}\|_{\star,h} \cdot \nu^{\frac{1}{2}}\|\underline{\nabla}\boldsymbol{u}\|_{\mathrm{L}^2(Q)} \leq \|\boldsymbol{f}\|_{\star,h} \cdot |||\boldsymbol{u}|||_{-}$$

for all $\boldsymbol{u} \in \mathcal{V}$ and all $\boldsymbol{f} \in \mathrm{L}^2(Q)$. The last statement holds since

$$\int_Q |\nabla\cdot\boldsymbol{v}|^2\, d\boldsymbol{x}dt = \int_Q ((v_1)_x + (v_2)_y)^2\, d\boldsymbol{x}dt \leq 2\|\underline{\nabla}\boldsymbol{v}\|^2_{\mathrm{L}^2(Q)}$$

for all $\boldsymbol{v} = [v_1, v_2]^T \in X$. □

Now consider the stability of the discrete solution. To this end it will be proved that $|||\hat{\boldsymbol{u}}_h|||_{-}$ is bounded above independent of the time $t_J > 0$ and the partitioning of $(0, t_J]$ used.

Theorem 4.3.1 *The discrete solution is bounded by the initial value and source function as follows*

$$\|\underline{\nabla}\hat{\boldsymbol{u}}_h\|^2_{\mathrm{L}^2(Q)} + \max_{j=1,\ldots,J}\{\|\hat{\boldsymbol{u}}_h^-(\boldsymbol{x}, t_j)\|^2_{\mathrm{L}^2(\Omega)}\} \leq 2\|\boldsymbol{f}\|^2_{\star,h} + 2\int_\Omega |\boldsymbol{u}_{0,h}(\boldsymbol{x})|^2\, d\boldsymbol{x} .$$

Proof. Since $\hat{u}_h \in \mathcal{V}$ substitution of $v = \hat{u}_h$ in (4.3.3) in combination with

$$\int_{Q_j} (\hat{u}_h)_t \hat{u}_h \, dx dt = - \int_{Q_j} \hat{u}_h (\hat{u}_h)_t \, dx dt + \int_\Omega (\hat{u}_h^-(x, t_j))^2 \, dx - \int_\Omega (\hat{u}_h^+(x, t_{j-1}))^2 \, dx$$

$$\int_{Q_j} (\hat{u}_h)_t \hat{u}_h \, dx dt = \tfrac{1}{2} \int_\Omega (\hat{u}_h^-(x, t_j))^2 \, dx - \tfrac{1}{2} \int_\Omega (\hat{u}_h^+(x, t_{j-1}))^2 \, dx$$

leads to

$$\int_{Q_j} (\hat{u}_h)_t \hat{u}_h + \nu \underline{\nabla} \hat{u}_h : \underline{\nabla} \hat{u}_h - f \hat{u}_h \, dx dt + \int_\Omega (\hat{u}_h^+ - \hat{u}_h^-) \hat{u}_h^+(x, t_{j-1}) \, dx = 0$$

or equivalently

$$\tfrac{1}{2} \int_\Omega (\hat{u}_h^-(x, t_j))^2 \, dx - \tfrac{1}{2} \int_\Omega (\hat{u}_h^+(x, t_{j-1}))^2 \, dx + \int_\Omega (\hat{u}_h^+(x, t_{j-1}))^2 \, dx + \nu \| \underline{\nabla} \hat{u}_h \|^2_{L^2(Q_j)} = \int_{Q_j} f \hat{u}_h \, dx dt + \int_\Omega \hat{u}_h^-(x, t_{j-1}) \hat{u}_h^+(x, t_{j-1}) \, dx .$$

Using the relations

$$\begin{cases} \displaystyle\int_{Q_j} f \hat{u}_h \, dx dt \le \tfrac{1}{2} \| f \|^2_{\star, h, j} + \tfrac{1}{2} \nu \| \underline{\nabla} \hat{u}_h \|^2_{L^2(Q_j)} \\ \displaystyle\int_\Omega \hat{u}_h^-(x, t_{j-1}) \hat{u}_h^+(x, t_{j-1}) \, dx \le \tfrac{1}{2} \int_\Omega (\hat{u}_h^-(x, t_{j-1}))^2 \, dx + \tfrac{1}{2} \int_\Omega (\hat{u}_h^+(x, t_{j-1}))^2 \, dx \end{cases}$$

Recombining terms, one gets

$$\tfrac{1}{2} \nu \| \underline{\nabla} \hat{u}_h \|^2_{L^2(Q_j)} + \tfrac{1}{2} \int_\Omega (\hat{u}_h^-(x, t_j))^2 \, dx \le \tfrac{1}{2} \| f \|_{\star, h, j} + \tfrac{1}{2} \int_\Omega (\hat{u}_h^-(x, t_{j-1}))^2 \, dx$$

for each time-slab. Summation over all time-slabs leads to

$$|||\hat{u}_h|||_-^2 \leq \|f\|_{\star,h}^2 + \int_\Omega |\hat{u}_h(x,0)|^2 \, dx .$$

Similarly, summation over all time-slabs yields

$$\tfrac{1}{2}\|\hat{u}_h^-(x,t_j)\|_{L^2(\Omega)}^2 \leq \tfrac{1}{2}\|f\|_{\star,h}^2 + \tfrac{1}{2}\int_\Omega |\hat{u}_h(x,0)|^2 \, dx .$$

Thus, as t_j is arbitrary, $\max_{j=1,\ldots,J}\{\|\hat{u}_h^-(x,t_j)\|_{L^2(\Omega)}^2\}$ is bounded in a similar way. Adding this $l^\infty(L^2(\Omega))$ bound to the previous $L^2(H^1(\Omega))$ bound yields the result. □

An error estimate for the velocity field is obtained for the time-space formulation by standard techniques. These can be refined to include pressure estimates under a suitably generalized inf-sup condition, to be considered in section 4.4.

Theorem 4.3.2 *If the boundary of the domain and the data are smooth enough then*

$$\begin{aligned}|||\hat{u}-\hat{u}_h|||_-^2 \leq 2 \inf_{v\in\mathcal{V}} \Big\{ & c\|(\hat{u}-v)_t\|_{\star,h}^2 + 2\nu\|\underline{\nabla}(\hat{u}-v)\|_{L^2(Q)}^2 + \\ & \int_\Omega (\hat{u}-v)^2(x,t_J)\,dx + \int_\Omega (u_0(x)-v(x,0))^2\,dx + \\ & \frac{c}{\nu}\sum_{j=1}^{J}\int_\Omega (v^+-v^-)(\hat{u}_h-v)^+(x,t_{j-1})\,dx \Big\} + \\ 2 \inf_{q\in\mathcal{M}} & \Big\{\frac{c}{\nu}\|p-q\|_{L^2(Q)}^2\Big\} + \int_\Omega (u_0(x)-u_{0,h}(x))^2\,dx\end{aligned}$$

where the summation term vanishes if $v = \eta \in \mathcal{V}$ is substituted.

Proof. Choose $\zeta \in \mathcal{V}$ arbitrary and write

$$\begin{cases} \theta = \hat{u} - \hat{u}_h & \text{the discretization error} \\ \eta = \hat{u} - \zeta & \text{the (to be) interpolation error} \\ \varphi = \hat{u}_h - \zeta & \text{the difference of the previous two} \end{cases}$$

Subtracting the variational formulation (4.3.3) from (4.3.2) leads for arbitrary $q \in \mathcal{M}$ to

$$\begin{aligned} &\int_{Q_j} \theta_t v + \nu \underline{\nabla} \theta : \underline{\nabla} v + (p-q) \nabla \cdot v \; dx dt + \\ &\int_{\Omega} (\theta^+ - \theta^-) v^+(x, t_{j-1}) \; dx = 0 \end{aligned} \tag{4.3.4}$$

for all $v \in \mathcal{V}$. Since this equation is linear with respect to the argument θ setting $\theta = \eta - \varphi$ and choosing $v = \varphi \in \mathcal{V}$ leads to

$$\begin{aligned} &\int_{Q_j} \varphi_t \varphi + \nu \underline{\nabla} \varphi : \underline{\nabla} \varphi + \int_{\Omega} (\varphi^+ - \varphi^-) \varphi^+(x, t_{j-1}) \; dx = \\ &\int_{Q_j} \eta_t \varphi + \nu \underline{\nabla} \eta : \underline{\nabla} \varphi + \int_{\Omega} (\eta^+ - \eta^-) \varphi^+(x, t_{j-1}) \; dx \; dx dt + \\ &\int_{\Omega} ((^+ - (^-) p^+(x, t_{j-1}) \; dx - q) \nabla \cdot \varphi \; dx dt \, . \end{aligned}$$

According to lemma 4.3.1 there exists a positive scalar c such that

$$\left| \int_{Q_j} \eta_t \varphi \; dx dt \right| \leq c \| \eta_t \|_{\star,h,j} \cdot \nu^{\frac{1}{2}} \| \underline{\nabla} \varphi \|_{\mathrm{L}^2(Q_j)}$$

$$\left| \int_{Q_j} \nu \underline{\nabla} \eta : \underline{\nabla} \varphi \; dx dt \right| \leq \nu^{\frac{1}{2}} \| \underline{\nabla} \eta \|_{\mathrm{L}^2(Q_j)} \cdot \nu^{\frac{1}{2}} \| \underline{\nabla} \varphi \|_{\mathrm{L}^2(Q_j)}$$

$$\left| \int_{Q_j} (p-q) \nabla \cdot \varphi \; dx dt \right| \leq \nu^{-\frac{1}{2}} \| p - q \|_{\mathrm{L}^2(Q_j)} \cdot c \nu^{\frac{1}{2}} \| \underline{\nabla} \varphi \|_{\mathrm{L}^2(Q_j)}$$

$$\left| \int_{\Omega} (\eta^+ - \eta^-) \varphi^+(x, t_{j-1}) \; dx \right|^2 \leq \int_{\Omega} (\eta^+ - \eta^-)^2 (x, t_{j-1}) \; dx \cdot \int_{\Omega} (\varphi^+(x, t_{j-1}))^2 \; dx \, .$$

Note that the first estimate above only holds for $\varphi \in \mathcal{V}$ according to the definition of the dual norm. Treating the terms involving the $\varphi^+(x, t_{j-1})$

and $\varphi^-(x, t_{j-1})$ as in the previous lemma now yields

$$\nu\|\underline{\nabla}\varphi\|^2_{L^2(Q_j)} + \tfrac{1}{2}\int_\Omega |\varphi^-(x,t_j)|^2\,dx \le$$
$$\tfrac{1}{2}\int_\Omega |\varphi^-(x,t_{j-1})|^2\,dx + c\nu^{-\frac{1}{2}}\int_\Omega (\eta^+ - \eta^-)\varphi^+(x,t_{j-1})\,dx +$$
$$\nu^{\frac{1}{2}}\|\underline{\nabla}\varphi\|_{L^2(Q_j)} \cdot \left\{ c\|\eta_t\|_{\star,h,j} + \nu^{\frac{1}{2}}\|\underline{\nabla}\eta\|_{L^2(Q_j)} + c\nu^{-\frac{1}{2}}\|p-q\|_{L^2(Q_j)} \right\}$$

for each time-slab. Addition over time-slabs $j = 1, \ldots, s \le J$ leads to

$$\frac{\nu}{2}\sum_{j=1}^{s}\|\underline{\nabla}\varphi\|_{L^2(Q_j)} + \tfrac{1}{2}\|\varphi^-(x,t_s)\|_{L^2(\Omega)} \le$$
$$\tfrac{1}{2}\int_\Omega (\varphi(x,0))^2\,dx + \tfrac{1}{2}c\|\eta_t\|^2_{\star,h} + \tfrac{1}{2}\nu\|\underline{\nabla}\eta\|^2_{L^2(Q)} + \tfrac{1}{2}\frac{c}{\nu}\|p-q\|^2_{L^2(Q)} +$$
$$\tfrac{1}{2}\frac{c}{\nu}\sum_{j=2}^{J}\int_\Omega (\eta^+ - \eta^-)\varphi^+(x,t_{j-1})\,dx\,.$$

If $\zeta = \hat{u}_I$, the interpolant of the solution $\hat{u}$ to (4.3.2), then $\eta^+(x,t_{j-1}) = \eta^-(x,t_{j-1})$ if the solution $\hat{u} \in C^0(\overline{Q})$ and if the finite element bases used on each $\mathcal{V}_j$ are such that $\eta^+(x,t_{j-1}) = \eta^-(x,t_{j-1})$ on each time t_j, $j = 1, \ldots, J$. The first assumption will be satisfied if the boundary of the domain is smooth enough and if the initial and source function data are small and smooth enough, the latter will be satisfied using a finite element space $\mathcal{V}$ for the whole domain Q and taking for $\mathcal{V}_j$ the restriction of this space to Q_j. This means that the last term in the equation above will vanish if we choose $\zeta = \hat{u}_I$. The first term on the right-hand side also vanishes if $\hat{u}_h(x,0) = u_0(x)$ is taken as a Dirichlet boundary condition on the first time-slab.

The use of the triangle inequality without the assumptions above concerning the choice of ζ now leads to

$$\begin{aligned}
|||\theta|||^2_- &\le 2|||\eta|||^2_- + 2|||\varphi|||^2_- \\
&\le 2c\|\eta_t\|^2_{\star,h} + 4\nu\|\underline{\nabla}\eta\|^2_{L^2(Q)} + \\
&\quad 2\int_\Omega (\eta^-(x,t_J))^2\,dx + 2\frac{c}{\nu}\|p-q\|^2_{L^2(Q)}
\end{aligned}$$

$$2\int_\Omega (\varphi(x,0))^2\,dx + 2\sum_{j=2}^{J}\int_\Omega (\eta^+ - \eta^-)\varphi^+(x,t_{j-1})\,dx\,.$$

Note that the integrand at $t = 0$ is bounded by

$$\|\varphi(x,0)\|_{L^2(\Omega)} \le \|u_0(x) - u_{0,h}(x)\|_{L^2(\Omega)} + \|u_0(x) - \zeta(x,0)\|_{L^2(\Omega)}\,.$$

Substituting the expressions for θ, η and φ above lead to the desired result. □

To obtain error estimates for the pressure, a suitable generalization of the inf-sup condition is imposed, to be discussed in more detail in section 4.5. The original version was due to Babuska and Brezzi. To this end introduce a slightly different triple bar norm

$$|||v|||^2_{j-1,+} = \nu\|\underline{\nabla} v\|^2_{L^2(Q_j)} + \tfrac{1}{2}\|v^+(x,t_{j-1})\|^2_{L^2(\Omega)}\,,$$

where now the integral over the lower instead of the upper time boundary of the domain Q_j is taken into account. For the moment being assume that

Assumption There exists a positive scalar β such that

$$\inf_{q\in\mathcal{M}_j}\sup_{v\in\mathcal{X}_j}\left\{\frac{\left|\int_{Q_j} q\nabla\cdot v\;dxdt\right|}{|||v|||_{j-1,+}\|q\|_{L^2(Q_j)}}\right\} \ge \beta_j > 0\,. \tag{4.3.5}$$

Remark 4.3.1 In section 4.4 several examples of $\mathcal{V}$ and $\mathcal{M}$ are constructed which satisfy this assumption above.

Lemma 4.3.2 *Suppose (4.3.5) holds, then*

(i) There exists a unique solution $\hat{u}_h, p_h$

(ii) There exists a positive scalar c *such that*

$$\inf_{v\in\mathcal{V}}\left(\|u_t - v_t\|^2_{L^2(Q_j)} + \|\underline{\nabla}(u-v)\|^2_{L^2(Q_j)}\right)$$
$$\le c\inf_{v\in\mathcal{X}}\left(\|u_t - v_t\|^2_{L^2(Q_j)} + \|\underline{\nabla}(u-v)\|^2_{L^2(Q_j)}\right)$$

for all $\boldsymbol{u} \in \mathrm{H}^1(Q_j)$.

Proof. Note that evidently the opposite of (ii) is true since $\mathcal{V} \subset \mathcal{X}$. □

Now consider the pressure error estimate.

Theorem 4.3.3 *Suppose the inf-sup condition 4.3.2 holds, then the approximate pressure* p_h *satisfies the following estimate*

$$\|p - p_h\|_{\mathrm{L}^2(Q)} \leq \frac{c}{\beta}\Bigg(\nu^{-\frac{1}{2}} \inf_{q \in \mathcal{M}}\{\|p - q\|_{\mathrm{L}^2(Q)}\} + \|\hat{\boldsymbol{u}}_t - (\hat{\boldsymbol{u}}_h)_t\|_{\star,h} + \nu^{\frac{1}{2}}\|\underline{\nabla}(\hat{\boldsymbol{u}} - \hat{\boldsymbol{u}}_h)\|_{\mathrm{L}^2(Q)} + \nu^{-\frac{1}{2}}\sum_{j=1}^{J}\int_{\Omega}(\hat{\boldsymbol{u}}_h^+ - \hat{\boldsymbol{u}}_h^-)^2(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x}\Bigg).$$

for some scalar $c > 0$ *and* $\beta = \min_{j=1,\ldots,J}\{\beta_j\}$.

Proof. Subtracting the variational equations of which $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{u}}_h$ are the solutions one obtains

$$\int_{Q_j} \boldsymbol{\theta}_t \boldsymbol{v} + \nu \underline{\nabla}\boldsymbol{\theta} : \underline{\nabla}\boldsymbol{v} + (p - p_h)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt + \int_{\Omega}(\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)\boldsymbol{v}^+(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x} = 0$$

for all $\boldsymbol{v} \in \mathcal{V}_j$. Reordering the terms in this equation and substituting $p_h - p = p_h - q + q - p$ for arbitrary $q \in \mathcal{M}$ gives us

$$\int_{Q_j}(p_h - q)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt = \int_{Q_j}(p - q)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt + \int_{Q_j}\boldsymbol{\theta}_t\boldsymbol{v} + \nu\underline{\nabla}\boldsymbol{\theta} : \underline{\nabla}\boldsymbol{v} + \int_{\Omega}(\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)\boldsymbol{v}^+(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x} \quad \forall_{\boldsymbol{v} \in V}$$

Substitution of the relations

$$\left|\int_{Q_j} \boldsymbol{\theta}_t \boldsymbol{v}\, d\boldsymbol{x}dt\right| \leq c\|\boldsymbol{\theta}_t\|_{\star,h,j} \cdot |||\boldsymbol{v}|||_{j-1,+}$$

$$\left|\int_{Q_j} \nu \underline{\nabla}\boldsymbol{\theta} : \underline{\nabla}\boldsymbol{v}\, d\boldsymbol{x}dt\right| \leq \nu^{\frac{1}{2}} \|\underline{\nabla}\boldsymbol{\theta}\|_{L^2(Q_j)} \cdot \nu^{\frac{1}{2}} |||\boldsymbol{v}|||_{j-1,+}$$

$$\left|\int_\Omega (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)\boldsymbol{v}^+(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x}\right| \leq c\nu^{-\frac{1}{2}} \sqrt{\int_\Omega (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)^2(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x}} \cdot |||\boldsymbol{v}|||_{j-1,+}$$

$$\left|\int_{Q_j} (p-q)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt\right| \leq 2\nu^{-\frac{1}{2}} \|p-q\|_{L^2(Q_j)} \cdot |||\boldsymbol{v}|||_{j-1,+}$$

in the equation above, leads to the estimate

$$\left|\int_{Q_j} (p_h - q)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt\right| \leq |||\boldsymbol{v}|||_{j-1,+} \cdot \left(2\nu^{-\frac{1}{2}} \|p-q\|_{L^2(Q_j)} + c\|\boldsymbol{\theta}_t\|_{\star,h,j} + \nu^{\frac{1}{2}} \|\underline{\nabla}\boldsymbol{\theta}\|_{L^2(Q_j)} + c\nu^{-\frac{1}{2}} \int_\Omega (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)^2(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x}\right)$$

for all $\boldsymbol{v} \in \mathcal{V}$ and all $q \in \mathcal{M}$. Dividing the equation by $|||\boldsymbol{v}|||_{j-1,+} \cdot \|p_h - q\|_{L^2(Q_j)}$, taking the supremum over all $\boldsymbol{v} \in \mathcal{V}$ and multiplying again by $\|p_h - q\|_{L^2(Q_j)}$ leads to

$$\begin{aligned}
&\|p_h - q\|_{L^2(Q_j)} \cdot \sup_{\boldsymbol{v}\in\mathcal{V}} \left\{ \frac{\left|\int_{Q_j} (p_h - q)\nabla\cdot\boldsymbol{v}\, d\boldsymbol{x}dt\right|}{|||\boldsymbol{v}|||_{j-1,+} \|p_h - q\|_{L^2(Q_j)}} \right\} \\
&\leq 2\nu^{-\frac{1}{2}} \|p-q\|_{L^2(Q_j)} + c\|\boldsymbol{\theta}_t\|_{\star,h,j} + \nu^{\frac{1}{2}} \|\underline{\nabla}\boldsymbol{\theta}\|_{L^2(Q_j)} + \\
&c\nu^{-\frac{1}{2}} \int_\Omega (\boldsymbol{\theta}^+ - \boldsymbol{\theta}^-)^2(\boldsymbol{x}, t_{j-1})\, d\boldsymbol{x}\,.
\end{aligned}$$

Now using the fact that for arbitrary functions $a, b: \mathcal{M} \mapsto \mathbb{R}$

$$\inf_{q\in\mathcal{M}} \{a(q)b(q)\} \geq \inf_{q\in\mathcal{M}} \{a(q)\} \cdot \inf_{q\in\mathcal{M}} \{b(q)\}$$

and using that the supremum over $v \in \mathcal{V}$ may be replaced by the supremum over $v \in \mathcal{X}$ (see lemma 4.3.1) one finds the existence of positive scalars β and c such that

$$\inf_{q\in\mathcal{M}}\{\|p_h - q\|_{L^2(Q_j)}\} \leq \frac{c}{\beta}\Bigg(\inf_{q\in\mathcal{M}}\{\nu^{-\frac{1}{2}}\|p - q\|_{L^2(Q_j)}\} + \|\theta_t\|_{\star,h,j} + \nu^{\frac{1}{2}}\|\underline{\nabla}\theta\|_{L^2(Q_j)} + \nu^{-\frac{1}{2}}\int_\Omega (\theta^+ - \theta^-)^2(x, t_{j-1})\, dx\Bigg).$$

Finally, a straightforward summation over all time-slabs and the triangle inequality lead to the desired result. □

Note that the summation term does not vanish, not even under the same conditions on the solution and the grid as posed in the previous theorem. Note also that this pressure error bound involves the error in the time derivative of the flow field. This can be estimated in various ways when standard finite element spaces are used. For example, note that $|||\cdot|||_+$ dominates the $L^2(Q)$ norm. Thus one can obtain (suboptimal) bounds for $\|(\hat{u} - \hat{u}_h)_t\|_{L^2(Q)}$ by using inverse estimates. Other methods, which can yield better results, involve duality techniques, following for example Aziz and Monk [3].

4.4 The continuous inf-sup condition

The inf-sup condition introduced in section 4.3 is a natural time-space analog of the steady-state version. However, the differences between the steady-state version and this present version for time-slabs introduce some new and interesting difficulties. As an introduction we first examine the local version arising in the continuous Galerkin time-slabbing formulation. Recall the classical static and local version

$$\inf_{q\in\mathcal{M}_j}\sup_{v\in\mathcal{X}_j}\{\frac{\left|\int_{Q_j} q\nabla\cdot v\, dxdt\right|}{\|\underline{\nabla}v\|_{L^2(Q_j)}\|q\|_{L^2(Q_j)}}\} \geq \beta_j > 0. \qquad (4.4.1)$$

Naturally, it is desirable that this condition be satisfied in the limit in the case where $h \to 0$. That is, for the case where the infimum over $q \in \hat{M}$ and the supremum over $v \in \hat{X}$ is taken. To this end, consider the following theorem

Theorem 4.4.1 *Let Ω be, as before, a connected and polygonal domain. Then*

$$\inf_{q \in \hat{M}_j} \sup_{v \in \hat{X}_j} \{ \frac{\left| \int_{Q_j} q \nabla \cdot v \, dx dt \right|}{\|\underline{\nabla} v\|_{L^2(Q_j)} \|q\|_{L^2(Q_j)}} \} \geq \beta_j > 0, \tag{4.4.2}$$

for a constant $\beta_j > 0$ independent of j.

Proof. Choose $q \in \hat{M}_j$ fixed. Since $q \in \hat{M}_j$, for each time $t \in (t_{j-1}, t_j]$ the function q satisfies the relation $\int_\Omega q(x,t)\, dx = 0$ implying $q \in L^2_0(\Omega)$ for almost every time $t \in (t_{j-1}, t_j]$. As $L^2_0(\Omega)$ is precisely the range of the divergence operator acting on $\left[H^1_0(\Omega)\right]^2$ (see Girault and Raviart [8], corollary 2.4, p. 24, or Girault and Raviart [7]) there is a $v \in \left[H^1_0(\Omega)\right]^2$, unique modulo the addition of a divergence free element of $\left[H^1_0(\Omega)\right]^2$, with $\nabla \cdot v = q$. Further, there exists a positive scalar c such that

$$\|\underline{\nabla} v\|_{L^2(\Omega)} \leq c \|q\|_{L^2(\Omega)}.$$

Now, in order to verify (4.4.2) define $v_q(x,t)$ by the above procedure for almost every $t \in (t_{j-1}, t_j]$. Since

$$\int_{t_{j-1}}^{t_j} \|\underline{\nabla} v_q\|^2_{L^2(\Omega)} \, dt \leq c^2 \int_{t_{j-1}}^{t_j} \|q\|^2_{L^2(\Omega)} \, dt$$

it follows that

$$\frac{\left| \int_{Q_j} q \nabla \cdot v_q \, dx dt \right|}{\|\underline{\nabla} v_q\|_{L^2(Q_j)} \|q\|_{L^2(Q_j)}} \geq \frac{\int_{Q_j} q^2 \, dx dt}{c \|q\|^2_{L^2(Q_j)}} = c^{-1} > 0.$$

This holds for v_q as constructed from which

$$\sup_{v \in \hat{X}_j} \{ \frac{\left| \int_{Q_j} q \nabla \cdot v \, dx dt \right|}{\|\underline{\nabla} v\|_{L^2(Q_j)} \|q\|_{L^2(Q_j)}} \} \geq c^{-1} > 0$$

and (4.4.2) follow, since this is valid for every $q \in \hat{M}_j$. □

Now consider the problem of verifying the discrete inf-sup condition arising in the continuous Galerkin method. It is shown that this condition holds, for example, when the finite element space approximating the pressure and each component of the velocity are tensor products provided that the spatial factors satisfy the classical condition. One observation which facilitates the following proof is that if $\mathcal{M}_j \times \tilde{X}_j$ satisfies an inf-sup condition for a certain intermediate space $\tilde{X}_j$ and if $\mathcal{X}_j$ contains $\tilde{X}_j$ then $\mathcal{M}_j \times \mathcal{X}_j$ does so too.

Let $(t_{j-1}, t_j]$ be the j-th time interval and let $\mathcal{A} = \mathcal{A}(t_{j-1}, t_j]$, and $\hat{\mathcal{A}} = \hat{\mathcal{A}}(t_{j-1}, t_j]$ be finite element spaces in $\mathrm{H}^1((t_{j-1}, t_j])$. Furhter, let $\mathcal{B}$, $\hat{\mathcal{B}}$ be finite element spaces, defined over Ω such that $\mathcal{B} \subset \mathrm{L}^2_0(\Omega)$ and $\hat{\mathcal{B}} \subset \mathrm{H}^1_0(\Omega)$ and define

$$\mathcal{M}_j = \mathcal{B} \times \mathcal{A}$$
$$\mathcal{X}_j = \left[\hat{\mathcal{B}} \times \hat{\mathcal{A}}\right]^2 .$$

The intermediate space $\tilde{X}_j \subset X_j$, where $\tilde{X}_j = \{v \in \mathcal{X}_j : v = v(x)b(t)\}$, will be helpful in the analysis of inf-sup condition on the pair $\mathcal{M}_j \times \mathcal{X}_j$. Considering the condition upon $\mathcal{M} \times \tilde{X}_j$ and substituting $q = q(x)a(t) \in \mathcal{M}_j$ and $v = v(x)b(t) \in \tilde{X}_j$ into (4.4.1) reduces this formula to

$$\inf_{aq \in \mathcal{M}} \sup_{bv \in \tilde{X}_j} \left\{ \frac{\int_\Omega q(x) \nabla \cdot v(x) \cdot \int_{t_{j-1}}^{t_j} a(t)b(t)\, dt\, dx}{\|q\|_{\mathrm{L}^2(\Omega)} \|\nabla v\|_{\mathrm{L}^2(\Omega)} \|a\|_{\mathrm{L}^2(I_j)} \|b\|_{\mathrm{L}^2(I_j)}} \right\} \geq \beta_j > 0, \quad (4.4.3)$$

since $(aq, bv) \to \int_{Q_j} aq \nabla \cdot (bv)\, dxdt$ is a continuous function of both arguments aq and bv (see Girault and Raviart [7], eq. (1.12) p.60). Note that, since $\tilde{X}_j \subset \mathcal{X}_j$, the inf-sup condition holds on the latter space if it holds on the former. Thus the inf-sup condition correspondingly splits into two conditions, of which one is the classical condition in the steady-state Stokes problem. This is recorded as

Theorem 4.4.2 *Let* $\mathcal{C}(\Omega) = \mathcal{B}$ *and* $\hat{\mathcal{C}}(\Omega) = \left[\hat{\mathcal{B}}\right]^2$. *Then equation (4.4.3)*

is implied by the following two conditions

$$\begin{aligned} &\inf_{q\in C(\Omega)} \sup_{v\in \hat{C}(\Omega)} \{ \frac{\int_\Omega q(x)\nabla\cdot v(x)\, dx}{\|q\|_{L^2(\Omega)}\|\nabla v\|_{L^2(\Omega)}} \} \geq \beta_{2,j} > 0 \\ &\inf_{a\in\mathcal{A}} \sup_{b\in\hat{\mathcal{A}}} \{ \frac{\int_{t_{j-1}}^{t_j} a(t)b(t)\, dt}{\|a\|_{L^2(I_j)}\|b\|_{L^2(I_j)}} \} \geq \beta_{1,j} > 0 \end{aligned} \tag{4.4.4}$$

Proof. This follows by rearrangement of (4.4.3). □

Consider some choices of products of finite element spaces which satisfy the conditions posed in this theorem. Note that the second condition in (4.4.4) is easy to satisfy is as shown below. The first condition of (4.4.4) is a classical one.

Theorem 4.4.3 *The following choices of $\mathcal{A}$ in combination with $\hat{\mathcal{A}}$ satisfy the second inequality in (4.4.4)*

(a) $\hat{\mathcal{A}} = \mathcal{A}$ *for any choice of* $\mathcal{A} = \hat{\mathcal{A}}$. *In particular for* $\mathcal{A} = \hat{\mathcal{A}} = \mathbb{P}^k(t_{j-1}, t_j]$, *the span of polynomials in t of degree less than or equal to* $k \in \mathbb{N}$.

(b) $\mathcal{A} = [\![1]\!]$ *and* $\hat{\mathcal{A}} = [\![1, t, t^2]\!]$.

(c) $\mathcal{A} = [\![1]\!]$ *and* $\hat{\mathcal{A}} = \{b \in \mathbb{P}^1(t_{j-1}, t_{j-\frac{1}{2}}) \wedge b \in \mathbb{P}^1(t_{j-\frac{1}{2}}, t_j)\}$.

(d) $\mathcal{A} = [\![1, t]\!]$ *and* $\hat{\mathcal{A}} = [\![1, t, t^2, t^3]\!]$, *or*

(e) more in general, any choice for which $\mathcal{A} \subset \hat{\mathcal{A}}$.

Proof. For all of the cases above, given $a \in \mathcal{A}$ pick $b = a \in \hat{\mathcal{A}}$. □

The cases (b), (c) and (d) are listed separately above since they will re-emerge when considering the discontinuous Galerkin method inf-sup condition. Now, as a concrete example of tensor product spaces $\mathcal{M}_j, \mathcal{X}_j$, consider the Hood-Taylor pair for the spaces $C(\Omega), \hat{C}(\Omega)$ and $\mathcal{A} = [\![1]\!]$, $\hat{\mathcal{A}} = [\![1, t]\!]$. This leads for a corresponding grid of prismatic elements $\triangle$ to

- $\mathcal{M}_j = \{q(x,t)\colon q \in C^0(Q_j) \cap L_0^2(Q_j) \wedge q_{|\triangle} \equiv b_0 + b_1 x_1 + b_2 x_2$ for some $b_0, b_1, b_2 \in \mathbb{R}\}$

- $\mathcal{X}_j = \{v(x,t): v \in \left[C^0(Q_j)\right]^2 \cap \left[H_0^1(Q_j)\right]^2 \wedge (v_{1,2})_{|\triangle} \equiv (a_0+a_1 t)\cdot (b_0+b_1x_1+b_2x_2+b_3x_1^2+b_4x_1x_2+b_5x_2^2)$ for some $a_0, a_1, b_0, \ldots, b_5 \in \mathbb{R}\}$.

Here the functions in $\mathcal{M}_j$ and $\mathcal{X}_j$ are defined elementwise by their restrictions to all prismatic elements $\triangle$.

4.5 The discontinuous inf-sup condition

With the error estimates for the pressure, involving the inf-sup condition, in mind define again the norm

$$|||v|||_{j-1,+}^2 = \nu \|\nabla v\|_{L^2(Q_j)}^2 + \tfrac{1}{2}\|v^+(x,t_{j-1})\|_{L^2(\Omega)}. \tag{4.5.1}$$

The inf-sup condition using this norm reads

$$\inf_{q\in\hat{M}_j} \sup_{v\in\hat{X}_j} \{\frac{\int_{Q_j} \mu\nabla\cdot v \, dx\,dt}{\|\mu\|_{L^2(Q_j)}|||v|||_{j-1,+}}\} \geq \beta_j > 0 \tag{4.5.2}$$

for the continuous case. Here $\hat{M}_j = \{\mu \in L^2(Q_j): \int_\Omega \mu(x,t)\,dx = 0$ for a.e. $t \in (0,t_J]\}$ and $\hat{X}_j$ is the closure of $\left[H_0^1(\Omega) \times H^1((t_{j-1},t_j])\right]^2$ in Q_j. For the discrete case $\hat{M}_j$ and $\hat{X}_j$ are replaced by $\mathcal{M}_j$ resp. $\mathcal{X}_j$.

In order to simplify the proof of the continuous analog of (4.5.2), only the case of a smooth boundary will be considered. Specifically, assume that the domain Ω is an open bounded set in $\mathbb{R}^N$ with $\Gamma = \partial\Omega$ an $n-1$ dimensional C^∞ variety with Ω locally on one side of Γ.

Let $\rho(x)$ denote, following Lions and Magenes [16], p. 171, a smooth positive function on Ω vanishing on Γ like $d(x,\Gamma)$, the distance of x to Γ. This means that for $x \in \Gamma$

$$\lim_{x\to x_0\in\Gamma} (\frac{\rho(x)}{d(x,\Gamma)}) \neq 0.$$

Now define the family of spaces Θ_j, following Lions and Magenes [16], pp. 180-185, as follows. Let $s \in \mathbb{R}$, $s > 0$ be such that $s - \frac{1}{2}$ is not an integer. Then

$$\Theta_j^s = \{\mu \in D'(Q_j): \rho^s\mu \in H_0^s(Q_j)\}$$

where D' denotes the dual of the space of infinitely times differentiable functions on Q_j with compact support on Q_j. For negative s the corresponding space Θ_j^s is defined as the dual of Θ_j^{-s}.

Theorem 4.5.1 *Suppose that the smoothness assumption on* Γ *holds. Then the inf-sup condition (4.5.2) holds.*

Proof. Let there be given a function $\mu \in L_0^2(Q_j)$. For almost every $t \in (0, t_J]$ construct a function $v \in \left[H_0^1(Q_j)\right]^2$ satisfying $\nabla \cdot v = \mu$ like in section 4.4. For this choice of v one clearly has

$$\frac{\int_{Q_j} \mu \nabla \cdot v \, dx dt}{|||v|||_{j-1,+} \|\mu\|_{L^2(Q_j)}} = \frac{\|\mu\|_{L^2(Q_j)}}{(\nu \|\underline{\nabla} v\|^2_{L^2(Q_j)} + \frac{1}{2}\|v^+(x, t_{j-1})\|^2_{L^2(\Omega)})^{\frac{1}{2}}}. \quad (4.5.3)$$

From standard results concerning this construction of v, see e.g. [7], it follows that for almost every time t there exists a positive scalar c, independent of the time, such that

$$\|\underline{\nabla} v\|_{L^2(\Omega)} \leq c \|\mu\|_{L^2(\Omega)}$$

and hence $\|\underline{\nabla} v\|_{L^2(Q_j)} \leq c \|\mu\|_{L^2(Q_j)}$. Thus it remains to be shown that

$$\|v^+(x, t_{j-1})\|_{L^2(\Omega)} \leq c \|\mu\|_{L^2(Q_j)}$$

for some $c > 0$. From standard results concerning trace operators in negative order Sobolev spaces, e.g. given in [16], pp. 175-177, it follows that for $r - \frac{1}{2} \notin \mathbb{N}$ for any $\mu \in D_A^{-r}(Q_j)$

$$\|\mu^+(x, t_{j-1})\|_{H^{-r-1/2}(\Omega)} \leq \|\mu\|_{D_A^{-r}(Q_j)}$$

where one may take the operator A to be, for example, the time-space Laplacian $A\mu = \mu_{tt} + \triangle \mu$ under Dirichlet boundary conditions. The D_A^{-r} norm is defined in [16], section 6, via interpolation spaces $\Theta^{-s}(Q_j)$ as

$$\|\mu\|^2_{D_A^{-r}(Q_j)} = \|\mu\|^2_{H^{-r}(Q_j)} + \|\mu_{tt} + \triangle \mu\|^2_{\Theta^{-2-r}(Q_j)}.$$

Regularity results associated with this construction of v give

$$\|v^+(x, t_{j-1})\|_{L^2(\Omega)} \leq c \|\mu^+(x, t_{j-1})\|_{H^{-1}(\Omega)}.$$

Further, with $r = \frac{1}{2} - \epsilon$ (so $r - \frac{1}{2}$ is not an integer) and $A\mu = \mu_{tt} + \triangle\mu$

$$\|\mu^+(x, t_{j-1})\|^2_{H^{-1}(\Omega)} \leq \|\mu\|^2_{D_A^{-\frac{1}{2}+\epsilon}(Q_j)} \leq c\|A^{-1}\mu\|^2_{D_A^{\frac{3}{2}+\epsilon}(Q_j)}, \quad (4.5.4)$$

where again results from [16], section 6.6, pp. 177-180, have been used to obtain the last inequality. Thus, from the definition of the D_A^{-s} norm

$$\|q\|^2_{D_A^s(Q_j)} = \|q\|^2_{H^s(Q_j)} + \|Aq\|^2_{\Theta^{-2+s}(Q_j)}, \quad (4.5.5)$$

(see [16], section 6.3, pp. 170-173), inserting (4.5.5) with the identification $q = A^{-1}\mu$ into (4.5.4), gives

$$\|\mu^+(x, t_{j-1})\|^2_{H^{-1}(\Omega)} \leq c\left(\|A^{-1}\mu\|^2_{H^{\frac{3}{2}+\epsilon}(Q_j)} + \|\mu\|^2_{\Theta^{-\frac{1}{2}+\epsilon}(Q_j)}\right). \quad (4.5.6)$$

Now for $0 < \epsilon < \frac{1}{2}$, theory of elliptic operators ensures that

$$\|A^{-1}\mu\|_{H^{\frac{3}{2}+\epsilon}(Q_j)} \leq c\|\mu\|_{L^2(Q_j)}.$$

Further, since for $s > 0$ one has the relation $H^s(Q_j) \subset \Theta^s(Q_j) \subset L^2(Q_j)$, it follows that for $s > 0$ this implies $H^{-s}(Q_j) \supset \Theta^{-s}(Q_j) \supset L^2(Q_j)$ with continuous imbeddings. Thus for $-s = -\frac{1}{2} + \epsilon$

$$\|\mu\|_{\Theta^{-\frac{1}{2}+\epsilon}(Q_j)} \leq c\|\mu\|_{L^2(Q_j)}, \quad 0 < \epsilon < \tfrac{1}{2}.$$

Substituting these two inequalities in the previous one (4.5.6) gives

$$\|v^+(x, t_{j-1})\|_{L^2(\Omega)} \leq c\|q\|_{L^2(Q_j)}$$

which suffices to complete the proof. □

Remark 4.5.1 The requirement that $\int_\Omega \mu(x,t)\,dx = 0$ for almost every $t_{j-1} \leq t \leq t_j$, rather than $\int_{Q_j} \mu(x,t)\,dxdt = 0$ is necessary for even the beginning of the verification of the inf-sup condition, if classical result are to be used to this end.

4.6 Two-level hierarchical bases

As is known one can decompose the solution of the Stokes equation into a part related to the smallest eigenvalues of the Laplacian operator $\triangle$, and into a second (high frequency) part corresponding to the larger eigenvalues. One can show that the high frequency part is essentially time-independent relative to the low frequency part (see [17]). Since a semi-discrete Galerkin time-stepping approach, using a fixed time-step size to approximate a discrete solution, neglects the influence of higher frequency components such a method can lead to significant errors, in particular when integration over a long time-period is taken into account.

Using time-slabbing techniques one can treat each high frequency component on its own time-scale since grid refinement here means refinement simultaneously in space and time. As usual the *higher frequency* basis functions correspond to those basis functions created hierarchically after refinement of a grid defining the *low frequency* basis functions. Creating hierarchically defined growing finite element spaces implies that for each time-slab one is able to solve for various scales of physical details.

Therefore, consider the use of a two-level hierarchical finite element basis for the approximation of the flow field vector $\hat{\boldsymbol{u}}_h$ in (4.2.4). It will be shown that, for a two-level hierarchical basis streamline upwind formulation of the Stokes problem (4.2.4), the finest level contribution to $\hat{\boldsymbol{u}}_h$ is small relative to coarse level contribution. Thereafter it is shown how to compute a cheap initial approximation for the finest level contribution.

As an example consider the solution of the Stokes problem (see also [21] on the unit-cube $Q = \Omega \times (0, t_J] \equiv (0,1)^2 \times (0,1)$, divided into h^{-3} small subcubes. A coarse piecewise linear basis function space $\mathcal{V}$ on Q is defined via the small cubes as is shown in fig. 1.3. Now suppose $\hat{\boldsymbol{u}}_h \in \mathcal{V}$ is a solution to the Stokes problem (4.2.4), then the use of the streamline upwind formulation (1.6.5) with $\delta = \nu$ and $\boldsymbol{b} \equiv 0$ leads to

$$\int_0^{t_J} \int_\Omega \left[(\hat{\boldsymbol{u}}_h)_t \hat{\boldsymbol{u}}_h + \nu \underline{\nabla}_{x,t} \hat{\boldsymbol{u}}_h : \underline{\nabla}_{x,t} \hat{\boldsymbol{u}}_h - \boldsymbol{f}(\hat{\boldsymbol{u}}_h + \delta(\hat{\boldsymbol{u}}_h)_t) \right] d\boldsymbol{x}\, dt = 0 \qquad (4.6.1)$$

whence, analogous to the derivation of the equation in theorem 4.3.1, one can show that there exists a positive scalar c such that

$$\nu \int_0^{t_J} \int_\Omega \left| \underline{\nabla}_{x,t} \hat{u}_h \right|^2 \, dx dt + \int_\Omega \hat{u}_h^2(x, t_J) \, dx \leq \frac{c}{\nu} |||f|||_0^2 + \int_\Omega u_0^2(x) \, dx \, .$$

A discretization error bound for (4.6.1) can be obtained analogously to the derivation (4.2.7).

A hierarchical extension to $\mathcal{V}$ now is constructed using the non-standard uniform refinement of tetrahedrons as shown in fig. 1.3. The span of the piecewise linear basis functions created will be denoted by $\mathcal{V}^+$. Now let $\hat{u}_h = u \oplus v \in \mathcal{V} \oplus \mathcal{V}^+$ be the solution of (4.6.1) on the new two-level hierarchical basis $\mathcal{V} \oplus \mathcal{V}^+$. For the sake of simplicity redefine $|||\cdot|||_1$ for the remainder of this section by

$$|||v|||_1 = |||\underline{\nabla}_{x,t} v|||_0 \quad \forall_{v \in V} \, ,$$

differing from the previous definition in that the gradient component in time is included. Then, using a strengthened Cauchy-Bunyakovski-Schwarz inequality given by

$$\int_0^{t_J} \int_\Omega \underline{\nabla}_{x,t} u : \underline{\nabla}_{x,t} v \, dx dt \leq \gamma |||u|||_1 \cdot |||v|||_1 \quad \forall_{u \in \mathcal{V}} \forall_{v \in \mathcal{V}^+} \, ,$$

one obtains that there exists a scalar $0 \leq \gamma < 1$, independent of ν, such that

$$\nu(1 - \gamma^2) \left\{ |||u|||_1^2 + |||v|||_1^2 \right\} \leq \frac{1}{\nu} |||f|||_0^2 + \int_\Omega u_0^2(x) \, dx \, , \qquad (4.6.2)$$

where the last term vanishes for homogeneous Dirichlet boundary conditions.

Now note that for elements of the sets $\mathcal{V}$ and $\mathcal{V}^+$ the following relations hold

$$\begin{aligned} c_1 |||u|||_0^2 &\leq |||u|||_1^2 \quad \forall_{u \in \mathcal{V}} \\ c_2 h^{-2} |||v|||_0^2 &\leq |||v|||_1^2 \quad \forall_{v \in \mathcal{V}^+} \end{aligned} \qquad (4.6.3)$$

For $u \in \mathcal{V}$ equation (4.6.3) holds since the restriction $u(t) \in H_0^1(\Omega)$ for all $t \in (0, t_J]$. Using a classical Friedrichs inequality for fixed time t

$$c_1 \cdot \int_\Omega u^2(x,t)\, dx \leq \int_\Omega |\underline{\nabla}_x u(x,t)|^2\, dx, \tag{4.6.4}$$

integrating over the interval $(0, t_J]$, and using $|\underline{\nabla}_x u|^2 \leq |\underline{\nabla}_{x,t} u|^2$ gives the desired result since the domain Ω does not vary with time. For $v \in \mathcal{V}^+$ equation (4.6.3) holds because there exist positive scalars $\hat{c}_1$ and $\hat{c}_2$ such that for all basis functions $w \in \mathcal{V}^+$

$$\int_0^{t_J} \int_\Omega |\underline{\nabla}_{x,t} w(x,t)|^2\, dx dt = \hat{c}_1 \cdot h \text{ and } \int_0^{t_J} \int_\Omega w^2(x,t)\, dx dt = \hat{c}_2 \cdot h^3 .$$

These basis functions have empty mutual *support*, whence for $w = \sum_i \alpha_i \varphi_i$ (φ_i are the basis functions spanning $\mathcal{V}^+$) and $c_2 = \hat{c}_1/\hat{c}_2$

$$\begin{aligned} |||w|||_0^2 &= \sum_i \alpha_i^2 |||\varphi_i|||_0^2 = \sum_i \alpha_i^2 c_2^{-1} h^2 |||\varphi_i|||_1^2 \\ &= c_2^{-1} h^2 \sum_i \alpha_i^2 |||\varphi_i|||_1^2 = c_2^{-1} h^2 |||w|||_1^2 \end{aligned}$$

leading to the inequalities in (4.6.3). Note that this proof differs from proofs in [22] and [20] since it exploits the fact that the basis functions have empty mutual support. With the use of (4.6.3) one now obtains

$$\nu(1-\gamma^2)\left\{|||u|||_0^2 + h^{-2}|||v|||_0^2\right\} \leq \frac{c}{\nu}|||f|||_0^2 + \int_\Omega u_0^2(x)\, dx\,, \tag{4.6.5}$$

whence for a fixed-size time-slab clearly the contribution of $v \in \mathcal{V}$ to $\hat{u}_h$ is small relative to the contribution of $u \in \mathcal{V}^+$.

Taking (4.6.5) into account, an initial approximation can be computed as follows. Let $u \in \mathcal{V}$ be the finite element solution of the Stokes problem (4.6.1) and let $\hat{u}_h \in \mathcal{V} \oplus \mathcal{V}^+$ be the finite element solution of (4.6.1), using the hierarchically extended basis. Then $\hat{u}_h$ can be approximated by $\hat{u}_h = u + v$ for some function $v \in \mathcal{V}^+$. Since the

contribution $v \in \mathcal{V}^+$ is time-independent relative to $u \in \mathcal{V}$ (see [17]) one may substitute $v_t = 0$ in (4.6.1), and because of (4.6.5) one may assume $\underline{\nabla}_x v : \underline{\nabla}_x v = 0$, in the same equation. This leads to the linear relation

$$\int_Q \left[u_t v + \nu \underline{\nabla}_x u : \underline{\nabla}_x v - f v\right] \, dx\,dt$$
$$= \tfrac{1}{2} \int_Q \left[f(u + \delta u_t) - u_t u - \nu \left|\underline{\nabla}_{x,t} u\right|^2\right] \, dx\,dt$$

which provides an initial approximation for $v \in \mathcal{V}^+$. As this system of equations for v has the dimension of $\mathcal{V}^+$ and the basis functions in $\mathcal{V}^+$ have empty mutual support, the approximation $v \in \mathcal{V}^+$ is easy to calculate.

Clearly, hierarchical finite element bases, which are exploited in many different ways in classical Galerkin variational formulations, can also be applied effectively for the global time-space approach. This enables the use of standard finite element packages for the solution of time-space variational formulations.

Acknowledgements

The authors wish to thank the Pittsburgh Supercomputing Center for its assistance and support. They also thank J. Boland for his helpful comments on a preliminary draft of this chapter and M. Marsden for a stimulating discussion of the inf-sup condition for the discontinuous Galerkin method.

4.7 References

[1] Axelsson O. and Maubach J., *Stability and high order approximation of monotone evolution equations valid for unbounded time by continuous time slabbing methods,* internal report of the Supercomputer Computations Research Institute, Florida State University, Tallahassee, U.S.A., 1990

[2] Axelsson O. and Maubach J., *A time-space finite element discretization technique for the calculation of the electromagnetic*

field in ferromagnetic materials, Journal for Numerical Methods in Engineering, 29(1989), 2085-2111

[3] Aziz A.K. and Monk P., *Continuous finite elements in space and time for the heat equation,* Mathematics of Computation, 52(1989), 255-274

[4] Boland J. and Layton W., *Error analysis for finite element methods for steady natural convection problems,* ICMA report, University of Pittsburgh, 1989, to appear in Numerical Functional Analysis and Applications

[5] Boland J. and Nicolaides N., *Stable and Semi-stable low order finite elements for viscous flows,* SIAM Journal on Numerical Analysis, 22(1985), 474-492

[6] French D., *Continuous finite element methods which preserve energy properties for nonlinear problems,* In preparation

[7] Girault V. and Raviart P.A., *Finite Element Approximation of the Navier-Stokes Equations,* Lecture Notes in Mathematics, Vol. 749, Springer Verlag, Berlin, Heidelberg, New York, 1979

[8] Girault V. and Raviart P.A., *Finite Element Methods for Navier-Stokes Equations,* Springer Verlag, Berlin, 1986

[9] Gunzburger M., *Finite Element Methods for Viscous Incompressible Flows: A Guide to Theory, Practice and Algorithms,* Academic Press, Boston, 1989

[10] Hulme B.L., *Discrete Galerkin methods and related one-step methods for ordinary differential equations,* Mathematics of Computation, 26(1972), 881-891

[11] Hulme B.L., *One-step piecewise polynomial Galerkin methods for initial value problems,* Mathematics of Computation, 26(1972), 415-426

[12] Jamet P., *Galerkin-type approximations which are discontinuous in time for parabolic equations in a variable domain,* SIAM Journal on Numerical Analysis, 15(1978), 912-928

[13] Jamet P., *Stability and convergence of a generalized Crank-Nicolson scheme on a variable mesh for the heat equation,* SIAM Journal on Numerical Analysis, 17(1980), 530-539

[14] Johnson C., *Numerical Solution of Partial Differential Equations by the Finite Element Method,* 3rd printing, Cambridge University Press, Cambridge, 1990

[15] Lions J.L., *Quelque Methodes de Resolution des Problemes aux Limites non Lineaires,* Dunod, Paris 1969

[16] Lions J.L. and Magenes E., *Non-Homogeneous Boundary Value Problems and Applications I,* Grundlehren der Mathematischen Wissenschaften, Vol. 181, Springer Verlag, Berlin, Heidelberg, New York, 1972

[17] Marion M. and Temam R., *Nonlinear Galerkin methods,* SIAM Journal on Numerical Analysis 26(1989), 1139-1157

[18] Maubach J.M., *Preconditioned iterative methods for problems discretized in time-space,* **in** Lecture Notes of the Summerschool on Preconditioned Conjugate Gradient Methods and Applications, 274-291 Nijmegen, The Netherlands, 1989

[19] Nečas J., *Les Méthodes Directes en Théorie des Equations Elliptiques,* Masson, Paris, 1967

[20] Ong M., *Hierarchical basis preconditioners for second order elliptic problems in three dimensions,* Ph.D. thesis, CAM report 89-31, Department of Mathematics, University of California at Los Angeles, Los Angeles, CA. 90024-1555

[21] Temam R., *Navier-Stokes Equations,* Third edition, North-Holland , Amsterdam, New York, 1984

[22] Yserentant H., *On the multilevel splitting of finite element spaces,* Numerische Mathematik, 49(1986), 379-412

5 On finite element matrices and locally refined grids

An extended version of: Maubach J.M., On the sparsity structure of hierarchical finite element matrices, published in Lecture Notes in Mathematics 1457, 79-104, (Axelsson O. and Kolotilina L. Yu. eds.), Springer Verlag, 1990, [Proceedings of the International Conference on Preconditioned Conjugate Gradient Methods and Applications, Nijmegen, The Netherlands, 1989].
A section on bisection refinement in three dimensions and one on block decay rates is included, the section on numerical examples is extended and the section on the standard nodal and hierarchical basis functions used to be part of the section on the newest vertex local grid refinement.
Printed with permission of Springer-Verlag GmbH & Co. KG; part of this chapter is copyrighted by Springer-Verlag.

Abstract

The performance of preconditioned conjugate gradient methods for the solution of a linear system of equations $Hx = b$ depends strongly on the quality of the preconditioner. In general, the matrix H is a sparse matrix whose sparsity pattern only depends on discretization choices for a given node ordering. For the construction of a preconditioner only the matrix entries are needed, but investigations so far have shown clearly that taking into account the sparsity pattern structure, leads to more effective preconditioning techniques.

As the sparsity pattern is of importance for the construction of good preconditioners, it is analyzed here for the hierarchical matrix H resulting from a given discretization. The hierarchy is induced by the grid refinement method applied. It is shown that the sparsity pattern is

irregular but well structured in general, and a simple refinement method is presented which enables a compact storage and quick retrieval of the matrix entries in the computer memory. An upper bound for the C.-B.-S. scalar for this method is determined to demonstrate that it is well suited for multi-level preconditioning and it is shown to have satisfying angle bounds. Further, it turns out that the hierarchical matrix may be partially constructed in parallel, is block structured and shows fast block decay rates.

Key words: Sparsity structure, Adaptive grid refinement, Hierarchical finite elements, Error indication, Parallel computing
AMS(MOS) subject classifications: 65N30, 65N50, 65W05, 65F10, 65F10

5.1 Introduction

At present the use of hierarchical finite element basis functions, in combination with adaptive grid refinement for the solution of elliptic boundary value problems, has been investigated thoroughly (see e.g. [11], [12], [14], [20], [28], [29], [31], [32] and [38]) to show that this approach is eminently applicable to problems with complicated boundary geometry. Most approaches investigated so far implicitly use the hierarchical matrix H, resulting from a grid obtained by an adaptive grid refinement method, via the relationship $H = I^T A I$ as in [38]. Here A is the standard nodal matrix on the grid mentioned and I is a grid geometry dependent lower triangular identity transformation from the hierarchical to the standard nodal basis. As multiplications with I^{-t} and I are easy to perform, solving $H\boldsymbol{x} = \boldsymbol{b}$ for a given vector $\boldsymbol{b}$ with the use of the equivalent system $A(I\boldsymbol{x}) = I^{-t}\boldsymbol{b}$ as in [21] has several advantages. It is possible to assemble and store A and multiply by it elementwise, see e.g. [17]. The number of matrix entries to be stored will be proportional to the number of unknowns. On the other hand, I is often defined recursively, prohibiting a parallel multiplication (one can use domain decomposition techniques to overcome this) and the use of elementwise stored data induces much indirect memory addressing each time a multiplication is performed, slowing down the overall computational performance.

All approaches mentioned avoid an analysis of the *sparsity pattern* J, the set of couplings (i,j) corresponding to possible non-zero entries h_{ij} of the hierarchical matrix H, which is more complicated in the hierarchical case than that of ordinary finite elements. Because one does not know the sparsity patterns structure a straightforward row-wise ordered storage technique (see e.g. [3]) is used in general; for each degree of freedom i, the numbers j for which $(i,j) \in J$ are stored in an integer row. This technique is often quite expensive due to a generally large number of shifts needed to store the numbers j in an increasing order. To check whether $(i,j) \in J$ for given i and j, e.g. occurring during a pointwise incomplete factorization of H, is costly because this involves searching through an integer row.

This chapter will show that for the discretization choices of any given grid refinement technique, a coarse initial grid and type of finite element basis functions, it is possible to determine the sparsity patterns structure of the resulting hierarchical matrix by studying the changes in J caused by refinement of a single triangle. This shows for frequently used refinement techniques that the sparsity pattern has a (sometimes even binary) tree structure which can be stored in a row-wise ordered manner without any shifting and such that checking whether $(i,j) \in J$ for given j involves at most one 'if...then' instruction, independent of the dimension of the matrix and the number of refinements applied to the initial coarse grid.

Unfortunately the number of possible non-zero entries of H is bounded above by $O(kN)$, k the number of refinements and N the number of unknowns, whence a matrix vector multiplication will not be of optimal computational complexity $O(N)$ if the number of refinement levels is unrestricted. In general the entries $[H]_{ij}$ are of order $O(\sqrt{2}^{-|(p-q)|})$ if the related basis functions i and j are of level p respectively q as is shown in [26]. Also, the storage of the hierarchical matrix in the computer memory allows for the direct application of many well-known preconditioning techniques.

Since the sparsity patterns structure depends on the type of local refinement technique used, some types of refinement techniques will be studied in more detail. A refinement technique introduced to this end is the *newest vertex bisection refinement* proposed by Sewell [34], [35] and

adapted by Mitchell [28], similar to the *longest edge bisection* method of Rivara [32] and [33]. The difference between the methods is that for a given triangle the former method bisects the edge opposite a specific vertex whereas the latter method bisects its longest edge. Unlike the *regular grid refinement* used by [5], Bank [12], [18], [20] and Yserentant [38] these bisection methods do not create children congruent to their parents.

The main advantage of the newest vertex technique proposed is its simplicity. Mitchells technique is a one fase recursive technique, which keeps all triangles *compatible* (at most one neighbour along each of the three edges) at all times contrary to Rivara, who has to enforce compatibility after each refinement by refining an additional number of triangles, and Bank and Yserentant [11], who even need a third fase in which the bisections used to enforce compatibility are removed. Further advantages are that there is no need to compute side lengths as in Rivaras refinement algorithm and that the number of different angles created during repeated refinement is at most eight times the number of triangles in the initial coarse grid whence a properly chosen initial grid avoids bad angles (see e.g. [9]) automatically. The newest vertex bisection method is generalizable to domains in more than two space dimensions and for higher order finite element basis functions.

As the sparsity pattern is analyzed bearing the construction of an effective multi-level preconditioner in mind, angle bounds for the refinement methods above are investigated and the Cauchy-Buniakowskii-Schwarz scalar γ^2 is derived. The results obtained demonstrate that grid refinement methods leading to a well structured sparsity pattern in addition can be well suited for multi-level preconditioning.

The remainder of the chapter is organized as follows. The newest vertex grid refinement method, which will be used to demonstrate the results of the sparsity pattern analysis in the view of its simplicity, is introduced in section 5.2. The construction of a standard nodal and hierarchical finite element basis is given brief attention in section 5.4 after which in section 5.5 a sparsity pattern analysis is presented for the hierarchical basis, for various grid refinement techniques. The block decay rate of the hierarchical matrices is studied in section 5.7. The number of the hierarchical matrix entries as well as the storage of the complete

hierarchical matrix in the computer memory are investigated briefly in section 5.6. Section 5.8 considers angle bounds for the grid refinement methods studied and provides estimates for γ^2 in order to show that the newest vertex refinement method, having one of the simplest structured sparsity patterns possible, may also be used for the construction of multilevel preconditioners. Finally in section 5.9 sparsity patterns resulting from some discretizations of example partial differential equations are presented and in section 5.10 some conclusions are drawn.

5.2 The newest vertex local grid refinement

Let the computational domain $Q \subset \mathbb{R}^n$, be an open and bounded polygonal domain covered by an initial coarse grid of simplices. Here *simplex* stands for *interval* in 1 dimension, for *triangle* in 2 dimensions and for *tetrahedron* in 3 dimensions. For instance, an initial coarse grid on a line could consist of a single interval, on the unit-square it could consist of two triangles, and on the unit-cube it could consist of at least 5 tetrahedra (see e.g. [30]). The domain will be only in space in the case of a static differential equation, and will be in time-space in the case of time-slabbing, where $Q = Q_j$ (see the notation introduced in chapter 1). In general it is very difficult to fit a grid to an arbitrary domain (see [24]), therefore, to start with, only the case of a refinement of a given initial coarse triangulation will be paid attention to. This section introduces the newest vertex grid refinement method and the properties of it that are essentially determined the resulting matrices sparsity pattern. These properties are very basic and are satisfied by most existing refinement methods, like e.g. the regular refinement method.

In order to obtain a grid which is suited for the finite element solution of a partial differential equation it is of importance that

- the refinement algorithm will always lead to a compatible grid,
- that the recursion involved will be of finite length and allows for local refinement, i.e., not forcing uniform refinement,
- the angles of the tetrahedra created are bounded above away from π (see [9]) which is automatically the case if only a finite number of similarity classes is generated.

For the two-dimensional variant this is shown by Mitchell using induction techniques in combination with graph theory.

Let a triangle be a topologically open subset of $\mathbb{R}^2$. Now let an initial coarse grid $\mathcal{Q}^{(0)} = \{\triangle\}$ of triangles be given such that $\{x \in \triangle : \triangle \in \mathcal{Q}^{(0)}\} = Q$. This implies that the boundary of Q should be piecewise linear, whence *isoparametric elements* as in [3] are not considered in this chapter. A basic building block of each grid refinement algorithm is the method to divide a triangle. In the newest vertex bisection case a triangle $\triangle \subset \mathbb{R}^2$ of a certain *level* $l(\triangle)$ has three vertices $x_1, x_2, x_3 \in \mathbb{R}^2$, of which the first is called the *newest vertex*, and denoted by denoted by $x_\triangle$. By definition the edge opposite this vertex is called the *base* and the triangle sharing this base is said to be the *neighbour*, if this exists (see fig. 5.4). The bisection of a triangle $\triangle$ always takes place by adding an edge from its newest vertex to the midpoint of its base. The vertex x created on the base will be the newest vertex of both *children* $\triangle_i$ $(i = 1, 2)$ which satisfy

$$\triangle = \cup_i \triangle_i, \quad \triangle_i \cap \triangle_j = \emptyset \text{ if } i \neq j\,. \tag{5.2.1}$$

By definition the levels of the children and their vertices are given by

$$l(\triangle_i) = l(x_{\triangle_i}) = l(\triangle) + 1\,. \tag{5.2.2}$$

All vertices, edges and triangles of the initial coarse grid $\mathcal{Q}^{(0)}$ have level 0. Further, the children $\triangle_i$ have the unique *parent* $P(\triangle_i)$ and the subsequent *ancestors*

$$P^k(\triangle) = \underbrace{(P \circ \cdots \circ P)}_{k}(\triangle) \quad \forall_{k=1,\ldots,l(\triangle)}$$

where $P^0(\triangle) = \triangle$ in order to simplify notations.

As mentioned before, one of the difficulties in grid refinement is that of maintaining the compatibility of the triangulation. Therefore consider the refinement of a triangle in a *perfect matching* given grid, i.e., a grid in which all triangles are *compatibly divisible*. This means that for every triangle either its base is lying at the boundary or its neighbours base

coincides with its own base. Now the refinement of a triangle is defined by the following recursive algorithm:

Refine the triangle:
If *the neighbour is not compatibly divisible*
Then *Refine the neighbour*
Fi
bisect the triangle and its neighbour.

Figure 5.9 shows, starting from a perfect matching coarse initial grid, some refinement steps including the refinement of a not compatibly divisible triangle. Further, figs. 5.10 and 5.11 show an already refined grid. It is of course of importance that the refinement algorithm above will always yield a compatible grid and that the recursion involved will be of finite length and not too large. That this is easy, is shown by the following lemma.

Lemma 5.2.1 *For the newest vertex method the following statement holds*

(i) Given any compatible initial triangulation, there exists a choice of newest vertices such that every triangle is compatibly divisible.
(ii) The length of the recursion involved with the refinement of an arbitrary triangle is bounded by its level plus 1.
(iii) For each pair of compatibly divisible triangles $\triangle_1$ *and* $\triangle_2$, $l(\triangle_1) = l(\triangle_2)$.

Proof. See Mitchell [29] for a proof in full detail. □

Note that the local refinement of a pair of triangles is completely determined by the local numbers of their vertices. This implies that the labeling of the vertices for the children is the most important part of the local newest vertex refinement. Now the refined or initial coarse grid will be used to construct the finite element basis functions needed for the solution of the partial differential equation under consideration.

5.3 Bisection refinement in three dimensions

In Maubach [27] it is shown that the newest vertex approach can not

straightforwardly be extended to the case of 3 or more dimensions. For higher dimensions, [27] introduces a new simplex bisection refinement based on coverings of the n-cube with simplices. This method leads to compatible grids and creates at most n different congruency classes for each simplex, independent on the level of subsequent refinement. The algorithm has the advantage that it can be used in any dimension, but it is only applicable to simplicial grids which are part of a standard simplicial covering of the n-cube (see for instance [1]). In three dimensions several refinement methods exist. One of the first discovered was a bisection method by Bänsch [15] which is likely to be equivalent to the bisection method described below.

The bisection of an n-simplex presented below, only involves the ordering of the vertices of this simplex (as is the case in Mitchell's newest vertex method), and the level of the simplex under consideration. Initially, all coarse grid n-simplices T are said to be of level $l(T) = 0$. If a simplex T is bisected, the two created simplices are called its children, and the ordering of their vertices is defined by the following bisection step.

Bisect (simplex):
BEGIN
Let k := n - l(simplex) mod n;
Get simplex vertices: $x_0, x_1, \ldots, x_{n-1}, x_n$;
Create the new vertex: $z := \frac{1}{2}\{x_0 + x_k\}$;
Create child$_1$: $x_1, x_2, \ldots, x_k,\ z, x_{k+1}, \ldots, x_n$;
Create child$_2$: $x_0, x_1, \ldots, x_{k-1}, z, x_{k+1}, \ldots, x_n$;
Let l(child$_1$) := l(simplex) + 1;
Let l(child$_2$) := l(simplex) + 1;
END.

Then, reformulating Mitchells proposal in [28] leads to the following refinement algorithm of a tetrahedron

Refine the tetrahedron:
While *a neighbour is not compatibly divisible*
Do *Refine the neighbour*
Od

bisect the tetrahedron and its neighbours.

Note that this is a recursive algorithm: in order to refine a single tetrahedron it is possible that a whole chain of incompatible neighbours – of neighbours – must be refined. The algorithm is easy to implement but for the part of finding neighbours (this can be done using permutations as in [27], but there may be more simple and effient ways).

The bisection step in combination with the refinement algorithm has been thoroughly tested in 2 and 3 dimensions. First consider some simple examples, where the simplices are refined if they intersect a certain line or plane, or contain a point.

- Uniform refinement of the cube is shown in fig. 5.46,
- Local refinement in a plane $\{(x, y, z): z = 0\}$ and along a line $\{(x, y, z): x = 0 \wedge y = z\}$ is shown in figs. 5.47 and 5.48,
- Local refinement around $(1, 1, 1)$ and $(\frac{1}{\sqrt{7}}, \frac{1}{2}\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{5}})$ can be found in the figures 5.49 resp. 5.50 and 5.51.

In a second example all tetrahedra intersecting the hemi-sphere $x \geq \frac{1}{2}$ and $(x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 + (z - \frac{1}{2})^2 = \frac{1}{16}$ are refined. Pictures to be found in figures 5.52 – 5.57 show that a refinement up to 18 levels deep is very local and that the positions of the vertices reflects the surface of the hemi-sphere. Further, random refinement creating grids containing up to $O(10^5)$ tetrahedrons has been observed to function well.

5.4 The standard nodal and hierarchical basis

As the sparsity pattern of the finite element matrix to be considered, not only depends on the underlying grid, but also depends on the type of finite element basis functions used, these functions will be considered in more detail. First the types of bases to be used in the numerical examples are described, after this basic properties of the basis functions, needed for the sparsity pattern analysis, are considered.

Let a compatibly divisible grid $\mathcal{Q}$ be given. At each grid point a finite element basis function φ is defined as is usual in finite element methods,

piecewise linear or quadratic, and only locally non-zero on the surrounding simplices (see section 1.5 or [3], [16], [19], [22], [36] and [39]). If all nodes on $\mathcal{Q}$, which can be a locally refined grid, are defined this way, the finite element basis will be called *standard nodal finite element basis.* By way of abbreviation a finite element basis function will sometimes be called *node* in the sequel.

Another method to define the finite element basis is to exploit the local grid refinement. First, on an initial coarse compatibly divisible grid $\mathcal{Q}$ a standard nodal finite element basis is defined as above. Then, during the local grid refinement, every time a new point is created, its related finite element basis function is defined as above, using the grid as this is directly after the local refinement. This leads to a situation in which nodes can exist with a strongly varying magnitude in support (see below), small support typically corresponding to nodes of a higher level. Proceeding this way for all points created during the refinement phase, the resulting basis is called a *hierarchical finite element basis.*

Now consider properties of the finite element basis functions which are of importance for the sparsity pattern analysis of the resulting matrix. To this end let the *support* and the *base* of a node φ, defined at a unique vertex $\boldsymbol{x}_\varphi$, be defined by respectively

$$D_\varphi := \{\boldsymbol{x} \in Q : \varphi(\boldsymbol{x}) \neq 0\}^\star$$
$$B_\varphi := \{\triangle \in \mathcal{Q} : D_\varphi \cap \triangle \neq \emptyset\}$$

where for a set X, $X^\star$ denotes the open part of its topological closure. The supports of nodes created by regular resp. bisection refinement are shown in figures 5.2 and 5.4. The level of φ is by definition given by $l(\varphi) = l(\boldsymbol{x}_\varphi)$. If $\boldsymbol{x}_\varphi = \boldsymbol{x}_\triangle$, the newest vertex of some triangle $\triangle \in \mathcal{Q}$, then in the sequel, in order to simplify the notation, $\varphi = \varphi_\triangle$. Further, as is standard in finite elements, the grid is supposed to be such that there exists a scalar $c \in \mathbb{N}$, independent of φ, such that $|B_\varphi| \leq c$, where for a set X, $|X|$ denotes its number of elements. Also, the refinement is supposed to lead to a *compatible* grid, which is formulated by the requirement

$$\triangle \in B_\varphi \Rightarrow \boldsymbol{x}_\varphi \in v(\triangle) \wedge l(\triangle) = l(\boldsymbol{x}_\varphi)\,, \tag{5.4.1}$$

where $v(\triangle)$ denotes the set of vertices of $\triangle$, implying that if $\triangle \in B_\varphi$ then $\boldsymbol{x}_\varphi$ is one its vertices.

For the following definitions, assume that one is dealing with a time-dependent partial differential equation, discretized with the use of a continuous global time-space finite element method. In the time independent case, the definitions are valid if all subscripts j are omitted. Now define

- the triangle set $\mathcal{T}^{(k)} = \mathcal{T}_j^{(k)}$ to be the set of all triangles of a certain level $k \geq 0$ on time-slab j. Note that a triangle in $\mathcal{T}^{(k)}$ can only be refined once and that its children belong to $\mathcal{T}^{(k+1)}$. As long as the children exist, the triangle will remain in $\mathcal{T}^{(k)}$.
- the grid $\mathcal{Q} = \mathcal{Q}_j = \mathcal{Q}_j^{(k)} = \cup_{s=1}^{k} \mathcal{T}_j^{(s)}$, i.e., the set of all triangles up to and including level k on time-slab j.
- the set containing all vertices $\mathcal{V}(\mathcal{T})$ of an arbitrary set of triangles $\mathcal{T}$. Note that $\mathcal{V}(\mathcal{Q}) - \mathcal{V}(\mathcal{Q}^{(0)})$ is equal to the set of all created vertices and that by definition $\mathcal{V}^{(k)} = \mathcal{V}(\mathcal{Q}_j^{(k)}) - \mathcal{V}(\mathcal{Q}_j^{(k-1)})$ is the set of all vertices of level $k \geq 1$. In order to simplify the notations in the sequel let $\mathcal{V} = \mathcal{V}(\mathcal{Q})$.
- the set of finite element basis functions $\mathcal{F}(\mathcal{T})$ defined at the different vertices of an arbitrary set of triangles $\mathcal{T}$. Also here $\mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ is equal to the set of all created basis functions.
- the span of the finite element basis functions in the set $\mathcal{F}(\mathcal{Q}_j^{(k)})$, denoted by $\mathcal{H}(\mathcal{Q}) = \mathcal{H}(\mathcal{Q}_j^{(k)})$. Note that $\mathcal{H}(\mathcal{Q}) \subset \mathrm{H}^1(Q)$ and that this span is identical for the hierarchical and the standard nodal basis.
- the total number of different vertices of level k, $n^{(k)} = \left|\mathcal{V}^{(k)}\right|$. Analogously define $N = N_j = N_j^{(k)} = \left|\mathcal{V}(\mathcal{Q}_j^{(k)})\right|$ to be the total number of vertices in $\mathcal{Q}$, also said to be the number of *degrees of freedom.*

Note that the subset of $\mathcal{Q}$ of triangles without children in the case of newest vertex grid refinement has exactly $\frac{1}{2}(|\mathcal{Q}| + |\mathcal{Q}_0|)$ elements, which can be proved easily by induction. This subset is important as it serves the definition of a standard nodal basis.

For the assembly of the finite element matrix, after the grid refinement has taken place, each node is assigned a unique number $i \in \{1, \ldots N\}$,

in such a way that for all nodes

$$l(\varphi_i) \leq l(\varphi_j) \Leftrightarrow i \leq j\,. \tag{5.4.2}$$

This leads to an hierarchical matrix with a natural block structure, where the blocks are determined by the levels of the refinement. It is advisable to number the vertices and corresponding nodes of the initial coarse grid such that the resulting coarse grid matrix is as sparse as possible (see e.g. [3] for an algorithm to this end). Fig. 5.7 shows a one-dimensional hierarchical basis.

5.5 A sparsity pattern analysis

In this section, the sparsity pattern of a matrix is defined with the use of some of the definitions from section 5.2 and some elementary results, valid for grid refinement methods satisfying the conditions (5.2.1) up to (5.4.2), are proved. Next, these results are used to examine the sparsity patterns structure for a general case. Finally, the general result obtained is considered in more detail for the regular and newest vertex bisection refinement.

Consider the definition of the sparsity pattern resulting from a chosen discretization. A node φ_j is said to be *coupled* with node φ_i and $(i,j) \in N \times N$ is said to be a *coupling* iff $D_{\varphi_i} \cap D_{\varphi_j} \neq \emptyset$. The *sparsity pattern* is defined as the set of couplings

$$J = \{(i,j) \in N \times N : \varphi_i, \varphi_j \in \mathcal{F} \wedge D_{\varphi_j} \cap D_{\varphi_i} \neq \emptyset\}\,,$$

depending on the grid geometry and the finite element basis functions used but *not* on the partial differential equation discretized. Note that the sparsity pattern is a symmetric subset of $N \times N$ by definition, whence it suffices to examine the *coupling sets* $C(\varphi_i)$

$$C(\varphi_i) := \{\varphi_j \in \mathcal{F} : D_{\varphi_j} \cap D_{\varphi_i} \neq \emptyset \wedge j \leq i\} \quad \forall_{i=1,\ldots,N}\,.$$

containing all nodes coupled to and of level lower than or equal to $l(\varphi_i)$. Before proceeding to the main analysis, consider the following lemma providing some basic and simple tools.

Lemma 5.5.1 *A grid refinement method satisfying (5.2.1) – (5.4.2) leads to a grid $\mathcal{Q}$ with*

- *The intersection of every $\triangle_1, \triangle_2 \in \mathcal{Q}$ is either one of them, or empty*

$$\forall_{\triangle_1,\triangle_2\in\mathcal{Q}}[\triangle_1 \subset \triangle_2 \vee \triangle_2 \subset \triangle_1 \vee \triangle_1 \cap \triangle_2 = \emptyset]\,. \tag{5.5.1}$$

- *If $\triangle_1, \triangle_2 \in \mathcal{Q}$ and $\triangle_1 \cap \triangle_2 \neq \emptyset$ then*

$$\begin{cases} l(\triangle_1) < l(\triangle_2) \Rightarrow \triangle_1 \supset \triangle_2 \\ l(\triangle_1) = l(\triangle_2) \Rightarrow \triangle_1 = \triangle_2 \\ l(\triangle_1) > l(\triangle_2) \Rightarrow \triangle_1 \subset \triangle_2 \end{cases}. \tag{5.5.2}$$

- *All triangles $\triangle, \tilde{\triangle} \in \mathcal{Q}$ satisfy*

$$\begin{cases} l(P^{l(\triangle)-i}(\triangle)) = i \quad \forall_{i=0,1,\ldots,l(\triangle)} \\ \triangle \subset \tilde{\triangle} \Leftrightarrow P^{l(\triangle)-l(\tilde{\triangle})}(\triangle) = \tilde{\triangle} \end{cases}. \tag{5.5.3}$$

Proof. The results above follow directly with the use of combinations of the equations (5.2.1) up to (5.4.2). □

Intuitively, considering the definition of the nodes in the previous section, a coupling set of a hierarchically defined node φ will contain nodes defined on ancestors of the triangles belonging the base of φ. For locally refined grids, consisting of a few triangles, this can easily be verified by hand, but in the general case one must exploit some of the basic properties provided in the previous section in order to prove this. Now let $p(\varphi)$ denote the number of parents which created a node φ, note that

$$p(\varphi) = |\{P(\triangle): \triangle \in B_\varphi\}|\,.$$

Furthermore, define the sets D and E for all triangles $\triangle \in \mathcal{Q}$ by

$$\begin{aligned} D(\triangle) &:= \{\mu \in \mathcal{F}: D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) \leq l(\triangle)\} \\ E(\triangle) &:= \{\mu \in \mathcal{F}: D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) = l(\triangle)\}\,. \end{aligned} \tag{5.5.4}$$

Because the coupling set of a level zero node is easy to determine, only coupling sets of created nodes are considered in the following theorem.

Theorem 5.5.1 *For all nodes $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ and all $\tilde{\triangle} \in \mathcal{Q}$ containing D_φ, $C(\varphi)$ is the union of three disjoint sets*

$$C(\varphi) = \bigcup_{\triangle\in B_\varphi} E(\triangle) \cup \bigcup_{\triangle\in B_\varphi} \bigcup_{k=1}^{l(\triangle)-l(\tilde{\triangle})-1} E(P^k(\triangle)) \cup \bigcup_{k=0}^{l(\tilde{\triangle})} E(P^k(\tilde{\triangle}))\,.$$

Proof. Let $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$. Nodes created after φ have a number greater than the number of φ according to (5.4.2), whence no such nodes are added to φ's coupling set after its creation. The creation of new triangles also does not influence the old nodes and couplings whence $C(\varphi)$ can be examined under the assumption that φ is the last node created.

Note that, due to (5.4.2), directly after a node $\varphi_i \equiv \varphi$ has been created, this node has the highest possible node number i, leading to

$$\begin{aligned} C(\varphi_i) &= \{\varphi_j \in \mathcal{F}: D_{\varphi_j} \cap D_{\varphi_i} \neq \emptyset \wedge j \leq i\} \\ &= \{\varphi_j \in \mathcal{F}: D_{\varphi_j} \cap D_{\varphi_i} \neq \emptyset\} \\ &= \{\mu \in \mathcal{F}: D_\mu \cap D_\varphi \neq \emptyset\} \\ &= \bigcup_{\Delta \in B_\varphi} \{\mu \in \mathcal{F}: D_\mu \cap \Delta \neq \emptyset\} \end{aligned}$$

because $D_\varphi = (\cup\{\Delta: \Delta \in B_\varphi\})^\star$. Furthermore, suppose $D_\mu \cap \Delta \neq \emptyset$, then there exists $\tilde{\Delta} \in B_\mu$ such that $\tilde{\Delta} \cap \Delta \neq \emptyset$. Property (5.2.1) now implies $\Delta \subset \tilde{\Delta}$ whence according to (5.5.2) $l(\mu) = l(\tilde{\Delta}) \leq l(\Delta)$. This leads to

$$C(\varphi) = \bigcup_{\Delta \in B_\varphi} \{\mu \in \mathcal{F}: D_\mu \cap \Delta \neq \emptyset \wedge l(\mu) \leq l(\Delta)\} = \bigcup_{\Delta \in B_\varphi} D(\Delta).$$

Now consider the sets $D(\Delta)$. Note first of all that for all $\Delta \in \mathcal{Q} - \mathcal{Q}^{(0)}$ for all $\mu \in \mathcal{F}$

$$\begin{aligned} &D_\mu \cap \Delta \neq \emptyset \wedge l(\mu) \leq l(P(\Delta)) \Leftrightarrow \\ &D_\mu \cap P(\Delta) \neq \emptyset \wedge l(\mu) \leq l(P(\Delta)) \end{aligned} \tag{5.5.5}$$

because for any $\Delta \in \mathcal{Q} - \mathcal{Q}^{(0)}$ and node $\mu \in \mathcal{F}$

$$\begin{aligned} &D_\mu \cap P(\Delta) \neq \emptyset \wedge l(\mu) \leq l(P(\Delta)) \Leftrightarrow \\ &\exists_{\tilde{\Delta} \in B_\mu}[\tilde{\Delta} \cap P(\Delta) \neq \emptyset \wedge l(\tilde{\Delta}) \leq l(P(\Delta))] \Leftrightarrow \\ &\exists_{\tilde{\Delta} \in B_\mu}[P(\Delta) \subset \tilde{\Delta} \wedge l(\tilde{\Delta}) \leq l(P(\Delta))] \Rightarrow \\ &\exists_{\tilde{\Delta} \in B_\mu}[\Delta \subset \tilde{\Delta} \wedge l(\tilde{\Delta}) \leq l(P(\Delta))] \Rightarrow \\ &\exists_{\tilde{\Delta} \in B_\mu}[\tilde{\Delta} \cap \Delta \neq \emptyset \wedge l(\tilde{\Delta}) \leq l(P(\Delta))] \Leftrightarrow \\ &D_\mu \cap \Delta \neq \emptyset \wedge l(\mu) \leq l(P(\Delta)) \end{aligned}$$

due to (5.5.2), (5.4.1) and $\triangle \subset P(\triangle)$. The opposite direction follows with the use of $\triangle \cap \tilde{\triangle} \neq \emptyset \Rightarrow P(\triangle) \cap \tilde{\triangle} \neq \emptyset$. Therefore the relationship between the sets $D(\triangle)$ and $E(\triangle)$ is, according to (5.5.5), given by

$$\begin{aligned} D(\triangle) =& \{\mu \in \mathcal{F} : D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) \leq l(\triangle)\} \\ =& \{\mu \in \mathcal{F} : D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) < l(\triangle)\} \cup \\ & \{\mu \in \mathcal{F} : D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) = l(\triangle)\} \\ =& \{\mu \in \mathcal{F} : D_\mu \cap \triangle \neq \emptyset \wedge l(\mu) \leq l(P(\triangle))\} \cup E(\triangle) \\ =& \{\mu \in \mathcal{F} : D_\mu \cap P(\triangle) \neq \emptyset \wedge l(\mu) \leq l(P(\triangle))\} \cup E(\triangle), \end{aligned}$$

leading to

$$D(\triangle) = D(P(\triangle)) \cup E(\triangle) \tag{5.5.6}$$

whence according to $D(P^{l(\triangle)}(\triangle)) = E(P^{l(\triangle)}(\triangle))$ and relation (5.5.6)

$$D(\triangle) = \bigcup_{k=0}^{l(\triangle)} E(P^k(\triangle)) \text{ and } C(\varphi) = \bigcup_{\triangle \in B_\varphi} \bigcup_{k=0}^{l(\triangle)} E(P^k(\triangle)).$$

Note that the sets $E(P^k(\triangle)), k = 0, 1, \ldots, l(\triangle)$ are mutually disjunct because they contain nodes of different levels.

Now suppose that there exists a triangle $\tilde{\triangle} \in \mathcal{Q}$ such that $D_\varphi \subset \tilde{\triangle}$ then due to (5.5.3), for all $\triangle \in B_\varphi$,

$$P^{l(\triangle)-l(\tilde{\triangle})}(\triangle) = \tilde{\triangle} \Rightarrow P^{l(\triangle)-l(\tilde{\triangle})+k}(\triangle) = P^k(\tilde{\triangle})$$

for all $k = 0, l(\tilde{\triangle})$ whence

$$\begin{aligned} C(\varphi) &= \bigcup_{\triangle \in B_\varphi} D(\triangle) = \bigcup_{\triangle \in B_\varphi} \bigcup_{k=0}^{l(\triangle)} E(P^k(\triangle)) = \bigcup_{k=0}^{l(\triangle)} \bigcup_{\triangle \in B_\varphi} E(P^k(\triangle)) \\ &= \bigcup_{\triangle \in B_\varphi} E(\triangle) \cup \bigcup_{\triangle \in B_\varphi} \bigcup_{k=1}^{l(\triangle)-l(\tilde{\triangle})-1} E(P^k(\triangle)) \cup \bigcup_{k=0}^{l(\tilde{\triangle})} E(P^k(\tilde{\triangle})). \end{aligned}$$

Clearly three different parts can be distinguished. The first part of the union, called head part, consists of a union of sets $E(\triangle)$ with a non-empty intersection containing the nodes created simultaneously with φ.

The second part, called middle part, consists of sets $E(\triangle)$ which may overlap partially but the last part, called tail part, consists of mutually disjoint sets $E(\triangle)$.

Now consider the existence of such a triangle $\tilde{\triangle} \in \mathcal{Q}$. If $p(\varphi) = 1$ then there exists a unique parent $\tilde{\triangle}$ of level $l(\triangle) - 1$ which contains D_φ whence the above formula reduces to

$$C(\varphi) = \bigcup_{\triangle \in B_\varphi} E(\triangle) \cup \bigcup_{k=1}^{l(\tilde{\triangle})} E(P^k(\tilde{\triangle})).$$

If $p(\varphi) = 2$ and if φ is created at a line of level $m > 0$, then clearly there exists a unique ancestor $\tilde{\triangle}$ of level $m - 1$ which contains D_φ whence

$$C(\varphi) = \bigcup_{\triangle \in B_\varphi} E(\triangle) \cup \bigcup_{\triangle \in B_\varphi} \bigcup_{k=1}^{l(\triangle)-l(\tilde{\triangle})-1} E(P^k(\triangle)) \cup \bigcup_{k=0}^{l(\tilde{\triangle})} E(P^k(\tilde{\triangle})).$$

For nodes with two parents, which are created at a line of level 0 the coupling set has only a head and middle part, the tail part has length zero. Note that this derivation is not restricted to the use of triangular elements. □

The theorem above shows that indeed all nodes coupled to a certain given node φ are defined on one of the ancestors of a triangle belonging to the base of φ. In addition, it shows that it is sufficient to take into account only those nodes defined on a vertex of an ancestor $P^k(\triangle)$ which are an element of $E(P^k(\triangle))$ for given k and triangle $\triangle \in B_\varphi$. Obviously the sparsity pattern is structured with respect to the level structure of the grid refinement under consideration.

In order to be able to estimate the number of couplings $|J|$, i.e., the number of matrix entries, consider the sets $E(\triangle)$, for certain $\triangle \in \mathcal{Q}$, in more detail.

Lemma 5.5.2 *For all $\triangle \in \mathcal{Q}$ the set $E(\triangle)$ satisfies*

$$E(\triangle) = \{\mu \in v(\triangle): l(\mu) = l(\triangle)\}$$

whence there exist positive scalars a and b such that for all created nodes $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$

$$|C(\varphi)| \leq al(\varphi) + b$$

the length of the coupling set is bounded linearly by the level.

Proof. Consider the first part and suppose $\mu \in E(\triangle)$. Then $D_\mu \cap \triangle \neq \emptyset$ so there exists a $\tilde{\triangle} \in B_\mu$ such that $\tilde{\triangle} \cap \triangle \neq \emptyset$. Further $l(\tilde{\triangle}) = l(\mu) = l(\triangle)$, according to (5.5.2) resulting in $\triangle = \tilde{\triangle} \in B_\mu$ whence due to (5.4.1) $\mu \in v(\triangle)$. The opposite direction is trivial. According to the characterization of the E sets above $|E(\triangle)| \leq 3$ for all triangles $\triangle \in \mathcal{Q}$. The coupling sets for nodes $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ satisfy, according to (5.5.6) and (5.4.1),

$$\begin{aligned} |C(\varphi)| &= \left| \bigcup_{\triangle \in B_\varphi} \bigcup_{k=0}^{l(\triangle)} E(P^k(\triangle)) \right| \leq \left| \bigcup_{\triangle \in B_\varphi} E(\triangle) \right| + 3p(\varphi)(l(\varphi)+1) \\ &= 3c + 3p(\varphi)(l(\varphi)+1) \end{aligned}$$

whence $a = 3p(\varphi)$ and $b = 3(c + p(\varphi))$ may be taken. □

This lemma provides an upper bound for the number of entries of the hierarchical matrix for a general refinement technique satisfying the conditions in section 5.2. In order to obtain sharper estimates for specific refinement techniques, consider again the sets $E(\triangle)$, first for the regular grid refinement technique considered by Bank and Yserentant in [38]. Their technique distinguishes between three different types of triangle refinement (see fig. 5.1) called *red triangle refinement* of the first and second kind and *green triangle refinement* which is necessary to maintain the compatibility of the grid. Here $|E(\triangle)|$ depends on the triangle: for the red refinement $E(\triangle) = 3$ for the triangle $\triangle$ created in the middle of its parent and $E(\triangle) = 2$ for the triangles $\triangle$ created along the boundaries (see e.g. fig. 5.2). In the green bisection case $E(\triangle) = 1$ for both children as will be demonstrated below.

For the bisection refinement of Mitchell and Rivara, (the support of the created nodes differs from the regular refinement case as is shown in figs. 5.4 and 5.2), $|E(\triangle)| = 1$ for all $\triangle \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$, as is a direct consequence of the lemma below.

Lemma 5.5.3 *The newest vertex grid refinement implies*

(i) A triangle $\triangle$ *of level* $l(\triangle) \geq 1$ *has exactly*
one vertex of level $l(\triangle)$, *its newest vertex* $x_\triangle$
one vertex of level $l(\triangle) - 1$, *the vertex* $x_{P(\triangle)}$ *and*
one vertex of level less than $l(\triangle) - 1$ *if* $l(\triangle) > 1$ *or of level* 0 *if* $l(\triangle) = 1$.

(ii) A compatibly divisible pair of triangles $\triangle_1$ *and* $\triangle_2$ *of level* $l > 1$ *have exactly*
two vertices of level l, *both newest vertices* $x_{\triangle_1}$ *and* $x_{\triangle_2}$
one common vertex of level $l - 1$, $x_{P(\triangle_1)} = x_{P(\triangle_2)}$
one common vertex of level less than $l - 1$

(iii) Each created vertex $x_\triangle$ *is situated on at most 7 different lines of which*
one of level less than $l(\triangle)$, *the base at which* $x_\triangle$ *is created*
two of level $l(\triangle)$, *created together with* $x_\triangle$ *and*
at most four of level $l(\triangle)+1$, *created afterwards eventually.*

(iv) If a vertex $x_\triangle$ *is created at a line* l_m *of level* $m > 0$ *then* $x_{P^{2k}(\triangle)} \in l_m$ *for all* $0 \leq 2k \leq l(x_\triangle) - m$ *where* $l(x_\triangle) - m \geq 0$ *is always even.*

Proof. The first statement is easy to verify for $l(\triangle)$ equal to 1 or 2. Suppose it is true for triangles $\triangle$ of a certain level $l(\triangle) > 1$ and take a triangle $\triangle$ with vertices $x_\triangle$, $x_{P(\triangle)}$ and x with $l(x) < l(\triangle) - 1$. The refinement of this triangle leads to the creation of one vertex y of level $l(\triangle) + 1$ and two children $\triangle_1$ and $\triangle_2$ such that $v(\triangle_1) = \{y, x_\triangle, x_{P(\triangle)}\}$ resp. $v(\triangle_2) = \{y, x_\triangle, x\}$. Clearly the vertices of both children satisfy the conditions posed on their level, whence by induction the statement is proved.

Note that two compatibly divisible triangles $\triangle_1$ and $\triangle_2$ of level k have four vertices always including the newest vertices $x_{\triangle_1}$ and $x_{\triangle_2}$ of level k and the $x_{P(\triangle_1)}$ and $x_{P(\triangle_2)}$ of level $k - 1$. If $x_{P(\triangle_1)} \neq x_{P(\triangle_2)}$ then both triangles must have one vertex of level k and two vertices of level $k - 1$, which is impossible for $k > 1$. For $k = 1$ the statement does not hold, see fig. 5.8 for a counter example.

The third statement follows with induction, analogous to the first, so consider the last statement and suppose that $x_\triangle$ is created at a line l_m. Then, according to (ii) note that $x_{P^2(\triangle)} \in l_m$ if, of course, $l(x_\triangle) \geq m-2$. Subsequent application of this result, taking into account

that on the initial grid all lines and vertices have level 0, leads to (iv). □

Theorem 5.5.1 in combination with lemma 5.5.3(ii) show that, for the newest vertex grid refinement technique the basis functions belonging to $\mathcal{F}(\mathcal{Q}_j^{(k)}) - \mathcal{F}(\mathcal{Q}_j^{(k-1)})$ are *mutually uncoupled.* This means that the resulting block structured hierarchical matrix (see (5.4.2)) will have diagonal blocks that are diagonal matrices. Since this refinement technique leads to a matrix with a simply structured sparsity pattern it is even possible to give a precise upper bound for the number of couplings $|C(\varphi)|$ of a node φ. To this end define the function $1_{\{m=0\}}$ on $\mathbb{N}$ to obtain value 1 if $m = 0$ and 0 otherwise and consider

Theorem 5.5.2 *If $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ is created at a line of level m then for $p(\varphi) = 1$ respectively $p(\varphi) = 2$*

$$\begin{cases} |C(\varphi)| = \ \ l(\varphi) + 3 \\ |C(\varphi)| = \frac{3}{2}l(\varphi) + 3 - \frac{1}{2}m + (1 - \frac{1}{2}\mathrm{mod}(l(\varphi), 2))1_{\{m=0\}}\,. \end{cases} \tag{5.5.7}$$

Proof. Let $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ have one parent whence automatically $m = 0$. Because φ is a created node, there exists a triangle $\triangle \in \mathcal{Q} - \mathcal{Q}^{(0)}$ such that $\varphi \equiv \varphi_\triangle$. Defining the *path* of this node by the row of nodes

$$\varphi_\triangle, \varphi_{P(\triangle)}, \varphi_{P^2(\triangle)}, \ldots, E(P^{l(\triangle)}(\triangle)) \tag{5.5.8}$$

then according to theorem 5.5.1 and lemma 5.5.2 the coupling set $C(\varphi)$ will exactly contain all elements of this row. All nodes in the coupling set are of different level, $|E(\triangle)| = 1$ for all $\triangle \in \mathcal{Q} - \mathcal{Q}^{(0)}$ according to the previous lemma and the nodes on the coarsest grid are defined standard nodally leading to the desired result.

If φ has two parents then according to theorem 5.5.1 $C(\varphi)$ contains exactly φ itself, at most $2 = p(\varphi)$ ancestors at each level $0 < k < l(\varphi)$ and at most 4 nodes of level 0. Lemma 5.5.2(iv) shows that for a node with two different parents $\triangle_1, \triangle_2 \in \mathcal{F}$ the middle part of the coupling set will have several overlapping sets $E(P^{2k}(\triangle_i))$. The elementary counting necessary to obtain the second result above is left to the interested reader. Note that for this type of refinement $|C(\varphi)| \leq \frac{3}{2}l(\varphi) + 4$ for all nodes $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$. □

Some graphs of paths of nodes of the grid in fig. 5.10 are provided in fig. 5.12. For this grid, and that shown in fig. 5.11, theorem 5.5.2 can easily be verified. For instance, figure 5.12 shows a graph of C(19), denoting the coupling set $C(\varphi_{19})$, which is in turn a union of the paths

$$\{\varphi_{19}, \varphi_{18}, \varphi_{14}, \varphi_{11}, \varphi_7, \varphi_6, \varphi_4, \varphi_3, \varphi_2, \varphi_1\},$$

and

$$\{\varphi_{19}, \varphi_{17}, \varphi_{14}, \varphi_{10}, \varphi_7, \varphi_5, \varphi_4, \varphi_3, \varphi_2, \varphi_1\}.$$

As there are at most two paths a graph is shown starting at the head, showing the middle part (with overlaps) and the tail. Note that the first elements of the path of a vertex are not always the vertex itself and the vertices of its parent(s). This is shown for the first path of $C(\varphi_{19})$ above, where parent vertex $\boldsymbol{x}_7$ is visited after non-parent vertices $\boldsymbol{x}_{10}$ and $\boldsymbol{x}_{11}$. However, the theorem assures that by following the path all coupled nodes, also those defined at the vertices of the parent(s) will be visited eventually. The bisection refinement around a singularity is shown in fig. 6.1 where $|C(\varphi_i)| = i$ for all nodes φ_i situated on the line $y = x$ with $i > 4$. The standard nodal basis sparsity pattern is a subset of that of the hierarchical case due to the definition of the basis functions in both cases.

In this section it has been demonstrated that the sparsity patterns structure only depends on the triangle ancestor hierarchy induced by the refinement strategy chosen. For certain types of refinement, such as the newest vertex refinement technique, it has been shown that one can determine all coupled nodes by simply following paths of ancestor triangles.

5.6 The storage of the hierarchical matrix

As has been shown the sparsity pattern of a matrix is a very irregular but structured subset of $N \times N$ in general. First it will be shown that the set of couplings J can be stored efficiently due to this structure and thereafter the number of the matrix entries $|J|$ is computed for two examples.

The ordinary row-wise ordered storage scheme can be described with the use of a map $\hat{M}: J \to \mathbb{N}$ as follows. For each $(i,j) \in J$ the number j is stored in a row of integers at position $\hat{M}(i,j) = g(i) + f(j)$ where f and g are defined by

$$\begin{cases} g(1) = 0 \\ g(i) = g(i-1) + |C(\varphi_{i-1})|, \; i \geq 2 \end{cases} \qquad \text{and } f(j) \in \{1, \ldots, |C(\varphi_i)|\}$$

for all $i,j \in \{1,\ldots,N\}$. Consider in the following the newest vertex bisection refinement, without loss of generality. In this case, according to theorem 5.5.1 and 5.5.2, a node φ_i maximally is coupled to itself, two nodes of level m for all $0 < m < l(\varphi_i)$ and four nodes of level 0. A special row-wise ordered storage scheme can be defined with the use of the map $M: J \to \mathbb{N}$, $M(i,j) = g(i) + f(j) + 3$ where f and g are defined by

$$\begin{cases} g(1) = 1 \\ g(i) = g(i-1) + 2l(\varphi_{i-1}) + 4, \; i \geq 2 \end{cases},$$

and

$$\begin{aligned} f(j) \; &\in \{-3,-2,-1,0\} && \text{if} \quad 0 = \; l(\varphi_j) \\ &\in \{2l(\varphi_j) - 1, 2l(\varphi_j)\} && \text{if} \quad 0 < \; l(\varphi_j) \\ &= \{2l(\varphi_i) - 1\} && \text{if } l(\varphi_j) = \; l(\varphi_i) \end{aligned}$$

for all $i,j \in \{1,\ldots,N\}$. As an example consider fig. 5.13 for the grid in fig. 5.10, where for each node φ_i separately the corresponding piece of the row is shown in a table, taking $g(i) = 1$ for all i for simplicity. For instance, fig. 5.13 shows in row 15 that node φ_{15} is coupled to nodes φ_1, φ_2 and φ_3 of level 0, to node φ_4 of level 1, etc.

The length of the coupling sets of nodes of level zero is typically 5 for a uniform coarse triangulation of the unit-square. In order to see this consider e.g. [3] or fig. 5.15 where the vertices are numbered from bottom to top in a left to right manner. The nodes with 9 couplings have only 4 couplings with nodes of lower number and are coupled to themselves. Hence, if the scalar 4 in the definition of g of the map M is changed to 5, and if the level of all nodes of a standard nodal base is set to zero, then this storage scheme can also be used for standard nodal basis functions.

The advantage of the special scheme is that for created nodes φ_i, φ_j checking whether $(i,j) \in J$ only involves at most one 'if...then'

instruction contrary to at most $\log_2 |C(\varphi_i)|$ of such instructions in the ordinary case. The disadvantage is that $|M(J)| > \left|\hat{M}(J)\right| = |J|$ but one can consider the computing speed gained more important than the loss of some computer memory.

The number of possible non-zero entries of the matrix resulting from the discretization is for $N^{(k)} = N_j^{(k)}$ equal to

$$|J| \equiv |J_k| = 2 \cdot \sum_{i=1}^{N^{(k)}} |C(\varphi_i)| - N^{(k)} =: 2E_k - N^{(k)} \tag{5.6.1}$$

according to the definition of C. Therefore J is determined by E_k.

Theorem 5.6.1 *If the initial coarse grid is as shown in the first picture of fig. 5.14 then*

- *after k successive uniform newest vertex bisection refinement steps shown in fig. 5.14*

$$N^{(k)} = \begin{cases} (2^{\frac{k}{2}} + 1)^2 & \text{if } k \text{ even} \\ (2^{\frac{k-1}{2}} + 1)^2 + 2^{k-1} & \text{elsewise,} \end{cases}$$
$$E_k = (c_1 + ak)2^k + (c_2 + c_3 ak)(2^k)^{\frac{1}{2}} + c_4$$
$$\lim_{k\to\infty} \frac{E_k}{k2^k} = a \Rightarrow E_k \sim aN^{(k)} \log N^{(k)}, \quad k \to \infty$$

  for some scalars c_i independent of k.

- *after k successive newest vertex bisection refinement steps around the corner $(0,0)$ as shown in fig. 6.1*

$$N^{(k)} = \begin{cases} \frac{3}{2}k & \text{if } k \text{ even} \\ \frac{3}{2}k - \frac{1}{2} & \text{elsewise,} \end{cases}$$
$$E_k = \tfrac{3}{4}ak^2 + \tfrac{1}{4}(12b - a)k$$
$$\lim_{k\to\infty} \frac{E_k}{k^2} = \tfrac{3}{4}a \Rightarrow E_k \sim \tfrac{3}{4}a(N^{(k)})^2, \quad k \to \infty .$$

Proof. As there exist positive scalars a and b such that $|C(\varphi)| \leq al(\varphi)+b$ for every node $\varphi \in V$ the number E_k is bounded by

$$\begin{aligned} E_k &= \sum_{i=1}^{N^{(k)}} |C(\varphi_i)| \leq \sum_{i=1}^{N^{(k)}} al(\varphi) + b \\ &= \sum_{i=1}^{k} (N^{(i)} - N^{(i-1)})(ai + b) + bN^{(0)} \\ &\leq \sum_{i=1}^{k} (N^{(i)} - N^{(i-1)})(ak + b) + bN^{(0)} \\ &= (ak + b)N^{(k)} - akN^{(0)} \end{aligned}$$

leading to $|J| = O(kN^{(k)})$, $k \to \infty$. Now $N^{(k)}$ and E_k are determined for the two cases above.

Note that the number of vertices after k steps of uniform newest vertex bisection of the unit-square is equal to the number of vertices after $k/2$ steps of uniform regular refinement for k even (compare fig. 5.14 and fig. 5.15), so for k even the formula for $N^{(k)}$ is straightforward to derive. For k odd the formula is obtained by taking differences from adjacent even levels. Then E_k is calculated exactly by taking the sum over the coupling sets as in (5.6.1). Note that for k even $k = \log_2((\sqrt{N^{(k)}} - 1)^2)$ leading to an asymptotic behaviour of $E_k \sim aN^{(k)} \log N^{(k)}$.

The formula for $N^{(k)}$ and E_k for the refinement around the origin $(0, 0)$ can easily be derived with the use of fig. 6.1. □

This theorem suggests that the upper bound $O(kN)$ will be somewhere between $O(N \cdot \log_2 N)$ and $O(N^2)$ if the number of refinement levels k is not restricted. If it is bounded then the hierarchical matrix will have $O(N)$ entries. For the newest vertex refinement method $|C(\varphi)| \leq \frac{3}{2}l(\varphi)+4$ for all $\varphi \in \mathcal{F}(\mathcal{Q}) - \mathcal{F}(\mathcal{Q}^{(0)})$ and $|C(\varphi)| \leq 4$ for all $\varphi \in v_0$ due to the regular numbering of the coarse grid leading to $a = \frac{3}{2}$ and $b = 4$ in the theorem above.

As far as the computation of an entry of the hierarchical matrix is concerned note that the matrix can be assembled nodewise in parallel. However, this can lead to memory bank conflicts because different nodes

may have common ancestors. An entry related to a coupling of two nodes of equal level can be computed with the use of an element matrix as usual in finite elements, the fast factorization method proposed in [6] may be used to this end. Note that an advantage of using a hierarchical matrix is that one only needs to assemble the part of the matrix related to the new basis functions $\mathcal{F}(\mathcal{Q}_j^{(k)}) - \mathcal{F}(\mathcal{Q}_j^{(k-1)})$, after the grid $\mathcal{Q}^{(k-1)}$ has been refined, in the case of a Laplacian tensor where the tensor ϵ is elementwise constant on the coarse grid $\mathcal{Q}^{(0)}$.

Matrix vector multiplications can be performed using the parent function P without the map M and they do not involve checks on J. A sparse incomplete Gaussian or Cholesky factorization such as that proposed in [13] is easy to construct for the special storage scheme.

5.7 Block decay rates

In this section it is shown that the block parts $H_{p,q}$ of the hierarchical matrix H, containing the entries of coupled basis functions of levels p and q, have entries decaying in absolute value for $|p-q| \to \infty$. To this end let $|\triangle|$ measure the largest edge of a triangle $\triangle \in \mathcal{Q}$ and define the *grid size parameters* $h^{(k)} := \max_{\triangle \in \mathcal{T}^{(k)}}\{|\triangle|\}$. One then can prove the following lemma.

Lemma 5.7.1 *For the newest vertex bisection respectively the regular refinement*

$$\begin{cases} h^{(k)} = h^{(0)} 2^{-\lfloor \frac{k-1}{2} \rfloor} & k \geq 0, \\ h^{(k)} = h^{(0)} 2^{-k} & k \geq 0. \end{cases} \tag{5.7.1}$$

Proof. Consider for given $\triangle \in \mathcal{Q}$ its four newest bisection similarity classes of children as is shown in fig. 5.5. By an induction argument it follows that triangles of classes 1 & 4 can only be refined into children of classes 2 & 3 and vice versa whence $|\triangle| = \frac{1}{2}|P^2(\triangle)|$ yielding the desired result. The relationship $|\triangle| = \frac{1}{2}|P(\triangle)|$ for the regular refinement as shown in fig. 5.15 is obvious. □

Note that the above result implies that all triangles in class 1 & 4 and 2 & 3 are of even level respectively odd level. Further, for each triangle

$\triangle \in \mathcal{Q}$ with vertices (x_1, y_1), (x_2, y_2) and (x_3, y_3) there exists an affine transformation

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} =: F_\triangle \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \quad (5.7.2)$$

mapping the *reference triangle* defined by $\hat{\triangle} = \{(\hat{x}, \hat{y}) : \hat{x} + \hat{y} < 1, \hat{x} > 0, \hat{y} > 0\}$ onto $\triangle$. Given a basis function on the $\hat{\triangle}$, the corresponding local finite element basis function φ_r on $\triangle$ is defined by $\varphi_r(x, y) = \hat{\varphi}_r(\hat{x}, \hat{y})$ using (5.7.2). Proceeding this way, for every node $\varphi_i \in V$ and $\triangle \in \mathcal{Q}$ either there exists a local basis function φ_r on $\triangle$ such that $\varphi_i = \varphi_r$ on $\triangle$, or $\varphi_i = 0$ on $\triangle$. The formulas for the grid parameters and the definition of the local basis functions above enable the computation of bounds on the partial derivatives of these local basis functions. To this end define the Frobenius norm for an n by n matrix A by

$$\|A\|_F := (\sum_{i,j=1}^{n} a_{ij}^2)^{1/2}$$

and the uniform pointwise partial derivative bound on the reference element by

$$c_\infty := \max_r \sup_{(\hat{x},\hat{y}) \in \hat{\triangle}} \{|\underline{\nabla} \hat{\varphi}_r(\hat{x}, \hat{y})|\}.$$

For the sake of simplicity an *isosceles triangle* $\triangle \in \mathcal{Q}_k$ is a triangle such that $Det(F_\triangle) = (h^{(k)})^2$ and $|f_{ij}| \leq h^{(k)}$.

Lemma 5.7.2 *For all $\triangle \in \mathcal{Q}_k$ and all φ_r, φ_s defined on thereon*

$$\left|(\underline{\nabla} \varphi_s{}^T \underline{\nabla} \varphi_r)(x, y)\right| \leq \tfrac{1}{2} \|F_\triangle^{-1}\|_F^2 \cdot (|\underline{\nabla} \hat{\varphi}_r|^2 + |\underline{\nabla} \hat{\varphi}_s|^2) \leq c_\infty^2 \|F_\triangle^{-1}\|_F^2$$

for all points (x, y) in $\triangle$. If additionally $\triangle$ is isosceles then for all (x, y) in $\triangle$, $|\underline{\nabla} \varphi_r(x, y)| \leq 2c_\infty \cdot (h^{(k)})^{-1}$.

Proof. Note that a partial derivative in the i-th direction of a basis function φ_r is equal to the inner product of column i of the inverse E of the Jacobian matrix of $F_\triangle$ with the column containing all derivatives of the corresponding reference basis function $\hat{\varphi}_r$, i.e.,

$$\frac{\partial \varphi_r}{\partial x} = \frac{\partial \hat{\varphi}_r}{\partial \hat{x}} \frac{\partial \hat{x}}{\partial x} + \frac{\partial \hat{\varphi}_r}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial x} = \begin{bmatrix} e_{11} \\ e_{21} \end{bmatrix}^T \begin{bmatrix} \frac{\partial \hat{\varphi}_r}{\partial \hat{x}} \\ \frac{\partial \hat{\varphi}_r}{\partial \hat{y}} \end{bmatrix} \quad (5.7.3)$$

whence

$$\begin{aligned}&\frac{\partial\varphi_r}{\partial x}\frac{\partial\varphi_s}{\partial x}+\frac{\partial\varphi_r}{\partial y}\frac{\partial\varphi_s}{\partial y}\\ =&(\begin{bmatrix} e_{11}^2 & e_{11}e_{21}\\ e_{21}e_{11} & e_{21}^2\end{bmatrix}+\begin{bmatrix} e_{22}^2 & e_{12}e_{22}\\ e_{12}e_{22} & e_{12}^2\end{bmatrix})\begin{bmatrix}\frac{\partial\hat\varphi_r}{\partial\hat x}\\ \frac{\partial\hat\varphi_r}{\partial\hat y}\end{bmatrix}^T\begin{bmatrix}\frac{\partial\hat\varphi_s}{\partial\hat x}\\ \frac{\partial\hat\varphi_s}{\partial\hat y}\end{bmatrix}\end{aligned}$$

implying that

$$\begin{aligned}\left|(\underline{\nabla}\varphi_s{}^T\underline{\nabla}\varphi_r)\right| &\le \|E\|_{\mathrm{F}}^2|\underline{\nabla}\hat\varphi_r||\underline{\nabla}\hat\varphi_s| \le \tfrac{1}{2}\|E\|_{\mathrm{F}}^2(|\underline{\nabla}\hat\varphi_r|^2+|\underline{\nabla}\hat\varphi_s|^2)\\ &\le c_\infty^2\|E\|_{\mathrm{F}}^2\end{aligned}$$

leading to the first part of lemma 5.7.2. For isosceles triangles $\triangle \in \mathcal{Q}$ by definition $Det(F_\triangle) = 2\cdot Area(\triangle) = (h^{(k)})^2$ and $|f_{ij}| \le h^{(k)}$ leading to $|e_{ij}| \le (h^{(k)})^{-1}$ and $\|F_\triangle^{-1}\|_{\mathrm{F}}^2 \le 4(h^{(k)})^{-2}$ yielding the desired result. □

Now, investigate the magnitude of the entries of the hierarchical matrix. As before, assume that the nodes are numbered in such way that lower level nodes have lower numbers. This leads to

Theorem 5.7.1 *Assume that all triangles in $\mathcal{Q}$, obtained with the newest vertex bisection refinement, are isosceles and assume that piecewise linear hierarchical finite element basis functions are defined on the vertices created. If node $\varphi_i \in \mathcal{F}$ of level p is coupled to a node $\varphi_j \in C(\varphi_i)$ of level $0 \le q \le p$ then*

$$\begin{aligned}[H]_{ij} =&\alpha\int_Q \underline{\nabla}\varphi_j{}^T\underline{\nabla}\varphi_i\,dx+\beta\int_Q\varphi_j\varphi_i\,dx\\ \le&48\alpha\cdot h^{(p)}(h^{(q)})^{-1}+\frac{8}{3}\beta(h^{(p)})^2\end{aligned}\qquad(5.7.4)$$

for all positive scalars α and β. For piecewise quadratic hierarchical basis functions a similar result holds.

Proof. Assume $\varphi_j \in C(\varphi_i)$, $l(\varphi_i) = p \geq q = l(\varphi_j)$ and $\beta = 0$. Note that the integration over the domain Q is reduced to the integration over all triangles $\triangle$ within the set B_{φ_i}, so consider first the contribution to $[H]_{ij}$ from the integration over one of the triangles $\triangle$. Because $\varphi_j \in C(\varphi_i)$, there exists a triangle $\triangle_2 \in B_{\varphi_j}$ such that $\triangle \subset \triangle_2$. As φ_j is defined at one of the vertices of $\triangle_2$, its restriction to $\triangle$ is a linear combination of the local basis functions φ_s defined at $\triangle$

$$\varphi_{j|\triangle} = \sum_s \varphi_j(N_s)\varphi_s \,.$$

Analogously there exists a number r such that $\varphi_{i|\triangle} = \varphi_r$, whence for an arbitrary point $\boldsymbol{x} \in \triangle$

$$\begin{aligned}
\left| \int_\triangle \underline{\nabla}\varphi_j{}^T \underline{\nabla}\varphi_i \, d\boldsymbol{x}dt \right| &= \left| \int_\triangle \sum_{s=1}^{3} \varphi_j(N_s) \underline{\nabla}\varphi_s{}^T \underline{\nabla}\varphi_r \, d\boldsymbol{x}dt \right| \\
&= \left| \int_\triangle \sum_{s=1}^{3} (\varphi_j(N_s) - \varphi_j(\boldsymbol{x})) \underline{\nabla}\varphi_s{}^T \underline{\nabla}\varphi_r \, d\boldsymbol{x}dt \right| + 0 \\
&\leq \sum_{s=1}^{3} |(\varphi_j(N_s) - \varphi_j(\boldsymbol{x}))| \cdot \left| \int_\triangle \underline{\nabla}\varphi_s{}^T \underline{\nabla}\varphi_r \, d\boldsymbol{x}dt \right| \\
&= \sum_{s=1}^{3} |(N_s - \boldsymbol{x})^T \underline{\nabla}\varphi_j(\boldsymbol{x})| \cdot \left| \int_\triangle \underline{\nabla}\varphi_s{}^T \underline{\nabla}\varphi_r \, d\boldsymbol{x}dt \right| \\
&\leq h^{(p)} \cdot c_\infty \|F_{\triangle_2}^{-1}\|_{\mathrm{F}} \sum_{s=1}^{3} \int_\triangle |\underline{\nabla}\varphi_s{}^T \underline{\nabla}\varphi_r| \, d\boldsymbol{x}dt \\
&\leq c_\infty h^{(p)} \|F_{\triangle_2}^{-1}\|_{\mathrm{F}} \sum_{s=1}^{3} \int_{\tilde{\triangle}} c_\infty^2 \|F_\triangle^{-1}\|_{\mathrm{F}}^2 |Det(F_\triangle)| \, d\tilde{\triangle} \\
&\leq \frac{3}{2} c_\infty^3 h^{(p)} \cdot \|F_{\triangle_2}^{-1}\|_{\mathrm{F}} \|F_\triangle^{-1}\|_{\mathrm{F}}^2 |Det(F_\triangle)|
\end{aligned}$$

because the basis functions are linear and the reference element matrix A

$$A = \frac{1}{2} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad [A]_{rs} = \int_{\hat{\triangle}} \underline{\nabla}\hat{\varphi}_s^T \underline{\nabla}\hat{\varphi}_r \, d\boldsymbol{x}$$

has zero row sums. If all $\triangle \in \mathcal{Q}$ are isosceles then

$$\|F_{\triangle_2}^{-1}\|_F \|F_{\triangle}^{-1}\|_F^2 |Det(F_{\triangle})| \leq 8(h^{(q)})^{-1}$$

because of lemma 5.7.2, whence for the newest vertex bisection refinement

$$|[H]_{ij}| \leq 12c_\infty^3 |B_{\varphi_i}| \cdot h^{(p)}(h^{(q)})^{-1} \leq 48h^{(p)}(h^{(q)})^{-1}$$

since $|B_{\varphi_i}| \leq 4$ for all nodes $\varphi_i \in V$ and $c_\infty \leq 1$ for the local linear finite element basis functions. For discretizations with $\beta \neq 0$ note that for piecewise linear basis functions there exists r, s such that

$$\left| \int_{\triangle} \varphi_i \varphi_j \, dx dt \right| = \int_{\tilde{\triangle}} |\hat{\varphi}_r \hat{\varphi}_s| \cdot |Det(F_{\triangle})| \, d\tilde{\triangle} \leq \frac{2}{3}(h^{(p)})^2$$

because of the reference element matrix

$$A = \frac{1}{6} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, [A]_{rs} = \int_{\tilde{\triangle}} |\hat{\varphi}_s \hat{\varphi}_r| \, d\tilde{\triangle}.$$

Note that analogous results are valid for other than piecewise linear basis functions and other grid refinement methods. In the regular refinement case similar results can be obtained. □

For the Dirichlet equation on the uniform refined unit-square many entries corresponding to couplings in the sparsity pattern are zero which is explained for the case of piecewise linear basis functions by

Theorem 5.7.2 *Let node $\varphi_j \in \mathcal{F}$ be coupled to node node $\varphi_i \in \mathcal{F}$. If $p(\varphi_i) = 2$ and $D_{\varphi_i} \subset \triangle$ for some $\triangle \in B_{\varphi_j}$ then*

$$[H]_{ij} = \int_Q \underline{\nabla} \varphi_j{}^T \underline{\nabla} \varphi_i \, dx = 0$$

for the newest vertex refinement method.

Proof. Note that for the newest vertex bisection refinement triangles of classes 1 & 4 can only create children of classes 2 & 3 and vice versa. By an induction argument it follows that the two parents must be of the same similarity class whence the affine transformation of one of them maps the union of both onto the unit-square, and maps φ_i to a node $\hat{\varphi}$ piecewise linear on the unit-square and defined at point $(\frac{1}{2}, \frac{1}{2})$. This node has value 1 at the center of the unit-square, vanishes at its boundary, and is linear on all 4 triangles bordered by the unit-square and the lines $y = x$ and $y = 1 - x$.

For both directional derivatives, the integral over the unit-square of this piecewise linear node is equal to zero This is easy to verify, taking into account that the restriction of $\hat{\varphi}$ to the four triangle parts is $2y$, $2(1 - x)$, $2(1 - y)$, and $2x$ respectively. The above mentioned mapping applied to the function φ_j leads to a piecewise linear image on the unit-square. As this image has a constant gradient, the theorem above follows. □

5.8 The C.-B.-S. scalar for the Laplacian equation

In section 5.5 it has been shown that frequently used grid refinement techniques lead to a hierarchical matrix with a structured sparsity pattern. However, in order to exploit this sparsity pattern structure for the construction of efficient multi-level preconditioners, the grid refinement techniques considered must satisfy additional conditions. For the regular refinement method these criteria, related to the angles in the grid under consideration, have already been studied in the literature (see below), but for the newest vertex bisection case very little seems to be known.

For the sake of completeness therefore, this section will quote well-known estimates for the regular refinement method and will derive analogous results for the newest vertex bisection case. Considered are angle bounds and the computation of the grid geometry dependent Cauchy-Buniakowskii-Schwarz scalar γ^2. This latter scalar determines the rate of convergence of algebraic multi-level preconditioners as considered in [7] and [8]. As is well-known, the angle bounds are of great importance for the determination of the discretization error. In order to obtain reasonable discretization error estimates the angles should be bounded below and away from π.

First consider the role of the scalar γ^2. Discretizing a linear partial differential equation using finite element basis functions defined on an underlying grid geometry leads (see section 5.9) to a system of linear equations

$$H_k x_k = b_k \tag{5.8.1}$$

where the matrix and vector representation depend on the type of finite element basis considered. Suppose a two-level hierarchical base (see [7], [8], [11] and [20] for multi-level methods) is used whence, due to the two-level block structure, the system can be written as

$$\begin{aligned} H_k &= \begin{bmatrix} A_{k-1} & H_{12,k} \\ H_{21,k} & H_{22,k} \end{bmatrix} = I_k^T A_k I_k \\ x_k &:= \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix} \quad b_k := \begin{bmatrix} b_{1,k} \\ b_{2,k} \end{bmatrix} \end{aligned} \tag{5.8.2}$$

where

- A_k is the matrix representing the Laplacian operator on the finite element basis defined standard nodally on $\mathcal{V}(\mathcal{Q}^{(k)})$.
- H_k is the matrix representing the Laplacian operator on the standard nodal finite element basis on all vertices of $\mathcal{V}(\mathcal{Q}^{(k-1)})$ and hierarchically on $\mathcal{V}^{(k)}$.
- I_k is the transformation matrix representing the identity operator from the hierarchical to the standard nodal basis as defined above.
- x_k and b_k are the solution and data vector.

The solution of the system $H_k x_k = b_k$ above is split into two parts (see also [2], [4], [10] and [25]) by the static condensation of the hierarchically defined nodes

$$\begin{aligned} S_{k-1} x_{1,k} &= b_{1,k} - H_{12,k} H_{22,k}^{-1} x_{2,k} \\ x_{2,k} &= H_{22,k}^{-1} (b_{2,k} - H_{21,k} x_{1,k}) \end{aligned} \tag{5.8.3}$$

where the *Schur complement* $S_{k-1} = (A_{k-1} - H_{12,k} H_{22,k}^{-1} H_{21,k})$ is a dense matrix in general. The rate of convergence of a PCG method for the solution of the first equation in (5.8.4) with A_{k-1} as a preconditioner depends on the C.-B.-S. scalar $0 < \gamma^2 < 1$ determined by

$$(1 - \gamma^2)(A_{k-1} x, x) \leq (S_{k-1} x, x) \leq (A_{k-1} x, x) \tag{5.8.4}$$

for all $\boldsymbol{x} \in \mathbb{R}^{N_k}$ with $\boldsymbol{x} \in Ker^{\perp}(A_{k-1})$, and since the latter inequality is trivially satisfied (see [7]). Due to

$$(1-\gamma^2)A_{k-1}\boldsymbol{x} = S_{k-1}\boldsymbol{x} \Leftrightarrow (H_{12,k}H_{22,k}^{-1}H_{21,k} - \gamma^2 A_{k-1})\boldsymbol{x} = 0$$

the scalar γ^2 is equal to

$$\max\{\lambda \in \mathbb{R}: \{0\} \neq Ker(B) \subset Ker^{\perp}(A_{k-1})\}, \tag{5.8.5}$$

where $B = H_{12,k}H_{22,k}^{-1}H_{21,k} - \lambda A_{k-1}$. An upper bound is given by $\gamma^2 \leq \max_{\triangle \in \mathcal{Q}}\{\gamma_{\triangle}^2\}$, where $\gamma_{\triangle}^2$ for all triangles $\triangle \in \mathcal{Q}$ is computed with the use of (5.8.5) (see e.g. [7]). The scalar γ^2 is in fact a measure of the cosine of the angle between the span of the nodal and hierarchical basis (see [4]). As it depends on the finite element basis functions used it will be denoted by γ_l^2 and γ_q^2 in the linear resp. quadratic case from now on. First γ^2 is considered for the well-known case of regular refinement, thereafter it will be computed for the bisection case. In order to simplify future notations (ζ, α, β) will denote a triangle with angles ζ, α and β oriented counter-clockwise and the first angle ζ corresponding to the newest vertex.

Lemma 5.8.1 *Consider a triangle $\triangle \in \mathcal{Q}$ with angles α, β and ζ. For the Laplace equation the regular refinement of this triangle in the linear resp. quadratic case leads to*

$$\gamma_l^2 = \frac{3}{4}\gamma_q^2, \quad \gamma_q^2 = \tfrac{1}{2} + \frac{1}{3}\sqrt{d - \frac{3}{4}} \tag{5.8.6}$$

where $d = \cos^2\alpha + \cos^2\beta + \cos^2\zeta$.

Proof. See [25] for the linear and the quadratic case. Note that in the linear case γ^2 always is bounded below $3/4$. □

Theorem 5.8.1 *Consider a triangle $\triangle \in \mathcal{Q}$ with angles α, β and ζ the angle to be bisected. For the Laplace equation the newest vertex bisection of this triangle in the linear resp. quadratic case leads to*

$$\gamma_l^2 = \tfrac{1}{2}\frac{(\tilde{\alpha}+\tilde{\beta})^2}{2+\tilde{\alpha}^2+\tilde{\beta}^2}, \quad \gamma_q^2 = \max\{\tfrac{1}{2}, \gamma_l^2, \gamma_c^2\} \tag{5.8.7}$$

where

$$\gamma_c^2 = \tfrac{1}{2}\frac{(\tilde\alpha+\tilde\beta)^4}{(\tilde\alpha^2+\tilde\beta^2+2)^2+4(\tilde\alpha^2+1)(\tilde\beta^2+1)}\cdot\frac{\tilde\alpha^2+\tilde\alpha\tilde\beta+\tilde\beta^2+3}{\tilde\alpha^2+\tilde\alpha\tilde\beta+\tilde\beta^2+1}$$

and $\tilde\alpha$, $\tilde\beta$ and $\tilde\zeta$ denote the cotangents of the corresponding angles.
Proof. Suppose that a triangle $\triangle \in \mathcal{Q}$ with nodes $\{\varphi_r\}_{r=1}^p$ is bisected such that q additional hierarchically defined nodes are created and define the corresponding hierarchical Laplacian element matrix H by

$$H = \begin{bmatrix} A & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \tag{5.8.8}$$

where $H_{rs} := \int_\triangle \underline{\nabla}\varphi_s^T \underline{\nabla}\varphi_r$ for all $r,s = 1,\ldots,p+q$. Note that for a node defined linearly resp. quadratically on $\triangle$ its restriction to a child will also be linear resp. quadratic and hence is a linear combination of the childs nodes. Therefore all entries of H can be computed with the use of linear combinations of the standard nodally defined block A.

Further, suppose that the angle ζ is bisected into angles δ_1 and δ_2 such that children $(\delta_1,\alpha,\epsilon_1)$ and $(\delta_2,\epsilon_2,\beta)$ are created (see fig. 5.1). Let the tilde accent "~" denote the cotangent corresponding to each angle and note that

$$\tilde\alpha+\tilde\beta > 0 \quad \forall_{0<\alpha+\beta<\pi} \tag{5.8.9}$$

because the cotangent is a strictly decreasing function on $(0,\pi)$ with properties

$$\cot(\pi-x) = -\cot(x), \quad \cot(x+y) = \frac{\cot(x)\cot(y)-1}{\cot(x)+\cot(y)} \tag{5.8.10}$$

for all $x,y \in (0,\pi)$. From now on the linear and quadratic case will be considered separately.

If linear basis functions are used the newest vertex bisection of ζ will create one new node, leading to the 4 by 4 matrix

$$H = \tfrac{1}{2}\begin{bmatrix} (\tilde\alpha+\tilde\beta) & -\tilde\beta & -\tilde\alpha & -(\tilde\alpha+\tilde\beta) \\ -\tilde\beta & (\tilde\beta+\tilde\zeta) & -\tilde\zeta & \tilde\beta \\ -\tilde\alpha & -\tilde\zeta & (\tilde\zeta+\tilde\alpha) & \tilde\alpha \\ -(\tilde\alpha+\tilde\beta) & \tilde\beta & \tilde\alpha & (\tilde\alpha+\tilde\beta+\tilde\delta_1+\tilde\delta_2) \end{bmatrix}.$$

Here the rank of H_{22} equals one, which implies the existence of exactly one generalized eigenvalue λ satisfying the condition above. Because matrix A has exactly one eigenvector $[\,1,1,1\,]^T$ corresponding to the eigenvalue 0, the condition $\subset Ker^{\perp}(A)$ requires the transformation of $H_{12}H_{22}^{-1}H_{21} - \lambda A$ onto a basis which enables the elimination of this eigenvector. Then $T^{-1}(H_{12}H_{22}^{-1}H_{21} - \lambda A)T$, the transformed problem, is solved, where in this case the orthonormal matrix

$$T = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}$$

transforms the standard basis to the basis $((1,1,1),(0,1,0),(0,0,1))$. Note that, by construction of this transformation, the transformed matrix has first row and column equal to zero, whence a 3×3 minor system is left to be solved. This system obviously must have two eigenvalues equal to zero, because the rank of H_{22} is equal to one, and it appears to have a third generalized eigenvalue

$$\gamma_l^2 = \frac{\tilde{\alpha} + \tilde{\beta}}{\tilde{\alpha} + \tilde{\beta} + \tilde{\delta}_1 + \tilde{\delta}_2}\,.$$

In order to check whether $0 < \gamma_l^2 < 1$, the relationships (see also Mitchell [28])

$$\cot(\epsilon_1) = \frac{\sin(\alpha - \beta)}{2\sin(\alpha)\sin(\beta)}\,, \qquad \epsilon_2 = \pi - \epsilon_1$$

lead by (5.8.10) to

$$\tilde{\epsilon}_1 + \tilde{\epsilon}_2 = 0$$

$$\tilde{\epsilon}_1 = \tfrac{1}{2}(\tilde{\beta} - \tilde{\alpha}) \wedge \tilde{\epsilon}_2 = \tfrac{1}{2}(\tilde{\alpha} - \tilde{\beta})$$

$$\tilde{\delta}_1 = \cot(\delta_1) = -\cot(\alpha + \epsilon_1) = -\frac{\tilde{\alpha}\tilde{\epsilon}_1 - 1}{\tilde{\alpha} + \tilde{\epsilon}_1} = \frac{2 - \tilde{\alpha}(\tilde{\beta} - \tilde{\alpha})}{\tilde{\alpha} + \tilde{\beta}}$$

$$\tilde{\delta}_2 = \frac{2 - \tilde{\beta}(\tilde{\alpha} - \tilde{\beta})}{\tilde{\alpha} + \tilde{\beta}}$$

$$\tilde{\delta}_1 + \tilde{\delta}_2 = \frac{4 + (\tilde{\alpha} - \tilde{\beta})^2}{\tilde{\alpha} + \tilde{\beta}}, \quad \tilde{\delta}_1 - \tilde{\delta}_2 = \tilde{\alpha} - \tilde{\beta}\,.$$

The substitution of these latter relations into the previously obtained formula for γ_l^2 will yield the formula as given in the theorem.

In the case of quadratic basis functions defined on the three vertices and the midpoints of the three edges of the parent, the newest vertex bisection leads to the creation of three new nodes (see fig. 5.14). Here H is the 9×9 symmetric matrix with blocks A (see [3]), H_{22} given by

$$\frac{1}{3}\begin{bmatrix} 3(\tilde{\alpha}+\tilde{\beta}) & \tilde{\beta} & \tilde{\alpha} & 0 & -4\tilde{\alpha} & -4\tilde{\beta} \\ \tilde{\beta} & 3(\tilde{\beta}+\tilde{\zeta}) & \tilde{\zeta} & -4\tilde{\zeta} & 0 & -4\tilde{\beta} \\ \tilde{\alpha} & \tilde{\zeta} & 3(\tilde{\zeta}+\tilde{\alpha}) & -4\tilde{\zeta} & -4\tilde{\alpha} & 0 \\ 0 & -4\tilde{\zeta} & -4\tilde{\zeta} & 8d_0 & -8\tilde{\beta} & -8\tilde{\alpha} \\ -4\tilde{\alpha} & 0 & -4\tilde{\alpha} & -8\tilde{\beta} & 8d_0 & -8\tilde{\zeta} \\ -4\tilde{\beta} & -4\tilde{\beta} & 0 & -8\tilde{\alpha} & -8\tilde{\zeta} & 8d_0 \end{bmatrix}$$

$$\frac{8}{3}\begin{bmatrix} d_1+d_2 & -\tilde{\epsilon}_1 & -\tilde{\epsilon}_2 \\ -\tilde{\epsilon}_1 & d_1 & 0 \\ -\tilde{\epsilon}_2 & 0 & d_2 \end{bmatrix}$$

and $3H_{12} = 3H_{21}^T$ defined by

$$\begin{bmatrix} -3(\tilde{\alpha}+\tilde{\beta}) & 0 & 0 \\ -(\tilde{\alpha}+\tilde{\beta}+\tilde{\delta}_1+\tilde{\delta}_2+4\tilde{\epsilon}_1) & 3\tilde{\alpha}-\tilde{\delta}_1+4\tilde{\epsilon}_1 & -(\tilde{\beta}+\tilde{\delta}_2) \\ -(\tilde{\alpha}+\tilde{\beta}+\tilde{\delta}_1+\tilde{\delta}_2+4\tilde{\epsilon}_2) & -(\tilde{\alpha}+\tilde{\delta}_1) & 3\tilde{\beta}-\tilde{\delta}_2+4\tilde{\epsilon}_2 \\ -2(\tilde{\alpha}+\tilde{\beta}-\tilde{\delta}_1-\tilde{\delta}_2+2\tilde{\epsilon}_1+2\tilde{\epsilon}_2) & 6\tilde{\alpha}+3\tilde{\delta}_1+6\tilde{\epsilon}_1 & 6\tilde{\beta}+3\tilde{\delta}_2+6\tilde{\epsilon}_2 \\ 4(\tilde{\alpha}+\tilde{\beta}+\tilde{\delta}_1-\tilde{\delta}_2+\tilde{\epsilon}_1+\tilde{\epsilon}_2) & -4\tilde{\epsilon}_1 & -8\tilde{\beta}+4\tilde{\epsilon}_2 \\ 4(\tilde{\alpha}+\tilde{\beta}-\tilde{\delta}_1+\tilde{\delta}_2+\tilde{\epsilon}_1+\tilde{\epsilon}_2) & -8\tilde{\alpha}+4\tilde{\epsilon}_1 & -4\tilde{\epsilon}_2 \end{bmatrix}$$

where $d_0 = \tilde{\alpha}+\tilde{\beta}+\tilde{\zeta}$, $d_1 = \tilde{\alpha}+\tilde{\delta}_1+\tilde{\epsilon}_1$ and $d_2 = \tilde{\beta}+\tilde{\delta}_2+\tilde{\epsilon}_2$. The substitution of the expressions derived for $\tilde{\epsilon}_1, \tilde{\epsilon}_2, \tilde{\delta}_1, \tilde{\delta}_2$ into the elements of the 3 by 3 block H_{22} leads to

$$Det(H_{22}) = (\frac{8}{3})^3(\tilde{\alpha}+\tilde{\beta})^{-3}(\tilde{\alpha}^4+\tilde{\beta}^4+6\tilde{\alpha}^2\tilde{\beta}^2+8(1+\tilde{\alpha}^2+\tilde{\beta}^2))(2+\tilde{\alpha}^2+\tilde{\beta}^2)$$

showing that this block is invertible and of rank 3 for all angles $0 < \alpha + \beta < \pi$, whence there may be at most three generalized eigenvalues λ which satisfy $H_{12}H_{22}^{-1}H_{21} - \lambda A = 0$. Matrix A has an eigenvector $[\,1,1,1,1,1,1\,]^T$ corresponding to eigenvalue 0, so in this case T will transform the standard basis into the basis ((1, 1, 1, 1, 1, 1),

$(0,1,0,0,0,0), \ldots, (0,0,0,0,0,1)$). Then $T^{-1}(H_{12}H_{22}^{-1}H_{21} - \lambda A)T$ is a 6×6 matrix with first row and column equal to zero, so λ can be determined by computing the determinant of the remaining 5×5 minor, factorizing it, setting it equal to 0 and solving the equation thus obtained for λ. This yields two eigenvalues 0, as to be expected, and exactly the three positive eigenvalues mentioned in (5.8.7). □

In contrast to the regular refinement, where the scalar γ^2 is a function of the sum of the squares of the cosines of the three angles of the triangle, in the newest vertex bisection method this is not the case, as is shown by

Theorem 5.8.2 *For the newest vertex bisection refinement of a triangle with an angle $\zeta \in (0,\pi)$ to be bisected*

(i) γ_l^2 *and* γ_c^2 *do not depend on the parameter* $d = \cos^2\alpha + \cos^2\beta + \cos^2\zeta$ *as they do for the regular refinement.*

(ii) γ_l^2 *is a function of* ζ *satifying*

$$0 < \min\{\tfrac{1}{2}, \sin^2(\tfrac{1}{2}\zeta)\} \leq \gamma_l^2(\zeta) \leq \max\{\tfrac{1}{2}, \sin^2(\tfrac{1}{2}\zeta)\} < 1$$

for all $\zeta \in (0,\pi)$ *and hence is bounded away from* 0 *and* 1 *independent of the other angles.*

(iii) γ_c^2 *is bounded away from* 1 *by*

$$0 < \gamma_c^2 < \frac{3c^2}{3c^2 + 4c + 1} < 1 \quad \forall_{0<\alpha+\beta<\pi}$$

where $c = \frac{1}{4}(\cot(\alpha) + \cot(\beta))^2$.

Proof. First consider (i). Note that for a triangle $(\frac{1}{6}\pi, \frac{1}{3}\pi, \frac{1}{2}\pi)$ $d = 1$ and $\gamma_l^2 = \frac{1}{14}$ but that for $(\frac{1}{3}\pi, \frac{1}{2}\pi, \frac{1}{6}\pi)$ $d = 1$ and $\gamma_l^2 = \frac{3}{10}$. In the quadratic case for the triangle $(\frac{1}{2}\pi, \frac{1}{4}\pi, \frac{1}{4}\pi)$ $d = 1$ and $\gamma_c^2 = \frac{3}{8}$ but for $(\frac{1}{2}\pi, \frac{1}{3}\pi, \frac{1}{6}\pi)$ $d = 1$ and $\gamma_c^2 = \frac{11}{28}$.

Now consider (ii). For each triangle (ζ, α, β) with $\zeta \in (0,\pi)$ and $\alpha \leq \beta$ there exists a positive angle $\xi \in [0, \frac{1}{2}(\pi - \zeta))$ such that

$$\alpha = \frac{1}{2}(\pi - \zeta) - \xi \text{ and } \beta = \frac{1}{2}(\pi - \zeta) + \xi\,.$$

Here ξ is a measure for the deformation of the triangle, which will be isosceles for $\xi = 0$ and degenerated to a line when ξ reaches $\frac{1}{2}(\pi - \zeta)$. For $\alpha > \beta$ reverse the roles of α and β.

The substitution of the relations $\tilde{\alpha} = \cot(\frac{1}{2}(\pi - \zeta) - \xi)$ and $\tilde{\beta} = \cot(\frac{1}{2}(\pi - \zeta) + \xi)$ into (5.8.7) and setting

$$\begin{cases} \zeta \in (0, \pi) & x = \cos^2(\frac{1}{2}(\pi - \zeta)) \Rightarrow x \in (0, 1) \\ \xi \in [0, \frac{1}{2}(\pi - \zeta)) & y = \cos^2(\xi) \Rightarrow y \in (x, 1) \end{cases}$$

yields the formula

$$\gamma_l^2 = \frac{x(1-x)}{x + y(1-2x)} \quad \forall_{x \in (0,1), y \in (x,1)}.$$

An investigation of this formula in order to determine its minimum and maximum in y for a given x, shows that for all $x \in (0, 1)$ the denominator will be positive and a linear function of y. For $x \in (0, \frac{1}{2})$, γ_l^2 will be a decreasing function of y, whereas for $x \in (\frac{1}{2}, 1)$ γ_l^2 will be an increasing function with respect to this argument. For $x = \frac{1}{2}$, $\gamma_l^2 = \frac{1}{2}$, independently of y. Therefore, in the first two cases the bounds are provided by the extreme values $y = x$ and $y = 1$ corresponding to a triangle degenerated to a line and an isosceles triangle, leading to

$$\min\{\tfrac{1}{2}, x\} \leq \gamma_l^2 \leq \max\{\tfrac{1}{2}, x\}.$$

Finally the substitution of $x = \cos^2(\frac{1}{2}(\pi - \zeta)) = \sin^2(\frac{1}{2}\zeta)$ yields the desired result. In the special case where $\zeta = \frac{1}{2}\pi$

$$\gamma_l^2(\tfrac{1}{2}\pi) = \tfrac{1}{2} \quad \forall_{0 < \alpha + \beta < \pi}$$

for all other angles α and β.

For a proof of (iii) note that the formulas in (5.8.7) are symmetric in $\tilde{\alpha}$ and $\tilde{\beta}$. In order to eliminate the cross terms $\tilde{\alpha}\tilde{\beta}$ in these formulas substitute the following transformation to the axis of symmetry $\tilde{\alpha} - \tilde{\beta} = 0$

$$\begin{bmatrix} \tilde{\alpha} \\ \tilde{\beta} \end{bmatrix} = \sqrt{c} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \sqrt{x} \begin{bmatrix} 1 \\ -1 \end{bmatrix} \text{ with } \begin{bmatrix} c \\ x \end{bmatrix} = \frac{1}{4} \begin{bmatrix} (\tilde{\alpha} + \tilde{\beta})^2 \\ (\tilde{\alpha} - \tilde{\beta})^2 \end{bmatrix}$$

yielding for all $x, c \in [0, \infty)$

$$0<\gamma_l^2(\tilde{\alpha},\tilde{\beta})=\frac{c}{1+x+c}\leq\frac{c}{1+c}<1$$
$$0<\gamma_c^2(\tilde{\alpha},\tilde{\beta})=\frac{(3+x+3c)c^2}{(1+x+3c)(1+2x+x^2+2c+c^2)}leq\frac{3c^2}{3c^2+4c+1}<1$$

because both γ_l^2 and γ_c^2 are decreasing functions of x on $[0, \infty)$ for fixed c, which attain their maximum at $x = 0$. □

The scalars $\gamma_\triangle^2$ now can be computed for arbitrary $\triangle \in \mathcal{Q}$ leading to a bound for the C.-B.-S. scalar γ^2 as below (5.8.5). For the newest vertex bisection refinement method the maximum over all triangles can be replaced by the maximum over $4\mathcal{Q}_0$ triangles because of the fact that each coarse grid triangle has exactly four similarity classes of children, as is shown in fig. 5.5 (fig. 5.1 shows the congruent children for the first red regular refinement). For an arbitrary $\triangle \in \mathcal{Q}_0$ given by (ζ, α, β) these four classes of children defined by eight different angles are (see fig. 5.6)

$$(\zeta, \alpha, \beta),\ (\epsilon_1, \delta_1, \alpha),\ (\epsilon_2, \beta, \delta_2),\ (\pi - \zeta, \delta_2, \delta_1)\,.$$

The C.-B.-S. scalar upper bound therefore can be computed for an arbitrary refinement by considering only the coarse grid triangles, see section 5.9 for an example. Due to the similarity class property also the angle bounds only depend on the coarse grid geometry whence a properly chosen coarse grid automatically avoids bad angles.

5.9 Numerical examples

This section presents the local grid refinement and interpolation between subsequently refined grids in order to introduce a solution method for non-linear time-dependent differential equations. Further, investigating some examples, it is shown that the sparsity pattern is structured in the case of local grid refinement. Finally, at the end of this section, the C.-B.-S. scalar is computed for some example triangles.

First, consider the numerical solution of a time-dependent equation, using the global time-space finite element technique as presented in chapters 1 and 2. As an example, but without loss of generality, look at the following problem. Find $u \in H^1_\gamma(Q)$, such that

$$\begin{aligned} u_t - u_{xx} &= f \quad \text{in } Q \\ u &= \gamma \quad \text{at } \Gamma_D \end{aligned}$$

for smooth boundary data γ on $\Gamma_D = \Gamma_1 \cup \Gamma_c$ and source function f, where γ prescribes the initial value function u_0 as well as the Dirichlet boundary value conditions u_c. In order to solve this problem, let for a given grid $\mathcal{Q}_j^{(k)}$, $\mathcal{V}(\mathcal{Q}_j^{(k)})$ denote the set of vertices of this grid and let $\mathcal{H}(\mathcal{Q}_j^{(k)})$ denote the span of the hierarchically or standard nodally defined finite element basis functions defined on these vertices. Further, define the set of trial functions

$$\mathcal{H}_\gamma(\mathcal{Q}_j^{(k)}) = \{u \in \mathcal{H}(\mathcal{Q}_j^{(k)}) : u(x) = \gamma(x) \text{ at } \mathcal{V}(\mathcal{Q}_j^{(k)}) \cap \Gamma_D\} . \quad (5.9.1)$$

Searching for a discrete solution $\hat{u}_{j,h}^{(k)}$ in $\mathcal{H}_\gamma(\mathcal{Q}_j^{(k)})$ implies that the Dirichlet boundary conditions are only approximated if the function γ is not elementwise linear, or, in the case of higher order finite element basis functions, piecewise polynomial of higher order. This is referred to by *variational crime* by Strange and Fix in [36].

For a given grid refinement technique and interpolation between subsequent grids, the *regridded damped inexact Newton iterative method RDIN* is given by the following solution algorithm

$k = 0$; $j = 1$
While $j \leq J$
Do
  While $k \leq K_j$
  Do
    Construct grid $\mathcal{Q}_j^{(k)}$;
    Construct interpolant $u_{j,I}^{(k)} \in \mathcal{H}_\gamma(\mathcal{Q}_j^{(k)})$;
    Find $\hat{u}_{j,h}^{(k)}$ *such that* $F(\hat{u}_{j,h}^{(k)}) = 0$
    with Newtons method,
    using $u_{j,I}^{(k)}$ *as start approximation*;

$k := k + 1$
Od
$k := 0$
$j := j + 1$
Od

for a given total number of time-slabs J and a given maximal level of subsequent grid refinements K_j per time-slab, chosen to be uniformly equal to K in order to simplify the algorithm. The construction of the grid and interpolant can be found below, the Newton method is discussed in section 8.2.

The construction of the grid and interpolant in the algorithm above is defined with the use of the newest vertex grid bisection refinement technique such that for all $k \geq 0$ and $j \geq 1$ by definition

- Initially, when $k = 0$ and $j = 1$ then $\mathcal{Q}_1^{(0)}$ is taken to be the initial coarse grid and $u_{0,I}^{(0)}$ is taken to be a provided initial coarse start approximation in $\mathcal{H}_\gamma(\mathcal{Q}^{(0)})$.
- If $k > 0$ and $j \geq 1$ then there are several possibilities to construct a new grid $\mathcal{Q}_j^{(k)}$ applying grid refinement to the grid $\mathcal{Q}_j^{(k-1)}$. For the sake of simplicity only three possibilities will be distinguished.
  - In the case of *uniform refinement* every triangle in $\mathcal{Q}_j^{(k-1)}$ is refined.
  - In the case of *line refinement* all triangles without children in the grid $\mathcal{Q}_j^{(k-1)}$ intersecting a predetermined line are refined. This type of refinement can be applied in such cases where one knows the position of boundary or internal layers. One can analogously distinguish *plane refinement* in the three-dimensional case.
  - A third possibility is to make use of the previously computed discrete solution $\hat{u}_{j,h}^{(k-1)}$, called *adaptive refinement.* A triangle of the grid $\mathcal{Q}_j^{(k-1)}$ is refined if and only if the discrete solution $\hat{u}_{j,h}^{(k-1)}$ has too large error on this triangle, the local error being estimated by some *error indicator.* Here, in order to reduce to complexity of the solution algorithm, each triangle of grid $\mathcal{Q}_j^{(k-1)}$ is refined if on this

triangle $\left|\underline{\nabla}_{x,t}\hat{u}_{j,h}^{(k-1)}(x,t)\right| > c$, where c is a predetermined positive scalar. In order to be able to correct the possibly bad predictions of this simple error estimator, the adaptive refinement is combined with *adaptive derefinement.* This means that every group of simultaneously created triangles on previous grids $\mathcal{Q}_j^{(s)}$, $0 \leq s < k$, without own children, will be deleted if on this group of triangles $\left|\underline{\nabla}_{x,t}\hat{u}_{j,h}^{(k-1)}(x,t)\right| \leq c$.

After the grid refinement, the interpolant $u_{j,I}^{(k)} \in \mathcal{H}(\mathcal{Q}_j^{(k)})$ is given by $u_{j,I}^{(k)} = \mathcal{I}^{(k)}\hat{u}_{j,h}^{(k-1)}$, where the interpolation functional $\mathcal{I}^{(k)}: \mathcal{H}_\gamma(\mathcal{Q}_j^{(k-1)}) \mapsto \mathcal{H}_\gamma(\mathcal{Q}_j^{(k)})$ is defined by

$$\begin{cases} (\mathcal{I}^{(k)}u)(x) = & u(x) & \text{if } x \in \mathcal{V}(\mathcal{Q}_j^{(k-1)}) \\ & u(x) & \text{if } x \in \mathcal{V}^{(k)} \wedge x \in Q - \Gamma_D \\ & \gamma(x) & \text{if } x \in \mathcal{V}^{(k)} \wedge x \in \Gamma_D \end{cases}$$

for given function $u \in \mathcal{H}_\gamma(\mathcal{Q}_j^{(k-1)})$. Note that this definition is independent of the finite element basis used.

- Having completed the computations on a previous time-slab, i.e., in the case where $k = 0$, but $j > 1$, an initial coarse grid $\mathcal{Q}_j^{(0)}$ for the new time-slab is constructed by reflecting the grid obtained on the previous time-slab along the grids upper boundary Γ_3 and by deleting all triangles which have no vertex at this line. Further, the new interpolant $u_{j,I}^{(0)} \in \mathcal{H}_\gamma(\mathcal{Q}_j^{(0)})$ is by definition equal to the previous discrete solution $\hat{u}_{j,h}^{(k-1)}$ at the remaining reflected vertices pointwise, except on those vertices lying on a Γ_c. At such vertices the interpolant takes the exact value of the Dirichlet boundary condition function γ. Note that any a priori knowledge concerning the discrete solution on this time-slab can be used to improve this interpolant and grid constructed.

Summarizing, if one uses the RDIN algorithm for the global time-space finite element solution of a non-linear partial differential equation, one has to specify

- the initial coarse grid $\mathcal{Q}^{(0)}$ and initial approximation $u_I^{(0)}$.
- the type of grid refinement, and the threshold c for the error indicator, if adaptive refinement is used.

- the non-linear Newton solution method.

Note that the weighing and streamline upwind finite element methods as proposed in the chapters 2 respectively 3, are only of influence for the assembly of the systems of linear equations to be solved within the Newton solution method. The grid refinement technique is not dependent on the type of finite element basis functions used.

Table 5.9.1 The uniformly refined unit-square.

k	$\lvert\mathcal{T}^{(k)}\rvert$	$\lvert\mathcal{Q}^{(k)}\rvert$	$n^{(k)}$	$N^{(k)}$	E_k (H)	E_k (N)
0	2	2	4	4	14	14
1	4	6	1	5	23	21
2	8	14	4	9	59	41
3	16	30	4	13	115	69
4	32	62	12	25	279	137
5	64	126	16	41	575	249
6	128	254	40	81	1319	497
7	256	510	64	145	2775	945
8	512	1022	144	289	6151	1889
9	1024	2046	256	545	13031	3681

Now two example partial differential equations will be studied. For the first equation, the sparsity pattern of the resulting finite element matrix will be studied in detail, both for the hierarchical and the standard nodal basis. For the second equation, where streamline upwind finite element basisfunctions are used, the sparsity pattern is only considered for the standard nodal basis. For this problem, the functioning of the regridding will be demonstrated in detail.

The first equation is the *Dirichlet equation*, given by

$$\begin{aligned} -\triangle u &= 0 \text{ in } Q \\ u &= \gamma \text{ at } \Gamma_D \end{aligned} \tag{5.9.2}$$

for smooth Dirichlet boundary data γ, taken from the span

$$\mathcal{S} = [\![1, x, y, xy, x^2 - y^2, x^3 - 3xy^2, y^3 - 3yx^2, x^3y - y^3x]\!] . \tag{5.9.3}$$

Here the unit-square will be refined using uniform newest vertex grid refinement starting from an initial coarse grid $\mathcal{Q}^{(0)}$ as in the first picture of fig. 5.14.

Table 5.9.1 gives for $Q^{(0)}$ to $Q^{(9)}$ an overview of the elementary parameters, all defined in section 5.2 to section 5.6. The parameters depending on the type of finite element basis used are followed by '(H)' or '(N)' to distinguish between the hierarchical and standard nodal base.

For this example, taking the Dirichlet boundary condition

$$\gamma = 100 \cdot \{-17 \cdot (x^3y - y^3x) + 13 \cdot (x^3 - 3xy^2) - 11 \cdot (y^3 - 3yx^2) + \frac{2}{10}(x^2 - y^2) - \frac{1}{10}xy + \frac{3}{10}x - \frac{1}{10}y - 1\},$$

figure 5.16 shows $Q^{(10)}$ and fig. 5.17 gives the isoclines of the solution. Then fig. 5.18 and fig. 5.19 present the sparsity pattern for a hierarchical resp. standard nodal basis on this grid geometry and figs. 5.20, 5.21 show the sparsity patterns subset of couplings corresponding to the non-zero matrix entries. In these pictures, those couplings $(i, j) \in J$ satisfying the conditions on h_{ij} are plotted in such a way that $(1, 1)$ is situated in the top-left corner, and (N, N) in the bottom-right corner. The horizontal and vertical lines show the block structure of the hierarchical matrix H, only serving reference purposes in the standard nodal case. The subsets of couplings related to positive matrix entries can be found in figs. 5.22 and 5.23.

In the hierarchical case, according to theorem 5.7.2, many entries are zero, which is demonstrated in fig. 5.20 and proved in theorem 5.7.2. Some off-diagonal entries are positive contrary to the standard nodal case. In the standard nodal case all nodes have by definition level 0 but they are numbered according to the corresponding vertices which are still of different levels. Fig. 5.21 shows that the upper left submatrix of the standard nodal matrix tends to become diagonal for $k \to \infty$, since nodes of lower level get de-coupled due to the uniform refinement.

The set $\mathcal{S}$ is a special set, its first seven elements span exactly the set of all harmonic polynomials of degree less than four. This implies that the solution of the Dirichlet problem is equal to the Dirichlet boundary condition γ on the whole unit-square. Now, suppose that a uniform $m \times m$ isosceles initial coarse triangulation is covering the unit-square. Assume that for given γ, one of the in section 8.2 described iterative solution methods is used for the solution of the resulting systems $F_k' d_k = -F_k$. Then, for all $0 < k \leq 10, m \leq 5$, and initial starting solution $x^{(0)} =$

γ pointwise equal at *all* vertices of $\mathcal{Q}^{(0)}$,

$$\boldsymbol{d}_k = \begin{bmatrix} \boldsymbol{d}_{k,1} \\ \boldsymbol{d}_{k,2} \end{bmatrix} = \begin{bmatrix} 0 \\ \star \end{bmatrix} \text{ and } \|\boldsymbol{x}^{(1)} - \gamma\|_{\infty,\mathcal{Q}^{(k)}} = 0\,, \tag{5.9.4}$$

observed to machine precision. Here '$\star$' denotes possibly non-zero vector components,

$$\|u\|_{\infty,\mathcal{Q}^{(k)}} = \max_{x \in \mathcal{V}(\mathcal{Q}^{(k)})} \{|u(x)|\}\,,$$

and $\boldsymbol{x}^{(1)} = \hat{u}_h^{(k)}$ is the first iterand of the damped Newton algorithm. Interesting is that $\boldsymbol{d}_k = C^{-1}(F'\boldsymbol{x}^{(0)} - F)$ exactly, i.e., all iterative solvers converge in one interation step. Here C_k denotes the pointwise ILU preconditioner for F'. This observation holds both for the standard nodal and hierarchical basis representation, which is remarkable since the pointwise factorization of F_k' is incomplete and F_k' is different for both finite element bases used. This property is not influenced by the way the nodes are numbered, as long as the node numbering reflects the levels of refinement. However, it is destroyed if the initial coarse triangulation is not isosceles.

For the sake of simplicity, denote the Jacobian matrix F_k' by H_k and F_k by $\boldsymbol{f}_k$. Inspired by the special class of Dirichlet problems above, substituting (5.9.4)

$$\hat{\boldsymbol{u}}_k = \begin{bmatrix} \hat{\boldsymbol{u}}_{k-1} \\ \hat{\boldsymbol{u}}_{k,2} \end{bmatrix} \text{ in } H_k\hat{\boldsymbol{u}}_k = \boldsymbol{f}_k\,, \tag{5.9.5}$$

which latter system of equations is equal to that presented in (5.8.2), the following recursive V-cycle multi-level preconditioner C_k is obtained to approximate the discrete solution $\hat{u}_k$

$$\begin{aligned} \boldsymbol{f}_{k-1} &= \boldsymbol{f}_{k,1} - H_{12,k}H_{22,k}^{-1}(\boldsymbol{f}_{k,2} - \{H_{21,k}\hat{\boldsymbol{u}}_{k-1}\}) \\ \hat{\boldsymbol{u}}_0 &= H_0^{-1}\boldsymbol{f}_0 \\ \hat{\boldsymbol{u}}_{k,2} &= H_{22,k}^{-1}(\boldsymbol{f}_{k,2} - H_{21,k}\hat{\boldsymbol{u}}_{k-1})\,, \end{aligned} \tag{5.9.6}$$

since $A_{k-1} = H_{k-1}$ in (5.8.2) for a multi-level hierarchical basis. Under the assumption $H_{k-1}\hat{\boldsymbol{u}}_{k-1} = \boldsymbol{f}_{k-1}$, the first line in the latter system

of three equations is equivalent to the first line in (5.8.3), whence for the special class of Dirichlet problems above, a half V-cycle sweep yields the pointwise exact discrete solution $\hat{u}_k$. A V-cycle multi-level preconditioner converging in one step for problems of this class clearly must use the solutions $\hat{u}_i$ on all previous levels $0 \leq i \leq k$. A little bit of algebra shows that leaving out the part between brackets in (5.9.6) leads to

$$C_k = \begin{bmatrix} C_{k-1} & H_{12,k} \\ 0 & H_{22,k} \end{bmatrix} \circ \begin{bmatrix} I & 0 \\ H_{22,k}^{-1} H_{21,k} & I \end{bmatrix}, \tag{5.9.7}$$

another recursively defined V-cycle multi-level preconditioner C_k. As is shown in [37], this preconditioner has a condition number of order $O(k^2)$.

The second example to be considered is the time-dependent convection-diffusion equation given by

$$\begin{cases} -\epsilon u_{xx} + \hat{\boldsymbol{b}}^T \nabla_{x,t} u = 0 & \text{in } -1 < x < 1, -1 < t \leq \infty \\ u(-1,t) = 1, u(1,t) = 0 & \text{on } -1 \leq t \leq 1 \\ u(x,-1) = 1 & \text{on } -1 < x \leq 0 \\ u(x,-1) = 0 & \text{on } 0 < x < 1, \end{cases} \tag{5.9.8}$$

where $\epsilon = 10^{-4}$ and $\hat{\boldsymbol{b}} = [-\frac{1}{2}\pi \cos(\pi t), 1]^T$. The equations solution has a shock, moving as a cosine in time, so that the grid $\mathcal{Q}_j^{(k+1)}$ is constructed from $\mathcal{Q}_j^{(k)}$ using the adaptive refinement with refinement threshold 4.0. The initial coarse grid is shown in fig. 5.24.

The computed SUPG standard nodal finite element solution and its equidistant isoclines are shown in figures 5.24 – 5.39 for the first two time-slabs. The sparsity pattern of the standard nodal finite element matrix related to $\mathcal{Q}_1^{(12)}$ is plotted in figs 5.40, 5.42 and 5.44. In the case of $\mathcal{Q}_2^{(12)}$ it is shown figs. 5.41, 5.43 and 5.45. In these examples the sparsity pattern locally is very irregular but clearly globally structured. The finite element matrix H is not symmetric, but, for instance, figure 5.44 seems to indicate that the subset of positive entries is symmetric. In all tests so far, the nodes of lower level have lower node numbers, as is required by relation (5.4.2). Note that on the second time-slab, on $\mathcal{Q}_2^{(0)}$, the sparsity pattern of the finite element matrix shows that nodes of

higher level exist. This is due to the regridding introduced at the beginning of this section. The remaining higher level grid points, after the construction of the grid $\mathcal{Q}_2^{(0)}$, have unchanged level. It would have been better to have set the level of all remaining nodes to zero, but the here followed approach is easier to implement.

The figures 5.24 – 5.39 show that the grid refinement strategy only produces angles of $\frac{1}{2}\pi$ and $\frac{1}{4}\pi$, which is certainly not true for the regular refinement strategy used by Bank and Yserentant, and Deuflhard, see for instance [23]. The under- and overshooting (see e.g. 5.37) on $\mathcal{Q}_1^{(16)}$ is approximately 5 percent, decreasing with increasing time-slab number. On grid $\mathcal{Q}_2^{(16)}$ on the second time-slab, it is in the order of a half percent. Note that the slabs 1 and 2 fit perfectly together as is demonstrated in the figures 5.32 and 5.34.

Table 5.9.2 A locally refined grid.

k	N	$n^{(k)}$	$n^{(k,16)}$	E_k	$\neq 0$
0	8	8	8	72	58%
2	21	3	7	189	68%
4	65	32	15	585	76%
6	72	4	27	648	78%
9	136	30	85	1224	84%
12	904	297	332	8136	87%
13	1497	512	527	13473	88%
16	6078	2382	2382	54702	88%

Due to the recursiveness of the bisection algorithm presented in section 5.2, the refinement of a triangle of level k can create several nodes of level $0 \leq i \leq k$. During the uniform refinement of $\mathcal{Q}^{(k)}$ in the first example this is not the case, only nodes of level $k+1$ are created which means that the sets of vertices $\mathcal{V}^{(i)}$, $0 \leq i \leq k$ are not altered after the construction of $\mathcal{Q}^{(k)}$. This is different for the second example, where the sets $\mathcal{V}^{(i)}$ can grow due to the refinement of triangles of level $i \leq k$ as is demonstrated in table 5.9.2 in the fourth column. Here $n^{(k,16)}$ denotes the number of nodes for given level $n^{(k)}$ after 16 levels of refinement. The percentage of non-zero matrix entries from the total number of entries stored in memory can be found in the last column of table 5.9.2. Note that the adaptive refinement and derefinement method works despite the

fact that the Dirichlet boundary conditions are never exactly satisfied, see for example the figures 5.24 and 5.25.

Table 5.9.3 Parameters for problem (5.9.8).

$Q_j^{(k)}$	T	N	N_c	N_d	#It
1,00	6	8	8	0	0
1,02	42	21	10	0	0
1,04	186	65	32	0	1
1,06	214	72	4	45	1
1,09	492	136	62	3	7
1,12	3506	904	538	16	26
1,13	5880	1497	647	54	34
1,16	24166	6078	2545	233	181
2,00	274	92	0	5986	1
2,04	410	135	27	0	3
2,06	420	135	4	49	2
2,12	3038	786	334	28	18
2,13	7298	1862	1096	20	65
2,16	30194	7594	3215	140	205

As table 5.9.3 for the first two time-slabs shows, the computation of the discrete finite element approximation on each time-slab can be done efficiently if the RDIN solution method is used to this end. Since the error estimator presented is cheaply to evaluate, it leads to a cheap and effective adaptive refinement method. On most grids on every time-slab, the ratio of created and deleted triangles is about 15:1. Sometimes a refined triangle is derefined later on. The solution time for slab 1 grids 1-12 is negligible compared to the time used for the solution of 13-16. methods used.

In order to get an impression of the values γ_Δ^2 for some triangles, note that in the case of linear basis functions the four similarity classes mentioned in section 5.8 correspond to the C.-B.-S. scalars (see (5.8.7))

$$\gamma_1^2,\gamma_2^2,\gamma_3^2,\gamma_4^2 = \tfrac{1}{2}\frac{(\tilde\alpha+\tilde\beta)^2}{\tilde\alpha^2+\tilde\beta^2+2}, \frac{\tilde\alpha^2+1}{\tilde\alpha^2+\tilde\beta^2+2}, \frac{\tilde\beta^2+1}{\tilde\alpha^2+\tilde\beta^2+2}, \tfrac{1}{2}\frac{(\tilde\alpha-\tilde\beta)^2+4}{\tilde\alpha^2+\tilde\beta^2+2}$$
$$\equiv \gamma_1^2,\gamma_2^2,1-\gamma_2^2,1-\gamma_1^2 .$$

Table 5.9.4 Scalars γ_l^2 and γ_c^2 for the newest vertex bisection.

$\triangle \backslash \gamma_l^2, \gamma_c^2$	1^{st} *Class*	2^{nd} *Class*	3^{rd} *Class*	4^{th} *Class*
$(\frac{1}{2}\pi, \frac{1}{4}\pi, \frac{1}{4}\pi)$	$\frac{1}{2}, \frac{3}{8}$	$\frac{1}{2}, \frac{3}{8}$	$\frac{1}{2}, \frac{3}{8}$	$\frac{1}{2}, \frac{3}{8}$
$(\frac{1}{3}\pi, \frac{1}{3}\pi, \frac{1}{3}\pi)$	$\frac{1}{4}, \frac{1}{8}$	$\frac{1}{2}, \frac{11}{28}$	$\frac{1}{2}, \frac{11}{28}$	$\frac{3}{4}, \frac{27}{40}$
$(\frac{1}{12}\pi, \frac{2}{3}\pi, \frac{1}{4}\pi)$	0.268,0.00157	$\frac{2}{5}$,0.303	$\frac{3}{5}$,0.668	0.973,0.977
$(\frac{2}{3}\pi, \frac{1}{12}\pi, \frac{1}{4}\pi)$	0.661,0.680	0.882,0.873	0.118,0.305	0.337,0.206
$(\frac{1}{4}\pi, \frac{2}{3}\pi, \frac{1}{12}\pi)$	0.306,0.166	0.0820,0.0130	0.918,0.938	0.649,0.774

Table 5.9.4 gives γ_l^2 and γ_c^2 for some example triangles (ζ, α, β) with ζ the angle to be bisected, computed with the use of (5.8.7). Note that γ_q^2 can be found using the table and the relationship $\gamma_q^2 = \max\{\frac{1}{2}\gamma_l^2, \gamma_c^2\}$. According to table 5.9.4, in the quadratic case it is best to have a largest angle of approximately $\frac{1}{2}\pi$ bisected which was already shown by lemma 5.8.1 for the linear case. Clearly the newest vertex bisection and regular refinement method are well suited for multi-level preconditioning methods which shows that the advantages of a simple sparsity pattern and multi-level preconditioning may be combined for the construction of a preconditioner which is considered in more detail in [26].

5.10 Conclusions

- If the number of refinement levels k is not bounded then the total amount of possibly non-zero entries of a hierarchical matrix is bounded above by $O(kN)$. In practice, one has the bound $\log N \leq k \leq N$. If the number of refinement levels is restricted then the number of possibly non-zero entries will be $O(N)$ as in the standard nodal finite element matrix case.
- The techniques provided to determine upper bounds for the lengths of coupling sets can easily be extended to all grid refinement techniques mentioned in the introduction, for higher order basis functions and for the case of more space dimensions.
- Three-dimensional local bisection refinement is possible and well suited for the generation of finite element grids.
- The simple recursive newest vertex bisection technique is well suited for multi-level preconditioning because the C.-B.-S. scalar

γ^2 which determines the rate of convergence is well bounded below 1 and only depends on the initial coarse grid.

- Newest vertex bisection refinement can be implemented highly efficient and compact, even in a programming language which does not allow for recursion, like fortran-77.

Acknowledgements

The author wishes to thank the Pittsburgh Supercomputing Center for the permission to use the Cray-YMP for the numerical experiments.

5.11 References

[1] Allgower E. and Georg K., *Generation of triangulations by reflections,* Utilitas Mathematica, 16(1979), 123-129

[2] Axelsson O., *An algebraic framework for multilevel methods,* internal report 8820 (October 1988), Department of Mathematics, University of Nijmegen, The Netherlands

[3] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[4] Axelsson O. and Gustafsson I., *Preconditioning and two-level multigrid methods of arbitrary degree of approximation,* Mathematics of Computation, 40(1983), 219-242

[5] Axelsson O. and Maubach J., *A time-space finite element discretization technique for the calculation of the electromagnetic field in ferromagnetic materials,* Journal for Numerical Methods in Engineering, 29(1989), 2085-2111

[6] Axelsson O. and Maubach J., *On the updating and assembly of the Hessian matrix in finite element methods,* Computer Methods in Applied Mechanics and Engineering, 71(1988), 41-67

[7] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods I,* Numerische Mathematik, 56(1989), 157-177

[8] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods II,* SIAM Journal on Numerical Analysis, 27(1990), 1569-1590

[9] Babuška I. and Aziz A.K., *On the angle condition in the finite element method,* SIAM Journal on Numerical Analysis, 13(1976), 214-226

[10] Bank R.E. and Dupont T.F., *Analysis of a two-level scheme for solving finite element equations,* Report CNA-159, Center for Numerical Analysis, University of Texas, Austin, 1980

[11] Bank R.E., Dupont T.F. and Yserentant H., *The hierarchical basis multigrid method,* Preprint SC 87-1 (1987), Konrad-Zuse-Zentrum für Informationstechnik, Berlin

[12] Bank R.E. and Sherman A.H., *The use of adaptive grid refinement for badly behaved elliptic partial differential equations,* **in** Advances in Computer Methods for Partial Differential Equations II (Vichnevetsky R. and Stepleman R.S. eds.), 33-39, IMACS, 1979

[13] Bank R.E. and Smith R.K., *General sparse elimination requires no permanent integer storage,* SIAM Journal on Scientific and Statistical Computing, 8(1987), 574-584

[14] Bank R.E. and Weiser A., *Some a posteriori error estimators for elliptic partial differential equations,* Mathematics of Computation, 44(1985), 283-301

[15] Bänsch E., *Local mesh refinement in 2 and 3 dimensions,* Impact of Computing in Science and Engineering, 3(1991), 181-191

[16] Becker E.B., *Finite Elements,* Prenctice Hall, Englewood Cliffs, New Jersey, 1981

[17] Carey G.F., Barragy R., Mclay R. and Sharma M., *Element by element vector and parallel computations,* Communications in Applied Numerical Methods, 4(1988), 299-307

[18] Carey G.F., Sharma M. and Wang K.C., *A class of data structures for 2-D and 3-D adaptive mesh refinement, in press,* International Journal for Numerical Methods in Engineering

[19] Ciarlet P.G., *The Finite Element Method for Elliptic Problems,* North-Holland Publ., Amsterdam, 1978

[20] Deuflhard P., Leinen P. and Yserentant H., *Concepts of an adaptive hierarchical finite element code,* Preprint SC 88-5, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, September 1988

[21] Greenbaum A., Congming Li and Han Zheng Chao, *Comparison of linear system solvers applied to diffusion-type finite element*

equations, internal report Ultracomputer Note #126, New York University, September 1987

[22] Kardestuncer H. (editor in chief) and Douglas H.N. (project editor), *Finite Element Handbook,* Mc Graw Hill, 1987

[23] Kornhuber R. and Roitzsch R., *On adaptive grid refinement in the presence of internal or boundary layers,* Preprint SC 89-5 of the Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1989

[24] Lee K.D., Huang M., Yu N.J. and. Rubbert P.E., *Grid generation for general three-dimensional configurations,* **in** Numerical Grid Generation, NASA Conference Publication 2166, 355-366 (1982)

[25] Maitre J.F. and Musy F., *The contraction number of a class of two-level methods; an exact evaluation for some finite element subspaces and model problems,* **in** Multigrid Methods (Hackbusch W. and Trottenberg U. eds.), 535-544 [Proceedings Köln-Porz 1981, LNM 960], Springer Verlag, 1982

[26] Maubach J., *Iterative Methods for Non-Linear Partial Differential Equations,* CWI-tract series of the C.W.I. department of the Dutch Organization for Scientific Research N.W.O., C.W.I. Press , Amsterdam, The Netherlands, accepted pending copyright transfers, 1994

[27] Maubach J., *Local bisection refinement for n-simplicial grids generated by reflections,* to appear, SIAM Journal on Scientific and Statistical Computing, 1994

[28] Mitchell W.F., *A comparison of adaptive refinement techniques for elliptic problems,* internal report no. UIUCDCS-R-1375, Department of computer science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1987

[29] Mitchell W.F., *Unified multilevel adaptive finite element methods for elliptic problems,* internal report no. UIUCDCS-R-88-1436, Department of computer science, University of Illinois at Urbana-Champaign, Urbana, Illinois, 1988

[30] Ong M., *Hierarchical basis preconditioners for second order elliptic problems in three dimensions,* Ph.D. thesis, CAM report 89-31, Department of Mathematics, University of California at Los Angeles, Los Angeles, CA. 90024-1555

[31] Rheinboldt W.C., *On a theory of mesh-refinement processes,*

SIAM Journal on Numerical Analysis, 17(1980), 766-778

[32] Rivara M.C., *Algorithms for refining triangular grids suitable for adaptive and multigrid techniques,* International Journal for Numerical Methods in Engineering, 20(1984), 745-756

[33] Rivara M.C., *A grid generator based on 4-triangles conforming mesh refinement algorithms,* International Journal for Numerical Methods in Engineering, 24(1987), 1343-1354

[34] Sewell E.G.,*Automatic generation of triangulations for piecewise polynomial approximation,* Ph.D. thesis, Purdue University, West Lafayette, IN, 1972

[35] Sewell E.G., *A finite element program with automatic user - controlled mesh grading,* **in** Advances in Computer Methods for Partial Differential Equations III (Vichnevetsky R. and Stepleman R.S. eds.), 8-10, IMACS, 1979

[36] Strang G. and Fix G.S., *An analysis of the finite element method,* Prentice Hall, Englewood Heights, New Jersey, 1973

[37] Vassilevski P., *Nearly optimal iterative methods for solving finite element elliptic equations based on the multilevel splitting of the matrix,* internal report of the Institute of Mathematics and Center of Computer Technology, Bulgarian Academy of Sciences, Sofia, Bulgaria, 1989

[38] Yserentant H., *On the multilevel splitting of finite element spaces,* Numerische Mathematik, 49(1986), 379-412

[39] Zienkiewicz O., *The Finite Element Method in Engineering Science,* 3^{rd} edition, Mc Graw-Hill, New York, 1977

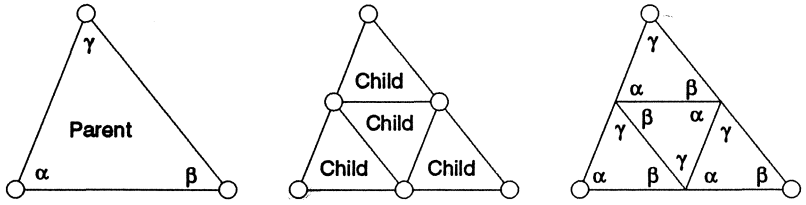


Fig. 5.1 Regular red-1 type of grid refinement and angles created.

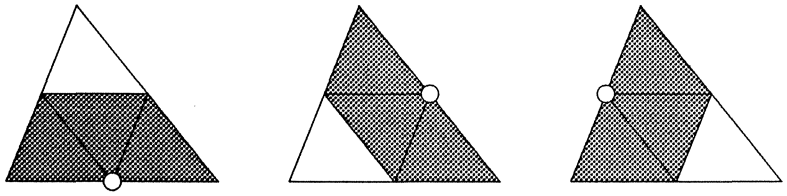


Fig. 5.2 Support of the nodes created by regular refinement.

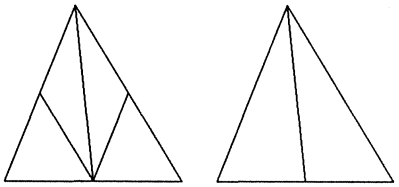


Fig. 5.3 Regular red-2 and green type of grid refinement.

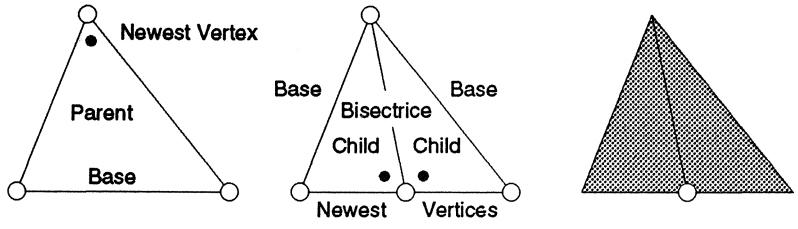


Fig. 5.4 Newest vertex grid refinement and support of created node.

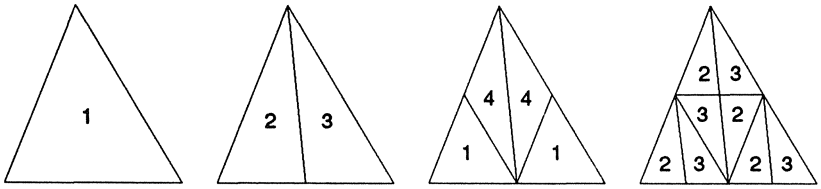


Fig. 5.5 The four congruency classes created by the newest vertex bisection.

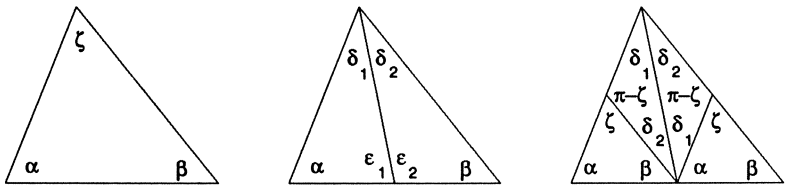


Fig. 5.6 The eight angles involved with bisection.

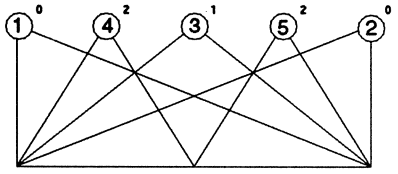


Fig. 5.7 1-Dimensional hierarchical basis functi ons 1, 2, 3, 4, 5 with levels 0, 0, 1, 2, and 2.

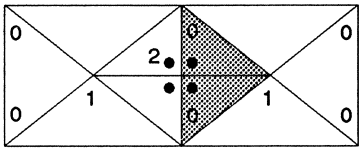
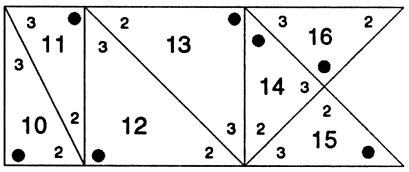
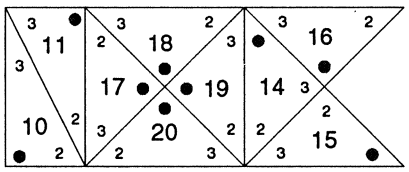


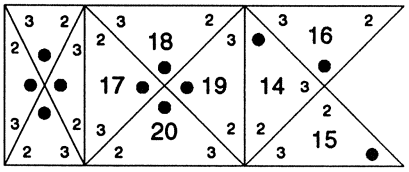
Fig. 5.8 Level 2 newest vertex with a parent with two level 0 vertices and one level 1 vertex.



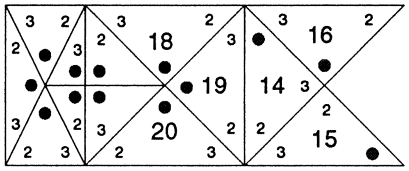
First refine triangles 12 and 13.



Then try to refine triangle 17.



To this end, first refine 10 and 11.



Now, triangle 17 can be refined.

Fig. 5.9 An example application of the newest vertex grid refinement.

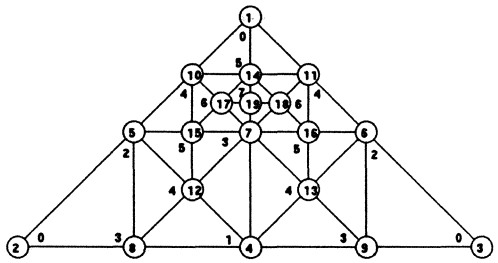


Fig. 5.10 A refined grid with numbers and levels of all vertices.

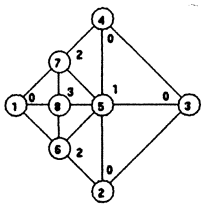


Fig. 5.11 Nodes 7, 5, and 8 have different paths.

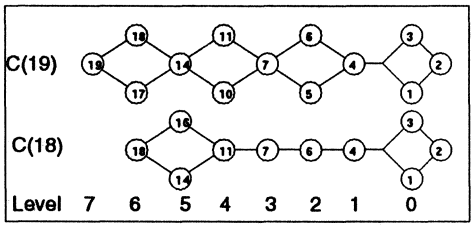


Fig. 5.12 Graphs of coupling sets of the nodes 18 and 19.

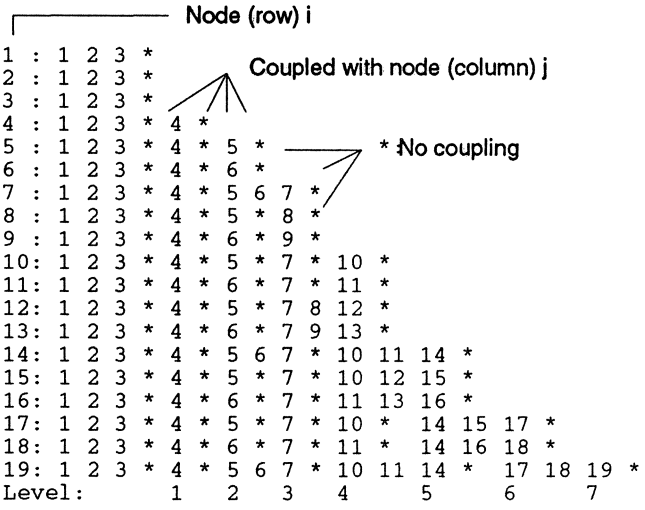


Fig. 5.13 Storage of the sparsity pattern couplings (i, j) in J for all $i, j = 1, \ldots, 19$.

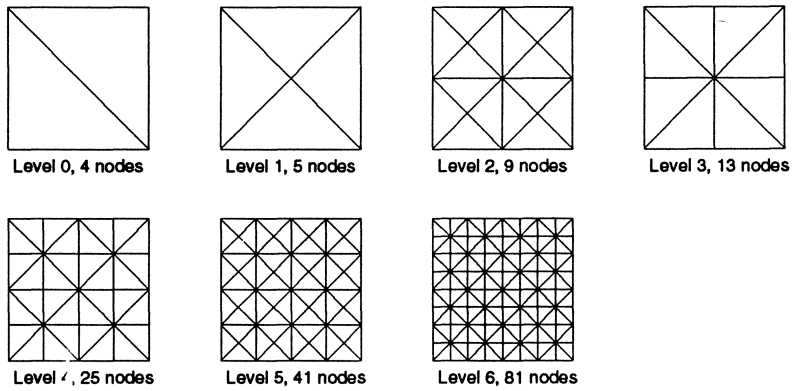


Fig. 5.14 Uniform newest vertex refinement of the unit-square.

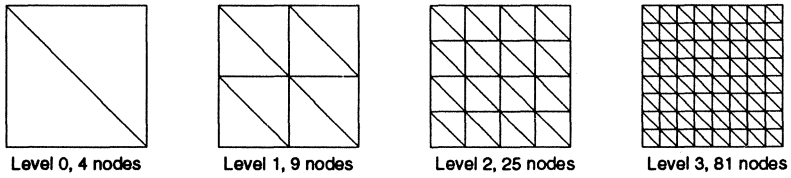


Fig. 5.15 Uniform regular red-1 refinement of the unit-square.

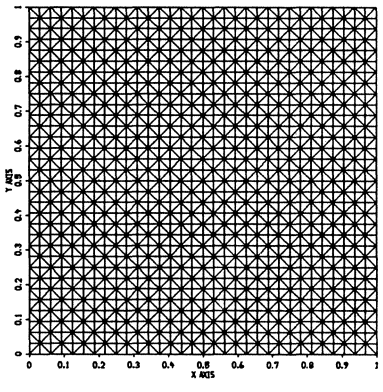


Fig. 5.16 The grid $\mathcal{Q}^{(10)}$.

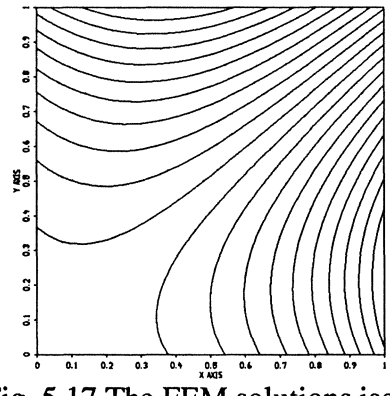


Fig. 5.17 The FEM solutions isoclines of (5.9.2) on $\mathcal{Q}^{(10)}$.

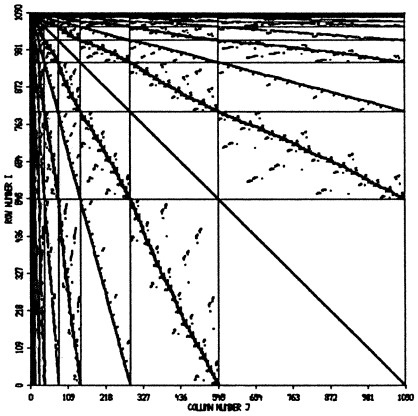


Fig. 5.18 Hierarchical pattern(H).

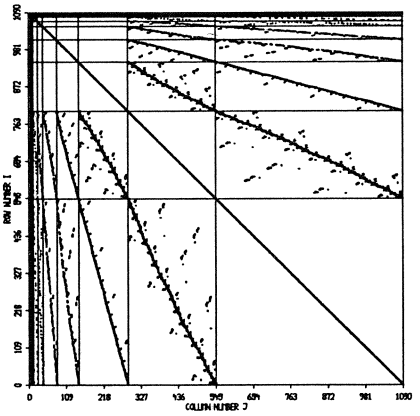


Fig. 5.19 Stand. nodal pattern(N).

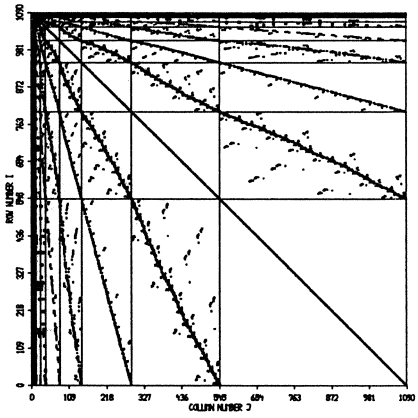


Fig. 5.20 Subset $h_{ij} \neq 0$ (H).

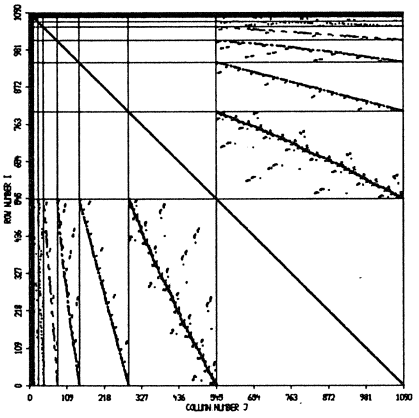


Fig. 5.21 Subset $h_{ij} \neq 0$ (N).

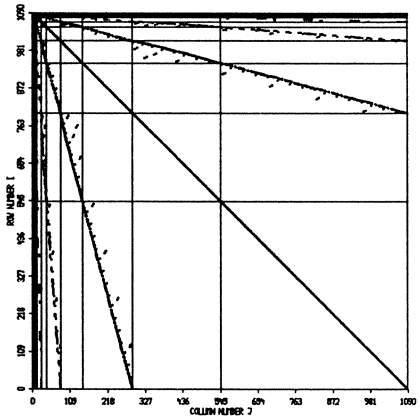


Fig. 5.22 Subset $h_{ij} > 0$ (H).

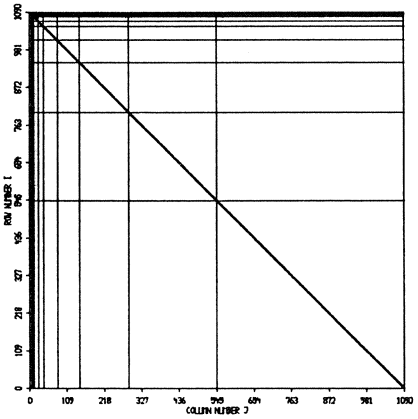


Fig. 5.23 Subset $h_{ij} > 0$ (N).

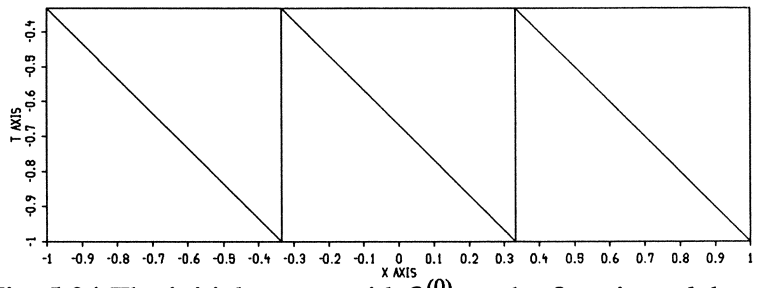


Fig. 5.24 The initial coarse grid $\mathcal{Q}_1^{(0)}$ on the first time-slab.

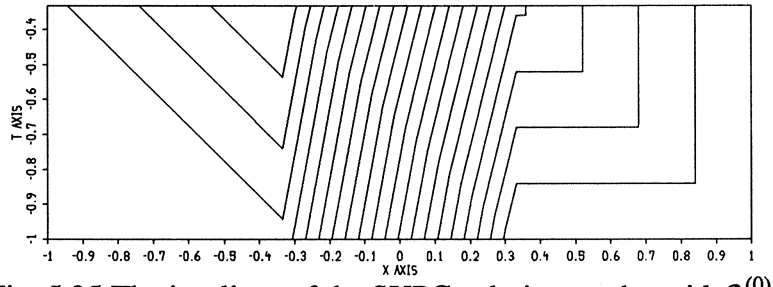


Fig. 5.25 The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(0)}$.

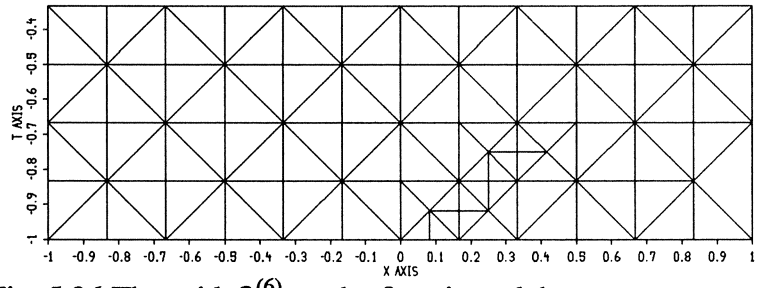


Fig. 5.26 The grid $\mathcal{Q}_1^{(6)}$ on the first time-slab.

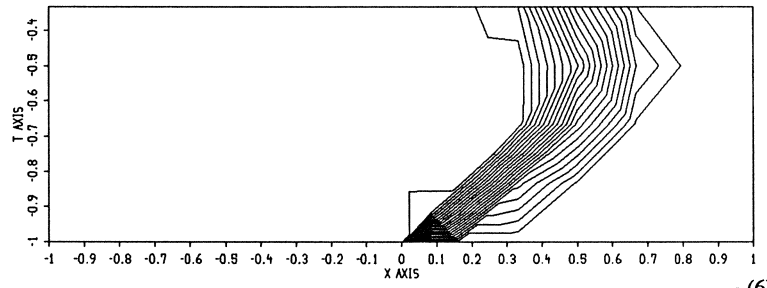


Fig. 5.27 The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(6)}$.

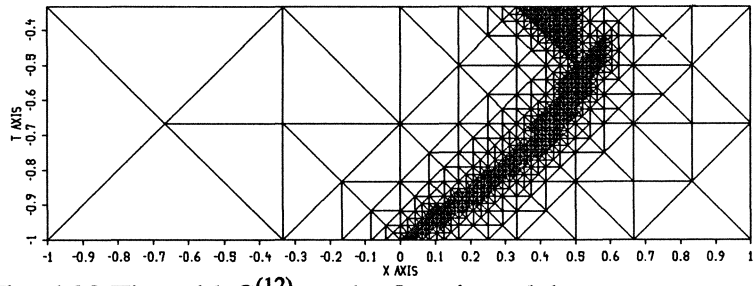


Fig. 5.28 The grid $\mathcal{Q}_1^{(12)}$ on the first time-slab.

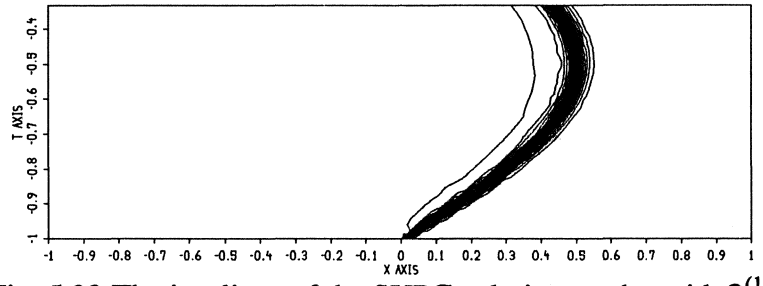


Fig. 5.29 The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(12)}$.

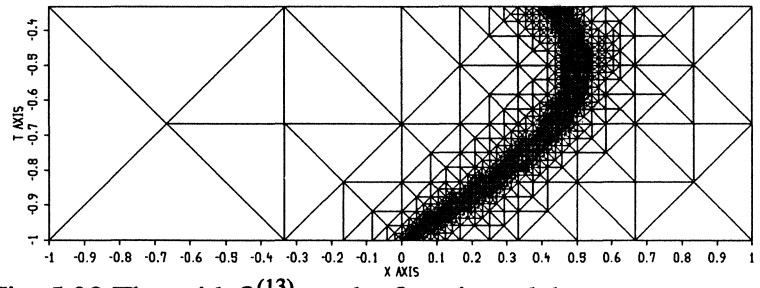


Fig. 5.30 The grid $\mathcal{Q}_1^{(13)}$ on the first time-slab.

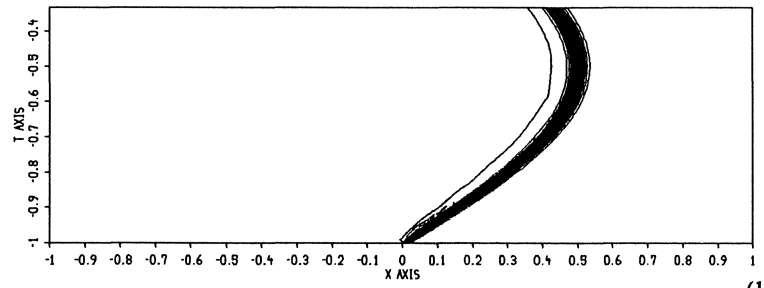


Fig. 5.31 The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(13)}$.

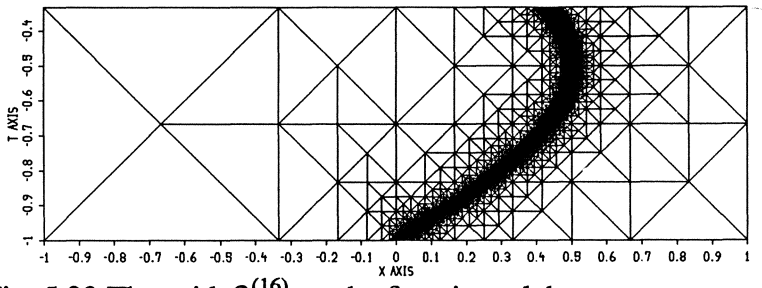


Fig. 5.32 The grid $\mathcal{Q}_1^{(16)}$ on the first time-slab.

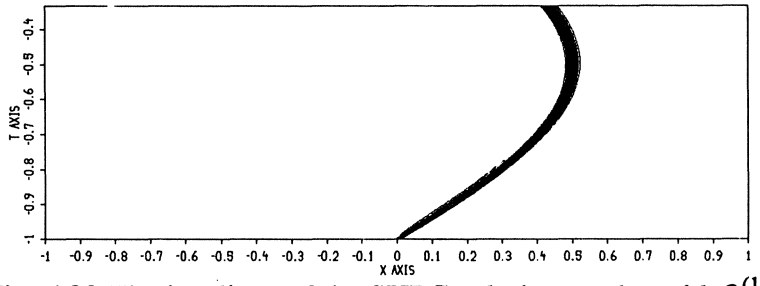


Fig. 5.33 The isoclines of the SUPG solution on the grid $\mathcal{Q}_1^{(16)}$.

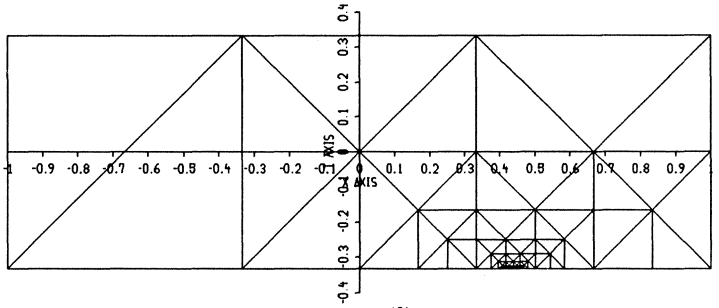


Fig. 5.34 The initial coarse grid $\mathcal{Q}_2^{(0)}$ on the second time-slab.

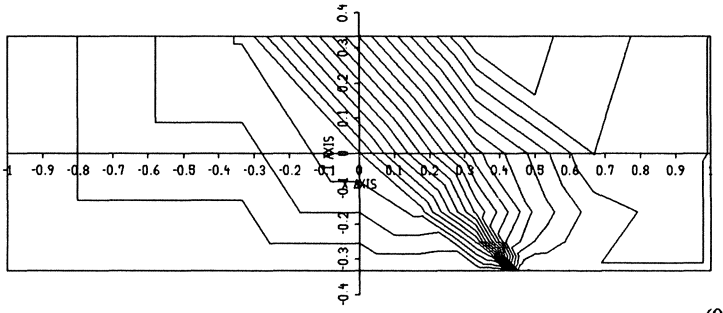


Fig. 5.35 The isoclines of the SUPG solution on the grid $\mathcal{Q}_2^{(0)}$.

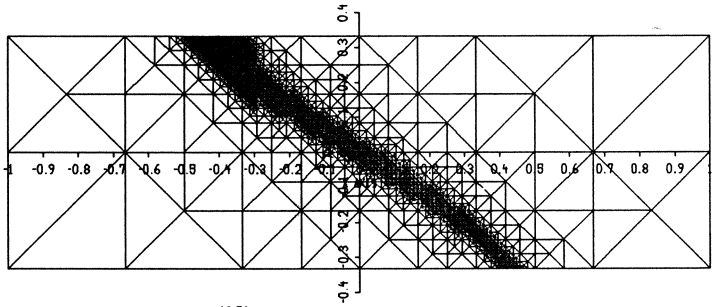


Fig. 5.36 The grid $\mathcal{Q}_2^{(13)}$ on the second time-slab.

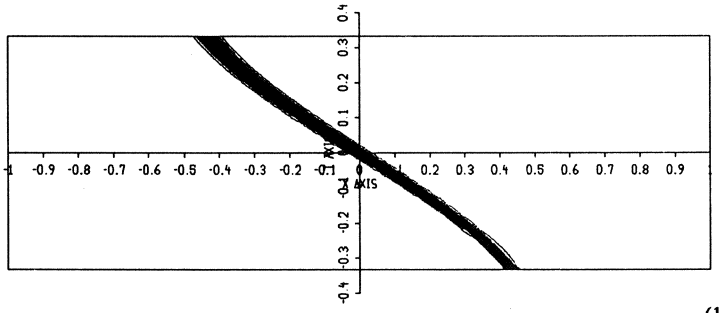


Fig. 5.37 The isoclines of the SUPG solution on the grid $\mathcal{Q}_2^{(13)}$.

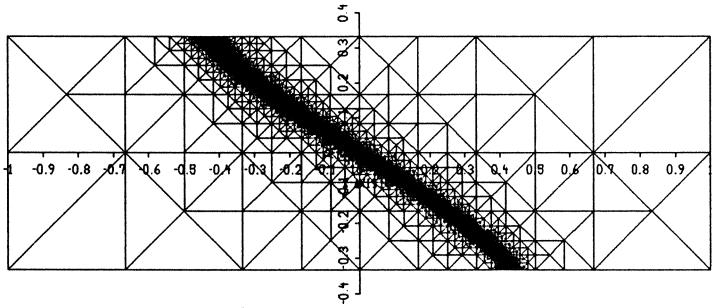


Fig. 5.38 The grid $\mathcal{Q}_2^{(16)}$ on the second time-slab.

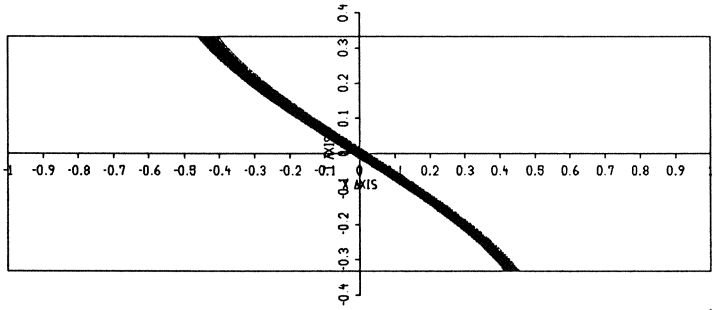


Fig. 5.39 The isoclines of the SUPG solution on the grid $\mathcal{Q}_2^{(16)}$.

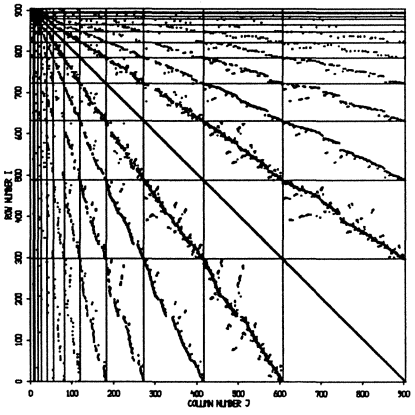


Fig. 5.40 Sparsity pattern $\mathcal{Q}_1^{(12)}$.

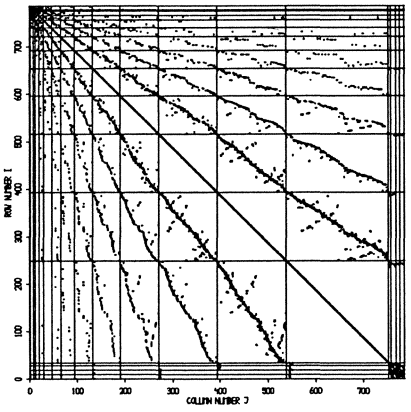


Fig. 5.41 Sparsity pattern $\mathcal{Q}_2^{(12)}$.

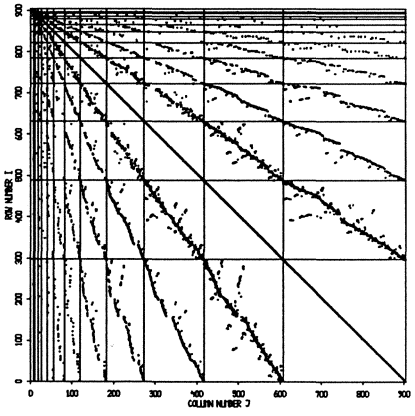


Fig. 5.42 Subset $h_{ij} \neq 0$ for $\mathcal{Q}_1^{(12)}$.

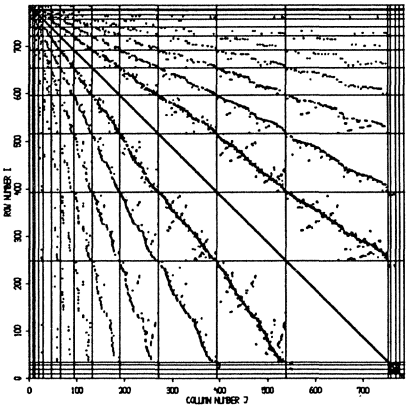


Fig. 5.43 Subset $h_{ij} \neq 0$ for $\mathcal{Q}_2^{(12)}$.

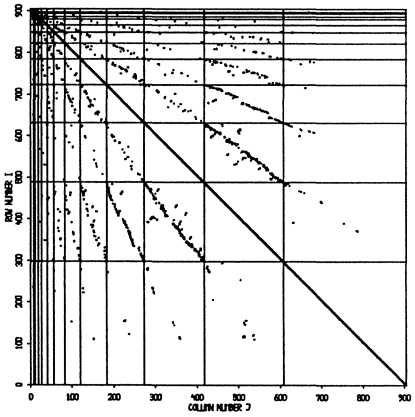


Fig. 5.44 Subs. $h_{ij} > 0$ for $\mathcal{Q}_1^{(12)}$.

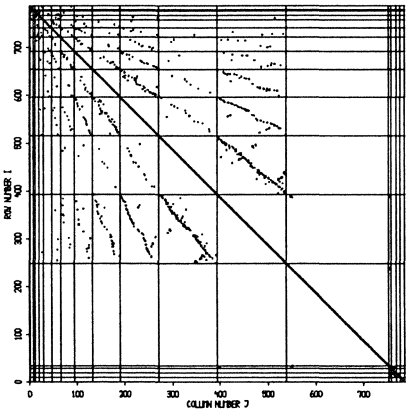


Fig. 5.45 Subs. $h_{ij} > 0$ for $\mathcal{Q}_2^{(12)}$.

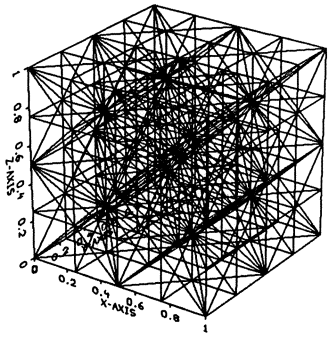


Fig. 5.46 Grid $Q^{(6)}$, uniform refined cube in figure 1.3.

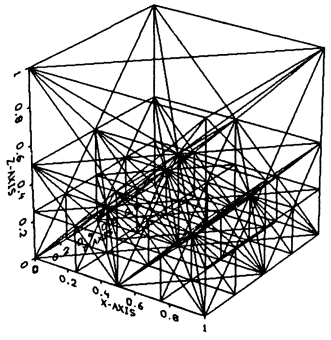


Fig. 5.47 Grid $Q^{(6)}$, refined in the plane $\{(x, y, z): z = 0\}$.

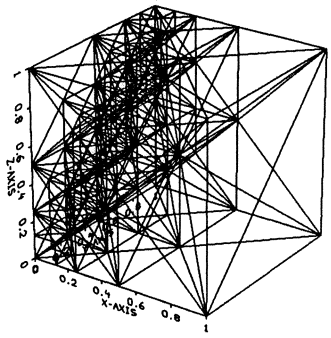


Fig. 5.48 Grid $Q^{(9)}$, refined along the line $\{(x, y, z): x = 0 \wedge y = z\}$.

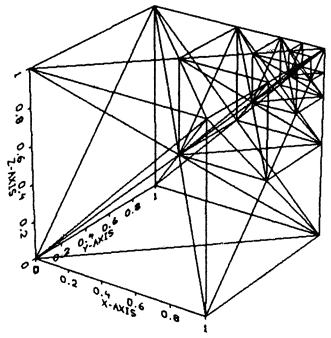


Fig. 5.49 Grid $Q^{(9)}$, refined at the point $(1, 1, 1)$.

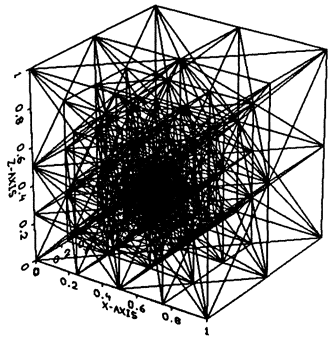


Fig. 5.50 Grid $Q^{(60)}$, refined at the point $(\frac{1}{\sqrt{7}}, \frac{1}{2}\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{5}})$.

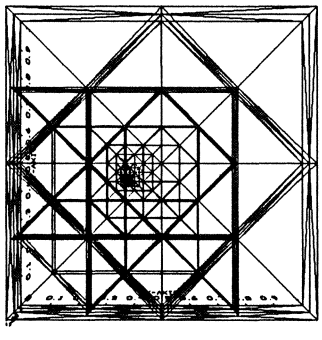


Fig. 5.51 The grid in figure 5.50 seen from a different viewpoint.

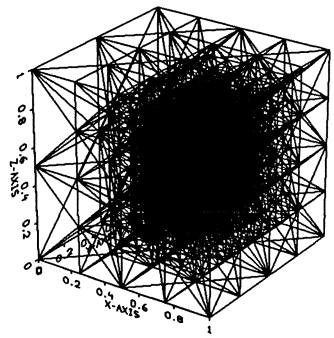


Fig. 5.52 Grid $\mathcal{Q}^{(18)}$ around the hemi-sphere $(x-\frac{1}{2})^2+(y-\frac{1}{2})^2+(z-\frac{1}{2})^2=\frac{1}{16}$ with $x \geq \frac{1}{2}$.

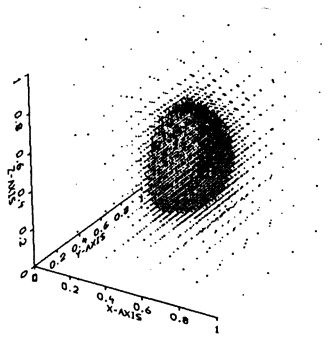


Fig. 5.53 Vertices of this grid.

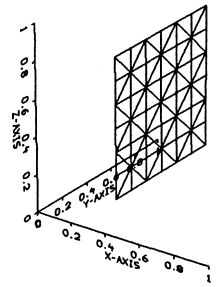


Fig. 5.54 Cross-intersection with plane $x=\frac{3}{8}$.

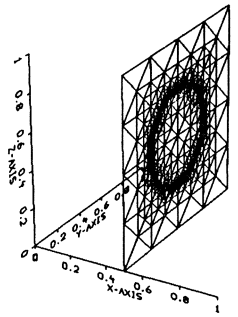


Fig. 5.55 Cross-intersection with plane $x=\frac{1}{2}$.

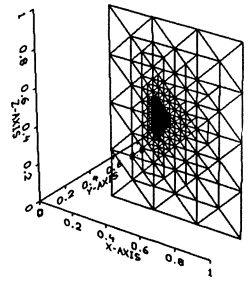


Fig. 5.56 Cross-intersection with plane $y=\frac{1}{4}$.

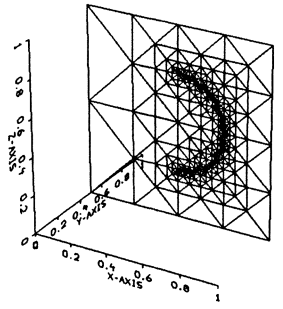


Fig. 5.57 Cross-intersection with plane $y=\frac{1}{2}$.

6 Preconditioners for newest vertex grid refinement

As in: Margenov S.D. and Maubach J.M., Optimal algebraic multi-level preconditioning for newest vertex grid refinement, Technical report no. 10, Bulgarian Academy of Sciences, Center of Informatics and Computer Technology, Sofia 1990.
The reports section concerning the newest vertex grid refinement has been skipped and the reports introduction has been split into two subsequent sections. An extension of the theory towards more general cases of refinement along a line is presented in [8].

Abstract

Recently proposed algebraic multi-level methods for the solution of two-dimensional finite element problems are studied for cases where the local newest vertex grid refinement is applied. After the introduction of this refinement technique it is shown that, by combining certain levels of refinement, a preconditioner of optimal order can be constructed for the case of local refinement along a line.

For all algebraic multi-level preconditioners considered the relative condition number is explicitly calculated. Numerical experiments which demonstrate the performance of these proposed preconditioners will be reported in a forthcoming paper.

Key words: Finite elements, multilevel methods, optimal order preconditioners, newest vertex mesh refinement
AMS(MOS) subject classifications: 65F10, 65N20, 65N30

6.1 Introduction

Recently proposed algebraic multi-level solution algorithms as in [4] and [5] are one of the most effective techniques for the numerical solution of elliptic boundary value problems. So far these algorithms have been developed and tested mostly on uniformly refined grids.

This chapter studies some variants of these methods based on grids which are locally refined with the use of newest vertex grid refinement. It is shown that these variants are an effective generalization of the uniform case.

In order to introduce the algebraic multi-level methods for locally refined grids, a model problem is provided in section 6.2. Thereafter section 6.3 quotes well-known results for the algebraic multi-level preconditioning on uniformly refined grids, and section 6.4 considers the application of this type of preconditioning for the case of local newest grid refinement along a line (see chapter 5). The approximation of matrix blocks, needed to obtain an efficient preconditioner, is commented on in section 6.5. Finally, section 6.6 provides the standard nodal matrices related to the refinement around a corner of the domain and a point in the inner of the domain showing that for those cases a direct solution method is optimal.

6.2 The model problem

Let $\Omega \in \mathbb{R}^2$ be an open bounded connected and polygonal domain with a boundary divided into a Dirichlet boundary part Γ_D with positive Lebesgue measure and a Neumann boundary part Γ_N such that $\Gamma = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$. The goal is to find a function $u \in C^2(\overline{\Omega})$ satisfying

$$\begin{aligned} -\nabla_x \cdot (\epsilon(x)\underline{\nabla}_x u) &= f \text{ in } \Omega \\ u &= \gamma \text{ at } \Gamma_D \\ \underline{\nabla}_x u^T \boldsymbol{n} &= 0 \text{ at } \Gamma_N \end{aligned}$$

where f is a square integrable source function, γ the Dirichlet boundary data and $\underline{\nabla}_x$ denotes the gradient on $\mathbb{R}^2$. The corresponding Galerkin variational formulation is to find a function $u \in H^1_\gamma(\Omega) = \{v \in$

$H^1(\Omega)$: $v \equiv \gamma$ at $\Gamma_D\}$ such that

$$\langle F(u), v\rangle = 0 \quad \forall_{v \in H_0^1(\Omega)} \tag{6.2.1}$$

where

$$\langle F(u), v\rangle = -\int_\Omega \left[\nabla_x \cdot (\epsilon(x)\nabla_x u) + f\right] v \, dx .$$

The domain Ω is assumed to be covered with the triangles in a initial coarse grid $\mathcal{Q}^{(1)}$, e.g. as in fig. 5.14 or 6.1, such that $\epsilon(x)$ is constant on every triangle $\triangle$.

For the solution of (6.2.1) a standard nodal finite element method will be used. If $\mathcal{H}_0^{(1)} = \{v \in \mathcal{H}(\mathcal{Q}^{(1)})$: $v(x) = 0$ at $\mathcal{V}(\mathcal{Q}^{(1)}) \cap \Gamma_D\} \subset H_0^1(\Omega)$ denotes the span of piecewise linear basis functions corresponding to the initial triangulation $\mathcal{Q}^{(1)}$ (see e.g. [2]) then the usual computational procedure leads to the linear system of equations

$$A^{(1)}x^{(1)} = b^{(1)} . \tag{6.2.2}$$

where $A^{(1)}$ is a weighted *stiffness matrix*, x the vector to be determined and b is determined by the source function f.

Now, in order to obtain a sufficiently accurate solution for the problem defined by (6.2.1), the recursive newest vertex grid refinement technique (see section 5.2) is used to construct a sequence of grids $\mathcal{Q}^{(1)} \subset \mathcal{Q}^{(2)} \subset \cdots \subset \mathcal{Q}^{(l)}$, corresponding finite element spaces $\mathcal{H}_0^{(1)} \subset \mathcal{H}_0^{(2)} \subset \cdots \subset \mathcal{H}_0^{(l)}$ and standard nodal stiffness matrices $A^{(1)}, A^{(2)}, \ldots, A^{(l)}$.

In order to calculate the solution of the system of equations related to the finest grid $\mathcal{Q}^{(l)}$

$$Ax = b \tag{6.2.3}$$

where $A = A^{(l)}, x = x^{(l)}$ and $b = b^{(l)}$, a preconditioned conjugate gradient (PCG) iterative solution method (see e.g. [2]) will be used. Solving this algebraic system of equations with a preconditioning matrix C, the convergence properties of the PCG method are given by the estimate

$$r^{(i)T} A^{-1} r^{(i)} \leq \left(2\frac{q^i}{1+q^{2i}}\right)^2 \cdot r^{(0)T} A^{-1} r^{(0)} \tag{6.2.4}$$

where

$$q = \frac{\sqrt{\chi} - 1}{\sqrt{\chi} + 1}, \quad \chi = \mathrm{Cond}(C^{-1}A)$$

and the residual is given by $r^{(i)} = b - Ax^{(i)}$. The estimate (6.2.4) shows that the number of iterations needed to reduce the norm of the residual by a factor ε is $O(\sqrt{\chi})$.

The main goal of this chapter is to construct a matrix C such that solving the system of linear equations for C requires an amount of arithmetic operations linear proportional to the number of unknowns. It also is required that the condition number χ above is bounded uniformly with respect to the number of the degrees of freedom.

6.3 Algebraic multi-level preconditioning

In this section the basic requirements needed for the possible application of algebraic multi-level preconditioning are summarized and checked for the simple case of uniform newest vertex grid refinement of the unit-square. Consider the system

$$A^{(k+1)}x^{(k+1)} = b^{(k+1)} \tag{6.3.1}$$

where $1 \leq k \leq l$, for l defined as in section 6.1. In order to define a preconditioning matrix $C^{(k+1)}$ the nodes $\mathcal{F}(\mathcal{Q}^{(k+1)})$ of grid $\mathcal{Q}^{(k+1)}$ are partitioned into two mutually disjunct subsets $\mathcal{F}(\mathcal{Q}^{(k+1)}) - \mathcal{F}(\mathcal{Q}^{(k)})$ and $\mathcal{F}(\mathcal{Q}^{(k)})$. Corresponding to this partitioning $A^{(k+1)}$ takes the following two by two block structure

$$A^{(k+1)} = \begin{bmatrix} A_{11}^{(k+1)} & A_{12}^{(k+1)} \\ A_{21}^{(k+1)} & A_{22}^{(k+1)} \end{bmatrix}, \tag{6.3.2}$$

where the first pivot block $A_{11}^{(k+1)}$ corresponds to the nodes of $\mathcal{F}(\mathcal{Q}^{(k+1)}) - \mathcal{F}(\mathcal{Q}^{(k)})$ and the second diagonal block $A_{22}^{(k+1)}$ corresponds to nodes of $\mathcal{F}(\mathcal{Q}^{(k)})$, following a notation by [4] and [5]. Note that this notation is slightly different from the notation used in chapter 5. Following the construction proposed in [4] a two-level preconditioner $C^{(k+1)}$ will be defined by

$$C^{(k+1)} = \begin{bmatrix} A_{11}^{(k+1)} & O \\ A_{21}^{(k+1)} & A^{(k)} \end{bmatrix} \circ \begin{bmatrix} I & A_{11}^{(k+1)^{-1}} A_{12}^{(k+1)} \\ O & I \end{bmatrix} \tag{6.3.3}$$

where $A^{(k)} = A_{22}^{(k+1)}$. Using this definition one can prove

Lemma 6.3.1 *The matrices $A^{(k+1)}$ and $C^{(k+1)}$ are spectrally equivalent, i.e., there exists a positive scalar μ such that*

$$\mu x^T C^{(k+1)} x \leq x^T A^{(k+1)} x \leq x^T C^{(k+1)} x \tag{6.3.4}$$

for all vectors x. Further, μ is independent of the number of the degrees of freedom $N^{(k+1)}$, the size of the matrix $A^{(k+1)}$, and $\mu = 1 - \gamma^2$, where γ is the constant in the strengthened C.-B.-S inequality.

Proof. Note that

$$C^{(k+1)} - A^{(k+1)} = \begin{bmatrix} O & 0 \\ O & A^{(k)} - S^{(k+1)} \end{bmatrix}$$

where the *Schur complement* is given by

$$S^{(k+1)} = A_{22}^{(k+1)} - A_{21}^{(k+1)} A_{11}^{(k+1)^{-1}} A_{12}^{(k+1)} .$$

Using e.g. [4], lemma 2.1, the lemma follows. This result was first obtained by Axelsson [1], Kuznetsov [6] and Axelsson and Gustafsson [3]. Note that the inequalities are sharp and that the scalar γ can be computed locally from the element stiffness matrices. □

After this example of a two-level preconditioner now consider the general multi-level case. The linear system of equations to be solved is again (6.2.3), but now the preconditioner $C = C^{(k+1)}$ is defined recursively by

$$\begin{aligned} C^{(1)} &= A^{(1)} \\ C^{(k+1)} &= \begin{bmatrix} A_{11}^{(k+1)} & O \\ A_{21}^{(k+1)} & \hat{A}^{(k)} \end{bmatrix} \circ \begin{bmatrix} I & A_{11}^{(k+1)^{-1}} A_{12}^{(k+1)} \\ O & I \end{bmatrix} \end{aligned} \tag{6.3.5}$$

for all $k = 1, 2, \ldots, l-1$, where

$$\hat{A}^{(k)^{-1}} = [I - p_\beta(C^{(k)^{-1}} A^{(k)})] A^{(k)^{-1}} \tag{6.3.6}$$

and $p_\beta(x)$ is a properly normalized and shifted *Chebyshev polynomial* of degree β. In this case one can prove

Lemma 6.3.2 *Let the preconditioning matrix C be defined by (6.3.5), (6.3.6) and let A be defined as in (6.3.2). Then*

- *if the polynomial degree $\beta > (1-\gamma^2)^{-1}$ then the relative spectral condition number $\chi(C^{-1}A)$ is bounded by a constant independent of the number of degrees of freedom $N \equiv N^{(l)}$ of the finest grid, i.e., C and A are spectrally equivalent*
- *the total computational costs Z per PCG iteration step involving preconditioner C are bounded by*

$$Z \leq cN \cdot (1 + r + r^2 + \cdots + r^{l-1}) = cN \frac{1 - r^l}{1 - r}$$

where $r = \beta \max_k \{N^{(k)}/N^{(k+1)}\}$ for some positive scalar c.

Proof. One can prove the first statement with the use of uniform estimates for $\chi(C^{(k+1)^{-1}} A^{(k+1)})$, as are obtained by Axelsson and Vassilevski [4] and [5]. The estimate of the total computational costs in the second statement follows directly from the factorized structure of the recursively defined algebraic multi-level preconditioner $C^{(k+1)}$ (for some more details, see e.g. [7]). Note that obviously $Z = O(N)$ for $r < 1$, which is necessary to obtain an optimal preconditioner. □

In order to illustrate the behaviour of the described multi-level preconditioner C above consider the following example where uniform newest vertex grid refinement is used (see also [10] where a generalization of the bisection refinement to higher dimensions can be found).

First let Q be the unit-square and let $\mathcal{Q}^{(1)} \subset \mathcal{Q}^{(2)} \subset \cdots \subset \mathcal{Q}^{(k)}$ be obtained by uniform newest vertex grid refinement, as is shown in figs. 5.9. Here $(1-\gamma^2)^{-1} = 2$ (see e.g. [9]) and according to lemma 6.3.1 $\beta \geq 2$ leads to $r > 1$. Therefore, straightforwardly taking the levels of refinement induced by the newest vertex refinement method as the levels defining the multi-level preconditioner C in (6.3.6) will not lead to an optimal multi-level preconditioner with respect to the number of degrees of freedom.

In order to overcome this difficulty consider a reformulation of the above where every two uniform refinement steps are joined into one level as far as the preconditioning is concerned. This means that each $\mathcal{Q}^{(k)}$ is obtained after $2k$ steps of refinement. Again local analysis shows that $(1 - \gamma^2)^{-1} = 2$ whence an optimal order preconditioner C is

yielded taking $\beta \in \{2,3\}$ alternatingly. One obtains respectively r =0.5 for $\beta = 2$ and r =0.75 for $\beta = 3$, and in both cases the computational costs are $Z = O(N)$. Note that in this case of uniform refinement the stiffness matrices $A^{(k)}$ on the grids in fig. 5.14 coincide with those corresponding to the uniform refinement, as shown in fig. 5.15 (see also [2], [4]).

6.4 Local refinement along a line

Consider the unit-square Ω with initial coarse grid $\mathcal{Q}^{(1)}$ as shown in fig. 6.1 (a). This domain is refined subsequently along the lower boundary such that $\mathcal{Q}^{(2)}$ as shown in fig. 6.1 (b) is obtained after four steps of local newest vertex grid refinement. Applying the same technique to $\mathcal{Q}^{(2)}$ leads to grid $\mathcal{Q}^{(3)}$ as shown in fig. 6.1 (c). Hence, in order to construct an optimal order preconditioner every four refinement levels are combined into one level regarding the multi-level preconditioning algorithm (6.3.6), i.e., $\mathcal{Q}^{(k)}$ is obtained after $4k$ refinement steps.

To simplify the presentation, assume that the boundary points, including those lying on a Dirichlet boundary, are taken into account for the number of unknowns $N^{(k)}$ related to $\mathcal{Q}^{(k)}$. In order to determine this number let, only in this section, $n^{(k)}$ denote the number of unknowns at the $2 \cdot k^{th}$ level of ordinary newest vertex refinement. Then $N^{(k)} = n^{(2k-1)}$ and, as is easy to verify,

$$n^{(k+1)} - 2n^{(k)} = 1 - k\,. \tag{6.4.1}$$

The solution of (6.4.1) can be written as the sum

$$n^{(k)} = c_1 z^k + d_0 + d_1 k$$

where $c_1 z^k$ is the general solution of the homogeneous difference equation $n^{(k+1)} - 2n^{(k)} = 0$, which has root $z = 2$ corresponding to the related characteristic equation. The term $d_0 + d_1 k$ is a partial solution of the inhomogenous equation (6.4.1). The substitution of $z = 2$ and the determination of the coefficients lead to

$$\begin{aligned} n^{(k)} &= 2^k + k \\ N^{(k)} &= 2^{2k} + 2k - 1\,. \end{aligned} \tag{6.4.2}$$

The obtained result hence can be summarized as follows

Theorem 6.4.1 *For the newest vertex refinement along a line as described in the beginning of this section*

$$\frac{N^{(k+1)}}{N^{(k)}} > 3.6 \tag{6.4.3}$$

for all $k \in \mathbb{N}$.

Proof. Following (6.4.2) one has

$$\frac{N^{(k+1)}}{N^{(k)}} = 4 - \frac{6k+2}{2^{2k+1}+2k} > 4 - 0.4 = 3.6$$

yielding the desired result. □

Now consider the estimation of the scalar γ in the strengthened C.-B.-S. inequality, or more precisely, the estimation of the relative condition number $\chi = (1-\gamma^2)^{-1}$. Following the general procedure to this end, let the domain Ω be partitioned into so-called *macro elements* $\mathcal{A}_i$, $i = 1, 2$, as are shown in fig. 6.2. The macro elements correspond naturally to the refinement procedure shown in fig. 6.1.

Using the usual partitioning of the nodes belonging to $\mathcal{F}(\mathcal{Q}^{(k+1)})-\mathcal{F}(\mathcal{Q}^{(k)})$ and to $\mathcal{F}(\mathcal{Q}^{(k)})$ one can write the macro element stiffness matrix related to the cases in fig. 6.2 in the form

$$A_{\mathcal{A}} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \tag{6.4.4}$$

such that A_{11} corresponds to the nodes in $\mathcal{F}(\mathcal{Q}^{(k+1)}) - \mathcal{F}(\mathcal{Q}^{(k)})$. As is well known, the relative spectral condition number $\chi = (1-\gamma^2)^{-1}$ is in this case equal to the largest eigenvalue λ of the generalized eigenvalue problem

$$A_{\mathcal{A}}\boldsymbol{x} = \lambda S_{\mathcal{A}}\boldsymbol{x}\,, \tag{6.4.5}$$

where $\mathcal{A}$ is equal to one of the two macro elements $\mathcal{A}_i$. The Schur complement $S_{\mathcal{A}}$ is defined as usual by $S_{\mathcal{A}} = A_{22} - A_{21}A_{11}^{-1}A_{12}$ using the blocks of the macro element matrix $A_{\mathcal{A}}$. Approximation of the largest generalized eigenvalue leads to the estimate

$$\frac{1}{1-\gamma^2} < 2.85 \tag{6.4.6}$$

whence one can conclude

Theorem 6.4.2 *The multi-level preconditioning method based on local newest vertex grid refinement along a line, as presented in this section, is of optimal arithmetic costs $O(N)$ for $\beta \in \{2,3\}$.*
Proof. Combine the results above with theorem 6.4.1. □

Note that the realization of the PCG method with the above defined multi-level preconditioning matrix C requires a repeatedly solution of systems with matrices $A_{11}^{(k+1)}$ for $k = 1,\ldots,l-1$. It is well known that the condition number of these matrices is bounded uniformly above whence a conjugate gradient method can be used effenciently to this end.

6.5 Algebraic multi-level preconditioning with approximate blocks

For the realization of a preconditioned conjugate gradient method with the preconditioner C in (6.3.5) one needs to solve linear systems involving the matrices $A_{11}^{(k)}$, $k = 1,2,\ldots,l$. These positive definite matrices have a condition number independent of N, but unfortunately this condition number $\chi(A_{11}^{(k)})$ increases strongly due to the combination of the newest vertex refinement levels. For the uniform refinement case (where no combination of refinement levels is needed) the blocks $A_{11}^{(k)}$ are diagonal leading to an algorithm of inverse free type (for more details about such methods see [4], [5] and [12]) but in this case one finds

$$\chi(A_{11}^{(k)}) \leq 36 \tag{6.5.1}$$

by solving the eigenvalue problems corresponding to the 11-subblocks of the matrices $A_{\mathcal{A}}$ as in section 6.4. This estimate shows that the so called direct approach with regard to the multi-level algorithm, where the conjugate gradient method is used for the solution of the systems with the matrices $A_{11}^{(k)}$, is not very efficient (the condition numbers are too large).

In order to overcome this difficulty, one could use a preconditioned conjugate gradient algorithm where the blocks $A_{11}^{(k)}$ are approximated by positive definite matrices $B_{11}^{(k)}$, for all $k = 2,3,\ldots,l$.

Lemma 6.5.1 *If* $\beta^2 > (1-\gamma^2)$ *and* $p > \chi(B_{11}^{(k)^{-1}} A_{11}^{(k)}) - 1 > 0$ *then the relative condition number of the resulting multi-level preconditioning matrix* $C^{(k)}$ *with respect to* A *is bounded uniformly by* α^{-1} *where for* $\mu = 1-\gamma^2$

- *for* $\beta = 2$,

$$\alpha = \frac{4\mu - 1}{1 + 2p + \sqrt{4\mu - 1 + (1+2p)^2}} \tag{6.5.2}$$

- *for* $\beta = 3$, $\alpha \in (0,1)$ *is the smallest positive root* x *of the cubic equation*

$$px^3 + (6p + 9 - \mu)x^2 + (9p + 6 - 6\mu)x + 1 - 9\mu = 0\,. \tag{6.5.3}$$

Note that such an α *exists for* $\mu > \frac{1}{9}$.

Proof. See the results in [5]. □

Consider the following construction of approximations $B_{11}^{(k)}$ for the blocks $A_{11}^{(k)}$ (for some related results, see [7]). The superscripts $^{(k)}$ are omitted whenever possible for the sake of convenience. For the construction of the approximation, let the macro element $\mathcal{B}$ be as in fig. 6.3 and let its associated nodes be partitioned into two sets, the first one containing the inner nodes and the second one the remaining boundary nodes. Then the block $A_{11} \equiv A_{11}^{\mathcal{B}}$ is factorized as follows

$$A_{11} = \begin{bmatrix} D & F \\ F^T & E \end{bmatrix} = \begin{bmatrix} D & 0 \\ F^T & S \end{bmatrix} \circ \begin{bmatrix} I & D^{-1}F \\ 0 & I \end{bmatrix}, \tag{6.5.4}$$

where the Schur complement $S = S^{\mathcal{B}}$ is defined by $S = E - F^T D^{-1} F$, and the D block corresponds to the inner nodes of the macro element $\mathcal{B}$. From the definition of the global matrix A, it follows that the global Schur complement S can be written as the sum

$$S = \sum_{\mathcal{B} \in \mathcal{Q}^{(k)}} S^{\mathcal{B}}\,. \tag{6.5.5}$$

Now the promised approximation $B_{11}^{(k)}$ is defined by the formula

$$B_{11}^{(k)} = \begin{bmatrix} D & 0 \\ F^T & \overline{S} \end{bmatrix} \circ \begin{bmatrix} I & D^{-1}F \\ 0 & I \end{bmatrix} \tag{6.5.6}$$

where the $\overline{S}$ is an approximation of S defined by

$$\overline{S} = \sum_{C \subset \mathcal{B} \subset \mathcal{Q}^{(k)}} S_C \,. \tag{6.5.7}$$

Here S_C stands for the macro element Schur complements corresponding to the partition (6.5.4) and the macro elements $C = C_i \subset \mathcal{B}$ as in fig. 6.4.

It is obvious, that S and $\overline{S}$, respectively A_{11} and B_{11}, are spectrally equivalent with a relative spectral condition number $\chi(B_{11}^{-1}A_{11})$ independent of N. Taking into account (6.5.5) and (6.5.6), this result is stated in

Lemma 6.5.2 *The spectral condition number $\chi(B_{11}^{-1}A_{11})$ is bounded above by the scalar* 1.93 *uniformly in N.*

Proof. Using a local analysis on macro element level, one obtains the estimations

$$\chi(B_{11}^{-1}A_{11}) \leq \chi(B_{11}^{\mathcal{B}\,-1}A_{11}^{\mathcal{B}}) \,.$$

In combination with the solution of the generalized eigenvalue problem

$$S^{\mathcal{B}}\boldsymbol{x} = \lambda \overline{S}^{\mathcal{B}}\boldsymbol{x} \,, \tag{6.5.8}$$

yielding $\lambda_{\max}/\lambda_{\min} = 1.9245$ by approximation, whence

$$\chi(B_{11}^{-1}A_{11}) \leq 1.93 \,.$$

□

Theorem 6.5.1 *The spectral condition number $\chi(C^{-1}A)$ is uniformly bounded by*

$$\chi(C^{-1}A) \leq 24 \tag{6.5.9}$$

for $\beta = 2$. Therefore the total arithmetic costs of the proposed algebraic multi-level algorithm with approximate blocks $B_{11}^{(k)}$ are linear proportional to N.

Proof. The first estimate follows from theorem 6.4.2, lemma 6.5.1 and the relations (6.5.2) and (6.5.3). As $\chi(C^{-1}A) = O(1)$, the total number of iterations for the preconditioned conjugate gradient solution of the system (6.2.3) is of $O(1)$. Now consider the structure of the block matrix B_{11}. The blocks D and $\overline{S}$ are block diagonal matrices with 2 by 2 blocks. Therefore the solution of the systems A_{11} with preconditioner B_{11} in a preconditioned conjugate gradient method involves only $O(N)$ arithmetic operations. Hence the total arithmetic costs per iteration is $O(N)$ and (6.5.9) holds. □

The special block structure of the matrices $\overline{S}$ makes the considered algorithm very suitable for a parallel realization.

6.6 Standard nodal matrices for point sources

The newest vertex refinement into a corner of the computational domain will lead to the following sparse standard nodal finite element matrix (see fig. 6.6 for the numbering of the degrees of freedom in this case).

$$2A = \begin{bmatrix} A_{11} & A_{12} & 0 & \cdots & 0 & F^T \\ A_{21} & B & C^T & & & 0 \\ 0 & C & B & C^T & & \vdots \\ \vdots & & C & \ddots & \ddots & 0 \\ 0 & & & \ddots & B & D^T \\ F & 0 & \cdots & 0 & D & E \end{bmatrix}.$$

The bordering submatrices are given by

$$A_{11} = \begin{bmatrix} 2 & 0 & 0 & \star \\ 0 & 2 & \star & \star \\ 0 & \star & 2 & \star \\ \star & \star & \star & 2 \end{bmatrix} \quad A_{12} = \begin{bmatrix} -2 & \star & \star \\ -1 & \star & -1 \\ -1 & -1 & \star \\ \star & \star & \star \end{bmatrix} \quad A_{21} = A_{12}^T$$

and

$$F = [\star \ \star \ \star \ -2]$$
$$D = [-2 \ -2 \ -2]$$
$$E = [8].$$

The block tridiagonal parts are given by

$$B = \begin{bmatrix} 8 & -1 & -1 \\ -1 & 4 & 0 \\ -1 & 0 & 4 \end{bmatrix} \quad C = \begin{bmatrix} -2 & -1 & -1 \\ \star & -1 & 0 \\ \star & \star & -1 \end{bmatrix}$$

Here an asterix at position (i, j) means that $[A]_{ij} = 0$ since the corresponding nodes are not coupled (see e.g. [11]). Therefore, the (possibly) non-zero entries above correspond to coupled nodes. This implies that every sparse matrix resulting from the standard nodal finite element discretization of an arbitrary partial differential equation on this grid will have precisely the above sparsity pattern. Clearly a direct solution method, like Gaussian elimination, will yield an optimal result concerning the number of arithmetic operations.

Note that figure 6.6 shows that the node numbering reflects the level structure as was demanded in section 5.2. The bordering matrices A_{11}, A_{12}, A_{21} and F will vanish if the computational domain is a subset of a larger computational domain and if domain decomposition techniques are used to isolate the subdomain. This will also happen if one imposes Dirichlet boundary conditions on all boundary points 1, 2, 3 and 4.

The finite element matrix A resulting from symmetric local newest vertex grid refinement around a point is the following (see fig. 6.5 for the node numbering applied in this case)

$$2A = \begin{bmatrix} A_{11} & A_{12} & 0 & \cdots & 0 & E^T \\ A_{21} & B & C^T & & & 0 \\ 0 & C & B & C^T & & \vdots \\ \vdots & & C & \ddots & \ddots & 0 \\ 0 & & & \ddots & B & C^T \\ E & 0 & \cdots & 0 & C & D \end{bmatrix}$$

where

$$A_{11} = \begin{bmatrix} 2I & J & 0G^T \\ J^T & 8 & J^T \\ 0G^T & J & 4I \end{bmatrix} \quad A_{12} = \begin{bmatrix} -2I & O \\ J^T & J^T \\ -G & -2I \end{bmatrix} \quad A_{21} = A_{12}^T$$

and

$$B = \begin{bmatrix} 8I & -G^T \\ -G & 8I \end{bmatrix} \quad C = \begin{bmatrix} -2I & -G^T \\ O & -2I \end{bmatrix}$$

$$D = \begin{bmatrix} 8I & -2G^T \\ -2G & 8I \end{bmatrix} \quad E = \begin{bmatrix} O & J^T & O \\ O & -2K^T & O \end{bmatrix}$$

with blocks

$$G = \begin{bmatrix} 1 & \star & 1 & \star \\ 1 & 1 & \star & \star \\ \star & \star & 1 & 1 \\ \star & 1 & \star & 1 \end{bmatrix} \quad I = \begin{bmatrix} 1 & \star & \star & \star \\ \star & 1 & \star & \star \\ \star & \star & 1 & \star \\ \star & \star & \star & 1 \end{bmatrix} \quad O = \begin{bmatrix} \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \\ \star & \star & \star & \star \end{bmatrix}$$

and

$$K^T = [1 \quad 1 \quad 1 \quad 1] \quad J^T = [\star \quad \star \quad \star \quad \star] .$$

In this case the A_{11} matrix block contains the couplings between the nodes of level 0, 1 and 2, and the following B blocks contain the couplings of each next two subsequent levels of refinement. Also in this case of a matrix of band-width 19 a direct solution method will be of optimal computational complexity.

6.7 References

[1] Axelsson O., *On multigrid methods of the two-level type,* **in** Multigrid Methods (Hackbusch W. and Trottenberg U. eds.), LNM 960, Springer Verlag, 1982, 352-367 [Proceedings, Köln-Porz 1981]

[2] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[3] Axelsson O. and Gustafsson I., *Preconditioning and two-level multigrid methods of arbitrary degree of approximation,* Mathematics of Computation, 40(1983), 219-242

[4] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods I,* Numerische Mathematik, 56(1989), 157-177

[5] Axelsson O. and Vassilevski P.S., *Algebraic multilevel preconditioning methods II,* SIAM Journal on Numerical Analysis, 27(1990), 1569-1590

[6] Kuznetsov Yu.A., *Multigrid Domain Decomposition Methods for Elliptic Problems,* Preprint of lectures of the VIII International Conference in Computational Methods for Applied Sciences and Engineering, 2(1987), 605-616

[7] Margenov S.D., *Inverse-free multilevel methods I,* Technical report no. 4, Bulgarian Academy of Sciences, Center of Informatics and Computer Technology, Sofia, 1989

[8] Margenov S.D. and Maubach J.M., *Optimal algebraic multilevel preconditioning for local refinement along a line,* accepted by the Journal of Numerical Linear Algebra with Applications, The Netherlands, 1994

[9] Margenov S.D., Vassilevski P.S. and Neytcheva M.G., *Optimal order algebraic multilevel preconditioners for finite element 2-D elasticity equations,* Report 9021, Mathematics department, University of Nijmegen, The Netherlands, 1990

[10] Maubach J., *Local bisection refinement for n-simplicial grids generated by reflections,* to appear, SIAM Journal on Scientific and Statistical Computing, 1994

[11] Maubach J.M., *On the sparsity pattern of hierarchical finite element matrices,* **in** Lecture Notes in Mathematics 1457, 79-104, (Axelsson O. and Kolotilina L. Yu. eds.), Springer Verlag, 1990, [Proceedings of the International Conference on Preconditioned Conjugate Gradient Methods and Applications, Nijmegen, The Netherlands, 1989]

[12] Vassilevski P., *Algebraic multilevel preconditioners of elliptic problems with condensation of the finite element stiffness matrix,* Compt. rend. de l'Acad. Bulg. Sci. 43(1990), no. 6, to appear

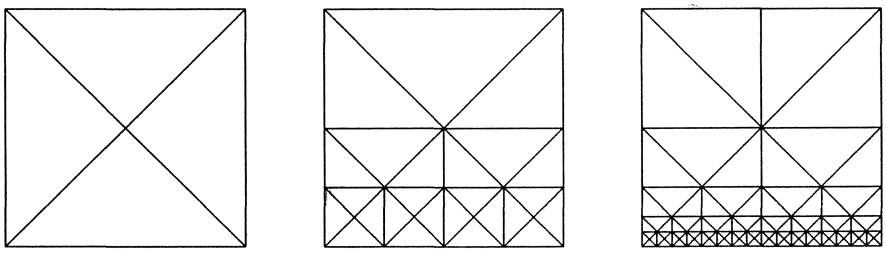


Fig. 6.1 Newest vertex bisection refinement along a line.

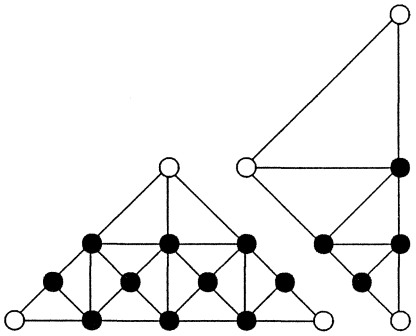


Fig. 6.2 Macro elements $\mathcal{A}_1$ and $\mathcal{A}_2$.

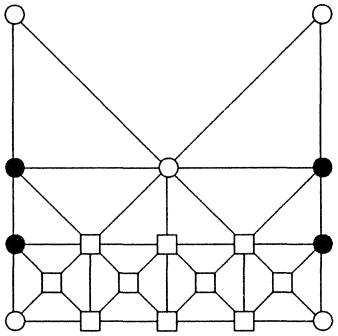


Fig. 6.3 Partitioning of the nodes for an approximation of the matrix A_{11}.

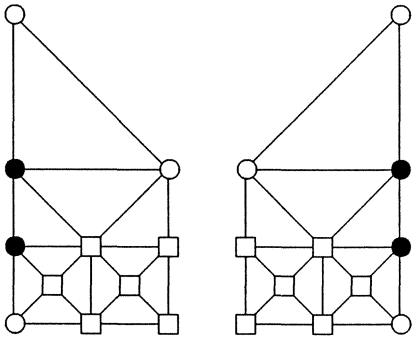


Fig. 6.4 Macro elements $\mathcal{C}_1$ and $\mathcal{C}_2$.

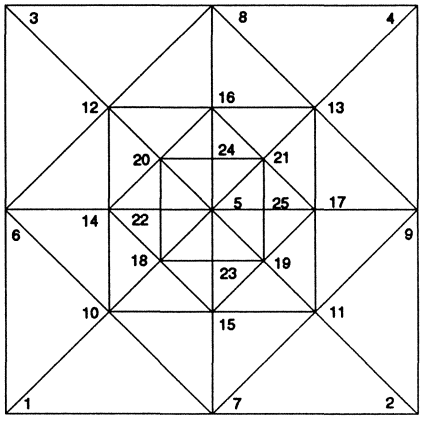


Fig. 6.5 Refinement around point.

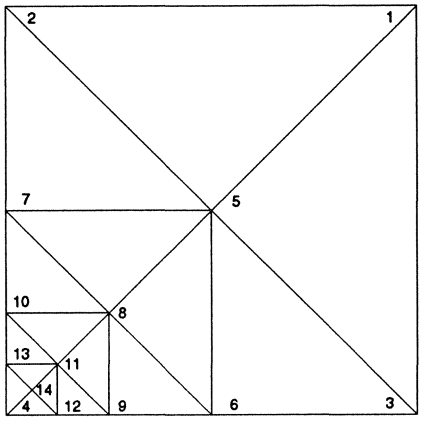


Fig. 6.6 Refinement into a corner.

7 On the updating and assembly of the Hessian matrix

As in: Axelsson O. and Maubach J., On the updating and assembly of the Hessian matrix in finite element methods, Computer Methods in Applied Mechanics and Engineering 71(1988),41-67.
The section with numerical results is extended with the introduction of the Mach number and an explanation of the modified diffusion tensor used. The numerical examples domain is taken to be an airplane wing.
Reprinted with the kind permission of the Elsevier Sequoia, Lausanne, Switzerland, publishers of Computer Methods in Mechanics and Engineering; part of this chapter is copyrighted by Elsevier Sequoia.

Abstract

For many non-linear differential equation solvers the assembly of the Hessian matrix is an expensive task. Hence numerical methods like the damped inexact Newton method (see [2], [11]), are often too expensive to use as the Hessian matrix has to be updated many times. The goal of this chapter is to show that in some special but frequently occurring cases, where the Hessian matrix has a special structure, updating it is not more expensive than updating the gradient vector if a particular factorization of the stiffness matrix is used. In these cases most of the computing time spent in the non-linear solver will be used to solve linear systems of equations, and the assembly of the Hessian matrix usually becomes a minor task. The present chapter extends earlier results of this nature in [3] and in [6].

Key words: Finite elements, Galerkin methods, Quadrature formulas
AMS(MOS) subject classifications: 65N30, 65D32

7.1 Introduction

This chapter considers the assembly of Hessian matrices in finite element methods. Finite element solution of non-linear differential equations like the electromagnetic field equation, the torsion of an elastic bar, or the full potential equation in aerodynamics, leads to a discretized non-linear system of equations $F(x) = 0$ in N variables, where N can be very large. Often non-linear elliptic differential equations are solved with the use of an iterative damped inexact Newton method or with multigrid methods (see e.g. [14], [15] where a flow problem is solved). In the present chapter Newton methods are considered because they involve the assembly of a Hessian matrix.

A damped inexact Newton method is an iterative method in which one encounters the approximate solution of a system of equations. The problem is to find a search direction $p^{(k)}$ such that $\|F'(x^{(k)})p^{(k)} + F(x^{(k)}))\| < \rho^{(k)}\|F(x^{(k)})\|$, or more generally

$$\|C_k^{-1}\left[F'(x^{(k)})p^{(k)} + F(x^{(k)})\right]\| < \rho^{(k)}\|C_k^{-1}F(x^{(k)})\|.$$

Then $x^{(k+1)} = x^{(k)} + \tau^{(k)}p^{(k)}$. C_k is a preconditioner of $F'(x^{(k)})$, $\rho^{(k)}$ a forcing sequence, $\tau^{(k)} > 0$ a damping parameter to be chosen small enough to obtain global convergence, and the matrix $F'(x^{(k)})$ is the derivative of the residual F at the point $x^{(k)}$. F' and F are also called *Hessian matrix* and *gradient*. For recent presentations of such methods, see [2] and [11].

Using a finite element method for a non-linear system of equations, one has to approximate $F'(x^{(k)})$ and $F(x^{(k)})$ because of the integrals imposed by the variational formulation. This approximation of $F'(x^{(k)})$ and $F(x^{(k)})$ often is very expensive to compute because it involves the repeated evaluation of integrals of non-linear functions of the finite element basis functions. In the past this has led to several techniques of (avoidance of) updating the Hessian matrix:

- only updating the Hessian matrix every p-th step or only updating it in regions of the domain of definition where it varies relatively much (see e.g. [8])
- the assembly of an approximation of $(F'(x^{(k)}))^{-1}$ preserving the sparsity structure patterns of the original Hessian matrix (see [10])
- assembly of a factorization of the Hessian matrix on element level in order to avoid the factorization of the global matrix (see e.g. [1]

where the natural factor method with a QR factorization, with Q an orthogonal matrix and R a triangular matrix, is presented)

- assembly of the Hessian matrix with exact integration but for a slightly modified vector $\tilde{x}^{(k)}$ instead of $x^{(k)}$ (see [8]).

The Hessian matrix can safely be approximated because it serves only to determine a search direction $p^{(k)}$. However, if it is nearly singular then it may be necessary to compute a very accurate approximation, or even the exact Hessian matrix, in order for it to be positive definite. In critical regions (regions with steep gradients or almost singular Hessian) this is difficult due to the large quadrature errors caused by fast varying functions within the finite elements.

The gradient has always to be approximated rather accurately because it serves as a stopping criterion of the iterative algorithm.

Linearizing a partial differential equation $F(x) = 0$ with a damped inexact Newton method leads to a sequence of linear equations $F'(x^{(k)})p^{(k)} = -F(x^{(k)})$. If we discretize this system by using a variational formulation and choosing appropriate basis functions and quadrature rules for the integration, this leads to linear systems of equations $F'(x^{(k)})p^{(k)} = F(x^{(k)})$, with $F'(x^{(k)})$ the Hessian matrix, $F(x^{(k)})$ the gradient vector and the search direction $p^{(k)}$ to be solved. In many important practical problems the matrix F' turns out to have a special structure which makes its updating cheap compared to the assembly of the gradient vector and the solution of the linear system. This will be shown by means of an example but the techniques provided can be used to derive cheap formulas for other cases as well.

A matrix F' associated with non-linear operators in divergence form, which will be called *divergence form stiffness matrix*, or for short, *stiffness matrix*, has the special structure mentioned above. For this class of matrices it will be proved in section 7.3 that there exists a factorization BMB^T, depending only on the basis functions and quadrature formulas used for the finite element discretization (section 7.2). For some frequently used basis functions and quadrature formulas the matrices B and M are cheap to compute elementwise. In some specific cases all errors caused by quadrature formulas are captured in the matrix M.

The factorization has some general advantages. Matrix B, for

instance, depends only on the grid geometry of the finite element grid, on the finite element basis functions and on the quadrature formulas used. If $F' \equiv F'(u)$ is a non-linear matrix, depending on a function u, then only M depends on this function u. Frequently M is a block diagonal matrix. The proposed factorization extends earlier results in [3] and [6] of this nature.

Similar factorizations of the stiffness matrix are well-known for finite element matrices derived from frame structures and electric networks, see for instance for early references [16], [17] and [18]. To the best of the authors knowledge, however, not much has previously appeared on this topic for finite element methods for continuous problems and for non-linear problems. However, in his recent book [22], Strang comments much about such factorizations.

7.2 Definition of a stiffness matrix

A discretization for the linearized differential equation is determined completely by the choice of basis functions and quadrature rules used to evaluate the integrals appearing in the variational formulation. Therefore the stiffness matrices will be presented after a short comment on these two choices. Thereafter it will be shown that the Hessian of differential equations in divergence form is often a stiffness matrix.

First consider the choice of the basis functions. Assume that $\Omega \in \mathbb{R}^n$ is an open and bounded polygonal domain with a finite element grid $\cup_{l=1}^{L} \Omega_l$ and global C^0 finite element functions. The local basis functions $\{\varphi_r^{(l)}\}_{r=1}^{T_l}$ on Ω_l are chosen such that the space $\mathbb{P}_{r_l}$ of polynomials up to degree r_l over Ω_l satisfies $\mathbb{P}_{r_l} \subset V_l$, where V_l is spanned by these basis functions (see fig. 7.2 for an example of $\mathbb{P}_i$). The global basis functions $\{\varphi_i\}_{i=1}^{N}$ are constructed in the usual manner from their contribution of the local basis functions on each element Ω_l. The set spanned by the global basis functions will henceforth be denoted by V.

Definition A quadrature rule is of degree d if it is exact for all polynomials in $\mathbb{P}_d$.

Now, consider the choice of a *quadrature rule* Q. On the topological closure of each element $\overline{\Omega_l}$, an element quadrature rule Q_l defined by a set of quadrature points $\{N_k^{(l)}\}_{k=1}^{q_l}$ and weights $w_k^{(l)} \in \mathbb{R}$ is chosen such that

$$\begin{aligned} Q_l(h) &\equiv \sum_{k=1}^{q_l} w_k^{(l)} \cdot h(N_k^{(l)}) \approx \int_{\Omega_l} h \, dx \\ Q_l(\frac{\partial}{\partial x_p}\varphi_r^{(l)} \frac{\partial}{\partial x_q}\varphi_s^{(l)}) &= \int_{\Omega_l} \frac{\partial}{\partial x_p}\varphi_r^{(l)} \frac{\partial}{\partial x_q}\varphi_s^{(l)} \, dx \end{aligned} \tag{7.2.1}$$

for all $h \in \mathrm{H}_0^1(\Omega)$, $\varphi_r^{(l)}, \varphi_s^{(l)} \in V_l$ and all $p, q \in \{1, \ldots, n\}$. Note that the element quadrature rule depends on the degree of the polynomial basis functions in V_l. If for instance $V_l = \mathbb{P}_{r_l}$ then Q_l has to be exact for all polynomials of $\mathbb{P}_{2(r_l-1)}$.

Now the quadrature rule Q on Ω is defined by

$$Q(h) \equiv \sum_{l=1}^{L} Q_l(h) \approx \int_{\Omega} h \, dx. \tag{7.2.2}$$

It will be used to approximate all domain integrals such as those appearing in stiffness matrices, from now on. Some well-known and easily derived properties are:

Lemma 7.2.1 *Let Q be defined as above then*

- *Q is a linear and continuous functional*
- *If the derivative of a function $G(u)$ is denoted by $\frac{\partial}{\partial v}G(u)$, this yields*

$$\frac{\partial}{\partial v}\{Q(G(u))\} = \lim_{\varsigma \downarrow 0}\{\frac{Q(G(u+\varsigma v)) - Q((G(u))}{\varsigma}\} = Q(\frac{\partial}{\partial v}G(u))$$

- *The sets of weights satisfy $\sum_{k=1}^{q_l} w_k^{(l)} = Area(\Omega_l)$ on each element Ω_l*
- *If all weights are positive then Q is monotone: $h \geq 0 \Rightarrow Q(h) \geq 0$.* □

Remark 7.2.1 The basis functions can have different orders of polynomial degree r_l in different elements Ω_l. However, it must be assumed that the finite element approximation is C^0, which typically leads to a situation as in figure 7.1.

Now the class of *stiffness matrices* is introduced. A matrix H is a *stiffness matrix* iff

$$[H]_{ij} = Q((A\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i) \tag{7.2.3}$$

for some n by n positive definite matrix A on Ω, where n is the number of space dimensions.

Especially for the solution of non-linear problems, where $A \equiv A(\boldsymbol{x}, \underline{\nabla}u)$ the efficient updating of these matrices is important because they vary from one iteration step to the next. Also the subsequent use of grid refinement, in order to approximate layers more accurately, makes a reevaluation necessary.

Stiffness matrices occur in many important practical problems. Consider for instance the following non-linear equation on divergence form:

$$\begin{aligned} -\nabla\cdot(A(\boldsymbol{x}, \underline{\nabla}u)\underline{\nabla}u) &= f \text{ on } \Omega \\ u &= 0 \text{ on } \Gamma_D \\ A(\boldsymbol{x}, \underline{\nabla}u)\underline{\nabla}u^T\boldsymbol{n} &= g \text{ on } \Gamma_N \end{aligned} \tag{7.2.4}$$

where

- $\Omega \subset \mathbb{R}^n$ open and bounded,
- $\Gamma_D \cup \Gamma_N = \partial\overline{\Omega}$, $\Gamma_D \cap \Gamma_N = \emptyset$,
- $f, g: \Omega \mapsto \mathbb{R}$ sufficiently smooth given functions,
- $A \equiv [a_{p,q}]$ an n by n matrix, $a_{p,q}$ given functions on Ω,
- $\boldsymbol{n}$ the unit outer normal vector.

Depending on the matrix A in (7.2.4) the differential equation can, for instance, represent

- the electromagnetic field equations, or torsion of an elastic bar. Here
$$A = \text{Diag}(a_{11}, \ldots, a_{nn})$$
with $a_{pp} \equiv a_{pp}(|\underline{\nabla}u|^2)$. The latter functions can differ by a large factor between two areas of different materials

- the potential equation in aerodynamics. In this case $A = aI_n$ with the gas density function $a \equiv a(|\underline{\nabla} u|^2) = c_\infty(1 - \tilde{\gamma}(|\underline{\nabla} u|^2 - v_\infty^2))^{1/(2\tilde{\gamma})}$ and I_n the identity matrix of order n. For more information see section 7.9
- Navier's equations of stress and strain which give a coupled system of equations (see e.g. [7]). In the linearized case A is the constant coefficient matrix

$$\begin{bmatrix} a & 0 & 0 & c \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ c & 0 & 0 & a \end{bmatrix}.$$

  For more information about this system of equations see section 7.7
- the classical stiffness matrix $[\Lambda]_{ij} = Q((I_n \underline{\nabla} \varphi_j)^T \underline{\nabla} \varphi_i)$, which is a discretization of the Laplacian on the domain Ω.

In all the problems above, A is a nonsingular matrix.

As promised in the introduction, it will now be shown that the Hessian sometimes is a stiffness matrix. Therefore, consider the classical variational formulation of (7.2.4): find $u \in \mathrm{H}_0^1(\Omega)$ such that

$$\int_\Omega (A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla} u)^T \underline{\nabla} v \, d\boldsymbol{x} = \int_\Omega f v \, d\boldsymbol{x} + \oint_{\Gamma_N} g v \, ds \quad \forall_{v \in \mathrm{H}_0^1(\Omega)} \quad (7.2.5)$$

where $\mathrm{H}^1(\Omega)$ is the Sobolev space of first order and (in the sense of traces) $\mathrm{H}_0^1(\Omega) = \{u \in \mathrm{H}^1(\Omega) : u = 0 \text{ on } \Gamma_D\}$. The (source) terms to the right in the equation are of no importance, so we disregard them in the following discussion.

The gradient $F(u)$ can be defined as a dual operator on $v \in \mathrm{H}_0^1(\Omega)$ by

$$\langle F(u), v \rangle = \int_\Omega (A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla} u)^T \underline{\nabla} v \, d\boldsymbol{x}.$$

In discretized form this becomes $[F(u)]_i = Q((A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla} u)^T \underline{\nabla} \varphi_i)$ whence the gradient is closely related to a stiffness matrix. For some special choices of A its derivative is a stiffness matrix:

Theorem 7.2.1 *Suppose matrix G is defined by*

$$[G]_{ij} = Q((A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla} u)\underline{\nabla}\varphi_i)$$

for some matrix A of order n with entries $a_{p,q} \equiv a_{p,q}(\boldsymbol{x}, \underline{\nabla} u)$. *Then the derivative G' of G is given by*

$$[G']_{ij} = Q((E(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i)$$

for a certain matrix E of order n.

Proof. An elementary computation for space dimension $n = 2$ yields (see chapter 3, lemma 3.3.1 for details)

$$E{=}2\cdot\begin{bmatrix} \frac{\partial}{\partial u_x}a_{11}u_x{+}\frac{\partial}{\partial u_x}a_{12}u_y & \frac{\partial}{\partial u_y}a_{11}u_x{+}\frac{\partial}{\partial u_y}a_{12}u_y \\ \frac{\partial}{\partial u_x}a_{21}u_x{+}\frac{\partial}{\partial u_x}a_{22}u_y & \frac{\partial}{\partial u_y}a_{21}u_x{+}\frac{\partial}{\partial u_y}a_{22}u_y \end{bmatrix}{+}A(\boldsymbol{x}, \underline{\nabla} u). \tag{7.2.6}$$

For higher space dimensions an analogous formula can be derived. Note that this theorem is also correct if Q represents exact integration. □

7.3 Factorization of a stiffness matrix

To prove the existence of the proposed factorization for a stiffness matrix H an additional set of Lagrangian basis functions is introduced. This set is closely related to a mixed variational formulation of (7.2.4) (see e.g. [3] and [6]) which will be discussed in section 7.8.

Connected to the set of quadrature points on each element define the Langrangian polynomial basis functions $\{\psi_k^{(l)}\}_{k=1}^{q_l}$ by

$$\psi_m(N_k^{(l)}) = \delta_{m,k} \tag{7.3.1}$$

where $\delta_{m,k}$ is the Kronecker function. To *each* such local function $\psi_k^{(l)}$ a unique discontinuous global function $\psi_{i_k^{(l)}}$ is defined by

$$\psi_{i_k^{(l)}} = \begin{cases} \psi_k^{(l)} & \text{on } \overline{\Omega_l} \\ 0 & \text{on } \overline{\Omega_l}^c \end{cases} \tag{7.3.2}$$

where $\overline{\Omega_l}^c$ is the complement of $\overline{\Omega_l}$ in Ω (note that the index mapping $(l,k) \mapsto i_k^{(l)}$ is a bijection). Hence each local function $\psi_k^{(l)}$ induces one global function $\psi_{i_k^{(l)}}$ which has only support on Ω_l. The set of all global functions $\psi_{i_k^{(l)}}$ is (omitting the subscript $i_k^{(l)}$) denoted by $\{\psi_i\}_{i=1}^{N'}$ where $N' = \sum_{l=1}^{L} q_l$. Henceforth W_l and M will denote the set spanned by these local respectively global basis functions.

Lemma 7.3.1 *Let the local basis functions $\{\psi_k^{(l)}\}_{k=1}^{q_l}$ be defined as above. Then*

$$Q(\psi_{i_k^{(l)}} h) = w_k^{(l)} \cdot h(N_k^{(l)})$$

for all functions h, all $l \in \{1,\ldots,L\}$ and all $k \in \{1,\ldots,q_l\}$.

Proof. Note that the global basis function $\psi_{i_k^{(l)}}$ has only support on the element l according to (7.3.2). Therefore $Q(\psi_{i_k^{(l)}} h) = Q_l(\psi_k^{(l)} h)$. Because $\psi_k^{(l)}(N_m^{(l)}) = \delta_{k,m}$ on Ω_l this completes the proof. If $h(N_k^{(l)}) = 0$ then by definition $Q^{-1}(\psi_{i_k^{(l)}} h^{-1}) \equiv 0$. □

Now consider the factorization theorem:

Theorem 7.3.1 *Let the N by N stiffness matrix H be defined by*

$$[H]_{ij} = Q((A\underline{\nabla}\varphi_j)^T \underline{\nabla}\varphi_i),$$

for some matrix A of order n, with the use of finite element basis functions $\{\varphi_i\}_{i=1}^{N}$, a quadrature formula Q and corresponding basis functions $\{\psi_i\}_{i=1}^{N'}$. Then there exists a factorization

$$H = BMB^T \tag{7.3.3}$$

where B is an N by $n \cdot N'$ rectangular block matrix,

$$B = \left[B^{(1)} \cdots B^{(n)}\right] \quad \text{with blocks} \quad [B^{(p)}]_{ij} = Q(\psi_j \frac{\partial}{\partial x_p} \varphi_i) \tag{7.3.4}$$

for all $p \in \{1,\ldots,n\}$, all $i \in \{1,\ldots,N\}$ and all $j \in \{1,\ldots,N'\}$

and M is a square block matrix of order $n \cdot N'$,

$$M = \begin{bmatrix} M^{(1,1)} & \dots & M^{(1,n)} \\ \vdots & \ddots & \vdots \\ M^{(n,1)} & \dots & M^{(n,n)} \end{bmatrix} \tag{7.3.5}$$

with blocks

$$M^{(p,q)} = \mathrm{Diag}(Q^{-1}(a_{p,q}^{-1}\psi_j^2))$$

for all $p, q \in \{1, \dots, n\}$ and all $j \in \{1, \dots, N'\}$.

Proof. By a straightforward computation one gets

$$\begin{aligned}
[H]_{ij} &= Q((A\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i) \\
&= \sum_{l=1}^{L} Q_l((A\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i) \\
&= \sum_{l=1}^{L}\sum_{p,q=1}^{n} Q_l(a_{p,q}\frac{\partial}{\partial x_p}\varphi_i\frac{\partial}{\partial x_q}\varphi_j) \\
&= \sum_{l=1}^{L}\sum_{p,q=1}^{n}\sum_{k=1}^{q_l}\{\frac{\partial}{\partial x_p}\varphi_i(N_k^{(l)})a_{p,q}(N_k^{(l)})\frac{\partial}{\partial x_q}\varphi_j(N_k^{(l)})\}\cdot w_k^{(l)} \\
&= \sum_{l=1}^{L}\sum_{p,q=1}^{n}\sum_{k=1}^{q_l}\{Q_l(\psi_k^{(l)}\frac{\partial}{\partial x_p}\varphi_i)Q_l^{-1}(a_{p,q}^{-1}\psi_k^{(l)^2})Q_l(\psi_k^{(l)}\frac{\partial}{\partial x_q}\varphi_j)\} \\
&= \sum_{p,q=1}^{n}\sum_{k=1}^{N'}[B^{(p)}]_{ik}[M^{(p,q)}]_{kk}[B^{(q)^T}]_{kj} \\
&= \sum_{p,q=1}^{n}[B^{(p)}M^{(p,q)}B^{(q)^T}]_{ij} \\
&= [BMB^T]_{ij}
\end{aligned}$$

according to lemma 7.3.1, (7.3.4) and (7.3.5). □

Remarks 7.3.1

- According to lemma 7.3.1, B is easy to compute (assume index $j = i_k^{(l)}$):

$$[B^{(p)}]_{ij} = \begin{cases} w_k^{(l)}\frac{\partial}{\partial x_p}\varphi_r^{(l)}(N_k^{(l)}) & \text{if } \varphi_{i|\Omega_l} = \varphi_r^{(l)} \\ 0 & \text{if } \varphi_{i|\Omega_l} \equiv 0\,. \end{cases} \tag{7.3.6}$$

The sparsity structure of B can be very irregular but it will be easy to compute elementwise, as will be shown in section 7.4. It turns out that there is no need to assemble B.

- Note that the submatrices $B^{(p)^T}$ are discretizations of the derivative operators on Ω, i.e., $B^{(p)^T}$ is the discretization of $f \mapsto \frac{\partial}{\partial x_p} f$ for functions f defined on Ω
- Matrix M is also easy to compute and store; according to (7.3.5) all $M^{(p,q)}$ are diagonal matrices
- If the set of Lagrangian basis functions W_l satisfies

$$V_l \subset \mathbb{P}_{r_l} \Rightarrow W_l \subset \mathbb{P}_{r_l - 1} \tag{7.3.7}$$

then

$$Q_l(\psi_k^{(l)} \frac{\partial}{\partial x_p} \varphi_r^{(l)}) = \int_{\Omega_l} \psi_k^{(l)} \frac{\partial}{\partial x_p} \varphi_r^{(l)} \, dx$$

which implies that quadrature errors will only arise during the calculation of matrix M and matrix B is calculated exactly. This is possible for triangular grid elements with linear or quadratic basis functions (for instance see table 7.3.1).

In section 7.4 it will be shown that a stiffness matrix H can be computed cheaply elementwise. But, since quadrature rules and basis functions play an important role in the factorization first a brief overview of some possible choices.

Example 7.3.1 Suppose that triangular grid elements and cubic basis functions are used. Then $V_l = \mathbb{P}_3$, which implies that the quadrature rule must at least be of degree $2 \cdot (3 - 1) = 4$ for the factorization of H, according to (7.2.1). Further (7.3.7) implies $W_l \subset \mathbb{P}_2$ whence there may be at most $6 = \mathrm{Dim}(\mathbb{P}_2)$ quadrature points (cf. fig. 7.2) if one wants the quadrature errors to be restricted to the calculation of matrix M.

Example 7.3.2 Depending on the degree of the polynomial basis functions $\tilde{\varphi}$ on the two-dimensional reference triangle $\tilde{\Omega} = \{(x, y) \in \mathbb{R}^2 : x > 0, y > 0, x+y < 1\}$ table 7.3.1 and fig. 7.3 gives some quadrature rules $\tilde{Q}$ ($A = Area(\tilde{\Omega})$). The quadrature rule on each specific element Ω_l can be

Table 7.3.1 Examples of quadrature rules on triangles.

basis function type	$\tilde{d}$	$\tilde{q}$	quadrature rule $\tilde{Q}$
linear on a triangle, degree 1 ($\tilde{V} \subset \mathbb{P}_1$)	1	1	1-point midpoint: $N = (\frac{1}{3}, \frac{1}{3}); w_1 = A$
quadratic on a triangle, degree 2 ($\tilde{V} \subset \mathbb{P}_2$)	2	3	3-point Gauss: $N_1 = (\frac{1}{2}, 0); w_1 = \frac{1}{3}A$ $N_2 = (0, \frac{1}{2}); w_2 = \frac{1}{3}A$ $N_3 = (\frac{1}{2}, \frac{1}{2}); w_3 = \frac{1}{3}A$
cubic on a triangle, degree 4 ($\tilde{V} \subset \mathbb{P}_3$) $z = \frac{155+\sqrt{15}}{1200}A$ $y = \frac{155-\sqrt{15}}{1200}A$ $r = \frac{6-\sqrt{15}}{21}, s = \frac{9+2\sqrt{15}}{21}$ $t = \frac{6+\sqrt{15}}{21}, v = \frac{9-2\sqrt{15}}{21}$	4	6	Radons 7-point: $N_1 = (\frac{1}{3}, \frac{1}{3}); w_1 = \frac{9}{40}A$ $N_2 = (r, r); w_2 = y$ $N_3 = (r, s); w_3 = y$ $N_4 = (s, r); w_4 = y$ $N_5 = (t, t); w_5 = z$ $N_6 = (v, t); w_6 = z$ $N_7 = (t, v); w_7 = z$

derived through

$$\int_{\Omega_l} h \, dx = \int_{\tilde{\Omega}} \tilde{h} |\mathrm{Det}(\tilde{J})| \, dx = |\mathrm{Det}(\tilde{J})| \int_{\tilde{\Omega}} \tilde{h} \, dx \approx |\mathrm{Det}(\tilde{J})| \cdot \tilde{Q}(\tilde{h})$$

where $\mathrm{Det}(\tilde{J})$ is the determinant of Jacobian of the affine transformation from $\tilde{\Omega}$ on to Ω_l and h the accordingly transformed function $\tilde{h}$. The table does not hold if isoparametric basis functions are used.

The second column of the table shows the necessary degree $\tilde{d}$ of the quadrature rule and the maximal number of quadrature points $\tilde{q}$ for which (7.2.1) respectively (7.3.7) are satisfied. Further an example of a quadrature rule found in recent literature (see e.g. [24]) is presented in the fourth column. Note that all proposed quadrature formulas have positive weights whence the signs of the transformed weights $(\tilde{w}_k \cdot |\mathrm{Det}(\tilde{J})|)$ are also positive.

Example 7.3.3 Table 7.3.2 and fig. 7.5 present some quadrature rules for rectangular elements. Note that in this case (7.3.7) is not satisfied. For a general space dimension n it is possible to find quadrature rules with positive weights of degree d_l satisfying (7.2.1) (see [24], pages 54, 55). For more recent information about quadrature rules see for example [12], [19], [13] and the references therein.

Table 7.3.2 Examples of quadrature rules on rectangles.

basis function type	$\tilde{d}$	$\tilde{q}$	quadrature rule $\tilde{Q}$
bilinear on a rectangle, degree 3 ($\tilde{V} \subset \mathbb{P}_2$)	2	3	4-point Gauss: $N_1 = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}); w_1 = \frac{1}{4}$ $N_2 = (-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}); w_2 = \frac{1}{4}$ $N_3 = (-\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}); w_3 = \frac{1}{4}$ $N_4 = (\frac{1}{\sqrt{3}}, -\frac{1}{\sqrt{3}}); w_4 = \frac{1}{4}$
biquadratic on a rectangle, degree 6 ($\tilde{V} \subset \mathbb{P}_4$)	6	10	Tylers 12-point (see e.g. [24])

7.4 Cheap evaluation of a stiffness matrix

Here the factorization method will be explored for a special choice of finite element basis functions and quadrature formula. It appears to be very cheap for this particular choice.

Consider an open and bounded two-dimensional polygonal domain Ω with triangular grid elements. On each grid element Ω_l standard biquadratic basis functions $\{\varphi_r^{(l)}\}_{r=1}^6$ are used (see e.g. [5], [25] or [23]). According to (7.2.1) the quadrature rule must be of second order so a 3-point Gauss quadrature formula (see table 7.3.1) is used on each element:

$$Q_l(h) = \sum_{k=1}^{3} w_k^{(l)} \cdot h(N_k^{(l)}) \tag{7.4.1}$$

where $N_k^{(l)}$ are the edge midpoints and $w_k^{(l)} = \frac{1}{3}\text{Area}(\Omega_l)$. This implies that there have to be three local Lagrangian basis functions $\{\psi_k^{(l)}\}_{k=1}^3$

(see fig. 7.4). In [6] linear basis functions are chosen to this end, but because only property (7.3.1) is used, no further specification is needed.

Now consider the assembly of the stiffness matrix

$$[H]_{ij} = \mathcal{Q}((A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla}\varphi_j)^T \underline{\nabla}\varphi_i)$$

for an arbitrary 2 by 2 matrix A, which will be done at element level, with 6 by 6 element matrices

$$[H_l]_{rs} = Q_l((A(\boldsymbol{x}, \underline{\nabla} u)\underline{\nabla}\varphi_s^{(l)})^T \underline{\nabla}\varphi_r^{(l)}). \tag{7.4.2}$$

In theorem 7.3.1 $\Omega \subset \mathbb{R}^n$ may be chosen arbitrary whence (choose Ω_l)

$$H_l \equiv \sum_{p,q=1}^{2} H_l^{(p,q)} \equiv \sum_{p,q=1}^{2} B_l^{(p)} M_l^{(p,q)} B_l^{(q)^T} \tag{7.4.3}$$

with $M_l^{(p,q)}$ a 3 by 3 diagonal matrix and $B_l^{(p)}$ a full 6 by 3 matrix (space dimension $n = 2$, number of finite element basis functions $N = 6$ and the number of quadrature points $N' = 3$).

In this particular case investigation of matrix $B_l^{(p)}$ shows that it has a simple structure: instead of eighteen possibly different entries it has only six different entries which are furthermore related as follows:

$$B_l^{(p)^T} = \begin{bmatrix} -\alpha & \beta & \gamma & -\delta & \delta & \delta \\ \alpha & -\beta & \gamma & \epsilon & -\epsilon & \epsilon \\ \alpha & \beta & -\gamma & \lambda & \lambda & -\lambda \end{bmatrix} \tag{7.4.4}$$

where $\alpha + \beta + \gamma = 0$ and

$$\begin{aligned} \alpha &= -Q_l(\psi_1^{(l)} \frac{\partial}{\partial x_p} \varphi_1^{(l)}) = -\tfrac{1}{3}\text{Area}(\Omega_l) \frac{\partial}{\partial x_p} \varphi_1^{(l)}(N_1^{(l)}) & \delta &= 2 \cdot \alpha \\ \beta &= -Q_l(\psi_1^{(l)} \frac{\partial}{\partial x_p} \varphi_2^{(l)}) & \epsilon &= 2 \cdot \beta \\ \gamma &= -Q_l(\psi_1^{(l)} \frac{\partial}{\partial x_p} \varphi_3^{(l)}) & \lambda &= 2 \cdot \gamma . \end{aligned}$$

The regular structure of this element matrix is caused by the fact that its entries are the values of the partial derivatives of the basis

functions at the quadrature points. Because these partial derivatives are simple polynomial expressions some of their values at different quadrature points are equal, leading to less different entries in $B_l^{(p)}$.

In the non-linear case there is no need at all to store matrix B in memory if all the partial derivatives of the basis functions $\frac{\partial}{\partial x_p}\varphi_r^{(l)}$ are computed to evaluate $\frac{\partial}{\partial x_p}u = \sum_{i=1}^{N} \alpha_i \frac{\partial}{\partial x_p}\varphi_i$.

Note that Q evaluates $B_l^{(p)}$ exactly in this case (see (7.3.6)) whence all quadrature errors made are limited to the matrix M.

According to (7.3.5) the matrix $M_l^{(p,q)}$ is described by

$$M_l^{(p,q)} = \mathrm{Diag}(\rho, \sigma, \tau) \tag{7.4.5}$$

where

$$\rho = (Q_l(a_{p,q}^{-1}\{\psi_1^{(l)}\}^2))^{-1} = (\tfrac{1}{3}\mathrm{Area}(\Omega_l))^{-1} a_{p,q}(N_1^{(l)}),$$
$$\sigma = (Q_l(a_{p,q}^{-1}\{\psi_2^{(l)}\}^2))^{-1} \text{ and } \tau = (Q_l(a_{p,q}^{-1}\{\psi_3^{(l)}\}^2))^{-1}.$$

Denoting the entries of $B_l^{(q)}$ as those in $B_l^{(p)}$, but with additional primes, an explicit formula for $H_l^{(p,q)}$ can be constructed

$$H_l^{(p,q)} = B_l^{(p)} M_l^{(p,q)} B_l^{(q)T} = \begin{bmatrix} C^{(1)} & C^{(3)} \\ C^{(2)} & C^{(4)} \end{bmatrix} \tag{7.4.6}$$

where $C^{(1)}$, $C^{(2)}$, $C^{(3)}$ and $C^{(4)}$ are given by

$$\begin{bmatrix} (\rho+\sigma+\tau)\alpha\alpha' & (-\rho-\sigma+\tau)\alpha\beta' & (-\rho+\sigma-\tau)\alpha\gamma' \\ (-\rho-\sigma+\tau)\beta\alpha' & (\rho+\sigma+\tau)\beta\beta' & (\rho-\sigma-\tau)\beta\gamma' \\ (-\rho+\sigma-\tau)\gamma\alpha' & (\rho-\sigma-\tau)\gamma\beta' & (\rho+\sigma+\tau)\gamma\gamma' \end{bmatrix},$$
$$\begin{bmatrix} (\rho\delta'+\sigma\varepsilon'+\tau\lambda')\alpha & (-\rho\delta'-\sigma\varepsilon'+\tau\lambda')\alpha & (-\rho\delta'+\sigma\varepsilon'-\tau\lambda')\alpha \\ (-\rho\delta'-\sigma\varepsilon'+\tau\lambda')\beta & (\rho\delta'+\sigma\varepsilon'+\tau\lambda')\beta & (\rho\delta'-\sigma\varepsilon'-\tau\lambda')\beta \\ (-\rho\delta'+\sigma\varepsilon'-\tau\lambda')\gamma & (\rho\delta'-\sigma\varepsilon'-\tau\lambda')\gamma & (\rho\delta'+\sigma\varepsilon'+\tau\lambda')\gamma \end{bmatrix},$$
$$\begin{bmatrix} (\rho\delta+\sigma\epsilon+\tau\lambda)\alpha' & (-\rho\delta-\sigma\epsilon+\tau\lambda)\beta' & (-\rho\delta+\sigma\epsilon-\tau\lambda)\gamma' \\ (-\rho\delta-\sigma\epsilon+\tau\lambda)\alpha' & (\rho\delta+\sigma\epsilon+\tau\lambda)\beta' & (\rho\delta-\sigma\epsilon-\tau\lambda)\gamma' \\ (-\rho\delta+\sigma\epsilon-\tau\lambda)\alpha' & (\rho\delta-\sigma\epsilon-\tau\lambda)\beta' & (\rho\delta+\sigma\epsilon+\tau\lambda)\gamma' \end{bmatrix}$$

and

$$\begin{bmatrix} \rho\delta\delta'+\sigma\epsilon\epsilon'+\tau\lambda\lambda' & -\rho\delta\delta'-\sigma\epsilon\epsilon'+\tau\lambda\lambda' & -\rho\delta\delta'+\sigma\epsilon\epsilon'-\tau\lambda\lambda' \\ -\rho\delta\delta'-\sigma\epsilon\epsilon'+\tau\lambda\lambda' & \rho\delta\delta'+\sigma\epsilon\epsilon'+\tau\lambda\lambda' & \rho\delta\delta'-\sigma\epsilon\epsilon'-\tau\lambda\lambda' \\ -\rho\delta\delta'+\sigma\epsilon\epsilon'-\tau\lambda\lambda' & \rho\delta\delta'-\sigma\epsilon\epsilon'-\tau\lambda\lambda' & \rho\delta\delta'+\sigma\epsilon\epsilon'+\tau\lambda\lambda' \end{bmatrix}.$$

This is obviously a bit messy to program, but since the formula is grid independent it has to be done only once.

Definition A floating point operation (flop) will stand for a multiplication or a multiplication combined with an addition.

In order to compare the number of floating point operations needed to construct H_l in (7.4.2) respectively (7.4.3), it is assumed that all entries $a_{p,q}$ and all (partial derivatives) of basis functions $\frac{\partial}{\partial x_p}\varphi_r^{(l)}$ are calculated and stored in advance. The costs of this preliminary work are the costs for the evaluation at the quadrature points of

- $\underline{\nabla}\varphi_r^{(l)}$ and $\underline{\nabla}u$. For each quadrature point this is approximately 66 flops with an additional 6 flops per element whence the overhead costs are 204 flops.
- the entries of the matrix A. Suppose the evaluation of each $a_{p,q}$ costs about d flops, then the matrix evaluation costs are: $12d$ flops.

The additional costs with the use of (7.4.2) for a nonsymmetric matrix A without zero entries are $36 \cdot (2 + 3 \cdot 4 \cdot 2) = 936$ flops because

$$[H_l]_{rs} = \tfrac{1}{3}\text{Area}(\Omega_l)\sum_{k=1}^{3}\sum_{p,q=1}^{2} a_{p,q}(N_k^{(l)}) \cdot \frac{\partial}{\partial x_q}\varphi_r^{(l)}(N_k^{(l)}) \cdot \frac{\partial}{\partial x_p}\varphi_s^{(l)}(N_k^{(l)}). \tag{7.4.7}$$

For a symmetric matrix A without zero entries the additional costs are approximately halved: $21 \cdot 26 = 546$ flops.

Now suppose that (7.4.3) is used for the assembly. The additional costs for an nonsymmetric matrix A without zero entries are $2 \cdot 27 + 2 \cdot 45 = 144$ flops, divided among the four submatrices $H_l^{(p,q)}$ in (7.4.6) as follows:

- 27 flops for the assembly of $H_l^{(p,p)}$: $\begin{bmatrix} 12 & 0 \\ 12 & 3 \end{bmatrix}$ (in this case $C^{(3)^T} = C^{(2)}$)

- 45 flops for the assembly of $H_l^{(p,q)}$: $\begin{bmatrix} 18 & 12 \\ 12 & 3 \end{bmatrix}$ $(p \neq q)$.

If A is symmetric then the additional costs will be $2 \cdot 27 + 45 = 99$ flops.

From the above one can conclude that the use of (7.4.6) instead of (7.4.7) gives a reduction of floating point operations with a factor

$$\frac{936 + 12d + 204}{144 + 12d + 204}. \tag{7.4.8}$$

Therefore, one can see that the assembly of a stiffness matrix with the use of the factorization method is relatively cheap compared to the straightforward method given by (7.4.2) if d is not too large. Hence it can be updated frequently even in the non-linear case.

Remarks 7.4.1

- For linear basis functions on a triangular grid elements $\Omega \subset \mathbb{R}^2$ an improvement can be achieved in a similar manner (see e.g. [3] or [6])
- The element matrices $B_l^{(p)}$ are expected to have a regular structure, if for example other basis functions are used in three-dimensional problems, leading to a reduction of the computational work if a factorization technique is used
- Evaluating the basis functions with the use of local barycentric coordinates (L_1, L_2, L_3) as in [21], instead of local coordinates (x, y), is advisable because the former implementation will take fewer flops
- In some special but frequently occurring cases d is typically about 15 flops (see e.g. section 7.9) depending on the entries $a_{p,q}$. In more complicated cases where matrix A has more different entries $a_{p,q}$, parts of the expressions for these entries are often related.

7.5 Preconditioning of a stiffness matrix

It appears to be relatively easy to determine the condition number of a matrix $G^{-1}H$ for two positive definite stiffness matrices H and G, which is of interest if a preconditioned iterative linear solver is used. To

this end the matrices B and M are investigated more closely to derive an estimate for this condition number.

Now consider the following definitions

- The spectrum $\sigma(H)$ of a matrix H is the set of all eigenvalues of H
- Two matrices H and G are called spectrally equivalent if there exist positive scalars λ_1 and λ_2 such that for all vectors $\boldsymbol{x}$

$$\lambda_1 \boldsymbol{x}^T H \boldsymbol{x} \leq \boldsymbol{x}^T G \boldsymbol{x} \leq \lambda_2 \boldsymbol{x}^T H \boldsymbol{x}.$$

Theorem 7.5.1 *Matrix BB^T is a stiffness matrix which is spectrally equivalent to the stiffness matrix Λ if all weights $w_k^{(l)}$ are positive.*

Proof. Assume that on all elements l quadrature formulas Q_l are used with positive weights. Note further that $BB^T = \sum_{p=1}^{n} B^{(p)} B^{(p)^T}$ according to theorem 7.3.1. Define the scalars $w_{\min}, w_{\max}$ to be the extreme quadrature weights of all weights $w_k^{(l)}$. Further let the matrix C of order n be defined by $C = \mathrm{Diag}(c, \ldots, c)$ with the function c satisfying

$$\begin{cases} c(N_k^{(l)}) = w_k^{(l)} > 0 \\ w_{\min} \leq \{c(\boldsymbol{x}) : \boldsymbol{x} \in \Omega\} \leq w_{\max} \end{cases} \tag{7.5.1}$$

on each element Ω_l for all $k \in \{1, \ldots, q_l\}$. Define a stiffness matrix S analogous to (7.2.3) where matrix C replaces A. Because C is a diagonal matrix

$$[S]_{ij} = Q((C \underline{\nabla} \varphi_j)^T \underline{\nabla} \varphi_i) \equiv \sum_{p=1}^{n} B^{(p)} M_c^{(p,p)} B^{(p)^T}$$

where $M_c^{(p,p)}$ are N' by N' diagonal matrices with $[M_c^{(p,p)}]_{mm} = Q^{-1}(c^{-1}{\psi_m}^2)$. According to (7.3.2) and lemma 7.3.1, $M_c^{(p,p)} = I_{N'}$ whence $S = BB^T$. Hence BB^T is a stiffness matrix.

Now let $u = \sum_{i=1}^{N} \alpha_i \varphi_i$ and define $\boldsymbol{u} = [\alpha_1, \ldots, \alpha_N]^T$. Then obviously

$$(BB^T \boldsymbol{u}, \boldsymbol{u}) = Q((C \underline{\nabla} u)^T \underline{\nabla} u)$$

where $(\cdot, \cdot)$ denotes the Eucledian inner product on $\mathbb{R}^N$. Further C is a positive definite diagonal matrix on Ω whence (Q is monotone according to lemma 7.2.1)

$$(BB^T\boldsymbol{u}, \boldsymbol{u}) = Q((C\underline{\nabla}u)^T\underline{\nabla}u) \geq w_{\min} \cdot Q((I\underline{\nabla}u)^T\underline{\nabla}u) = w_{\min} \cdot (\Lambda\boldsymbol{u}, \boldsymbol{u})$$

and analogously $(BB^T\boldsymbol{u}, \boldsymbol{u}) \leq w_{\max} \cdot (\Lambda\boldsymbol{u}, \boldsymbol{u})$. Hence the matrices BB^T and Λ are spectrally equivalent. □

Remarks 7.5.1

- For a given bounded domain $\Omega \subset \mathbb{R}^n$ there always exists a quadrature rule of degree d such that all quadrature points belong to Ω and such that all weights are positive (see e.g. [24], page 58 and further)
- If the assumptions of theorem 7.5.1 are satisfied then $\sigma(BB^T) \subset (0, \infty)$ because $\sigma(\Lambda) \subset (0, \infty)$. This implies also that $\{\boldsymbol{x} \in \mathbb{R}^N : B^T\boldsymbol{x} = \boldsymbol{0}\} = \emptyset$
- As can be seen from the above it is not advisable to use quadrature rules with negative weights because they may lead to the loss of the spectral equivalence between BB^T and Λ.

The following theorem relates the spectrum of H to the spectrum of A if A is a positive (or negative) definite matrix. As can be seen the eigenvalues of M are easy to determine.

Theorem 7.5.2 *Assume that M is defined as in (7.3.5) and define matrix $M^{(m)}$ by*

$$M^{(m)} = (w_k^{(l)})^{-1} \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} (N_k^{(l)})$$

for some index $m = i_k^{(l)}$. Then $\sigma(M) = \cup_{m=1}^{N'} \sigma(M^{(m)})$.
If all weights $w_k^{(l)}$ are positive then $\sigma(M) \subset (0, \infty) \Rightarrow \sigma(H) \subset (0, \infty)$.
Proof. Note that there exists a permutation matrix P such that PMP^{-1} is a block diagonal matrix with n by n blocks $M^{(m)}$ as defined above. Bearing this in mind it is clear that $\sigma(M) = \cup_{m=1}^{N'} \sigma(M^{(m)})$.

Now suppose all weights are positive and M is positive definite. Use of the fact that $H = BMB^T$ and that $\{\boldsymbol{x} \in \mathbb{R}^N : B^T\boldsymbol{x} = \boldsymbol{0}\} = \emptyset$

yields

$$(Hx,x) = (BMB^Tx,x) = (MB^Tx,B^Tx) > 0.$$

whence H is also positive definite. □

Remarks 7.5.2

- For the last part of the proof above it is sufficient to assume that M is positive definite on the set $\{B^Tx : x \in \mathbb{R}^N\}$
- The eigenvalues of M are the eigenvalues of A at the quadrature points $N_k^{(l)}$ scaled with $(w_k^{(l)})^{-1}$
- If the spectrum $\sigma(M^{(m)})$ contains a negative element or a zero element for just one $m \in \{1,\ldots,N'\}$ then stiffness matrix H can be indefinite.

Let H and G be two symmetric stiffness matrices and consider the following estimate for the condition number of $G^{-1}H$.

Theorem 7.5.3 *Define the stiffness matrices H and G by*

$$[H]_{ij} = Q((A\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i), \text{ and } [G]_{ij} = Q((Z\underline{\nabla}\varphi_j)^T\underline{\nabla}\varphi_i)$$

where the positive definite symmetric matrices A and Z are given by $A \equiv [a_{p,q}]$, $Z \equiv [z_{p,q}]$ with $a_{p,q}$, $z_{p,q}$ given functions on Ω, and assume that Q has positive weights.

Suppose G is factored into LL^T with L a lower triangular matrix and let λ_1, λ_n be the minimum respectively maximum eigenvalue of the complete spectrum $\sigma = \{\lambda_1,\ldots,\lambda_n\} \subset (0,\infty)$ of the generalized eigenvalue problem $Ax = \lambda Zx$.
Then the condition number $\mathrm{Cond}(L^{-1}HL^{-T})$ *satisfies*

$$\mathrm{Cond}(L^{-1}HL^{-T}) \leq \frac{\lambda_n}{\lambda_1} \tag{7.5.2}$$

independent of the choice of the basis functions $\{\varphi_i\}_{i=1}^N$.

Proof. Because A and Z are symmetric positive definite there exist a complete set of eigenvectors and a spectrum σ as above. Let for $u = [\alpha_1,\ldots,\alpha_N]^T$ the function u be defined by $u = \sum_{i=1}^N \alpha_i\varphi_i$, then $u^THu = Q((A\underline{\nabla}u)^T\underline{\nabla}u)$. From standard algebra it is known that if A satisfies the assumptions above then $\lambda_1 x^TZx \leq x^TAx \leq$

$\lambda_n \boldsymbol{x}^T Z \boldsymbol{x}$ for all $\boldsymbol{x} \in \mathbb{R}^N$. Because Q is monotone and linear this yields $\lambda_1 Q((Z\underline{\nabla} u)^T \underline{\nabla} u) \leq Q((A\underline{\nabla} u)^T \underline{\nabla} u) \leq \lambda_n Q((Z\underline{\nabla} u)^T \underline{\nabla} u)$ on Ω. From this it follows directly that $\text{Cond}(L^{-1} H L^{-T}) \leq \lambda_n / \lambda_1$. □

Remarks 7.5.3

- For $Z \equiv I_n$ this leads to ($LL^T = \Lambda$) $\text{Cond}(L^{-1} H L^{-T}) \leq \lambda_n(A)/\lambda_1(A)$ where $0 < \lambda_1(A) \leq \lambda_n(A)$ are the extreme eigenvalues of A
- If furthermore $A \equiv C$ as defined in theorem 7.5.1 then the condition number satisfies $\text{Cond}(L^{-1} B B^T L^{-T}) = w_{\max}/w_{\min}$.

7.6 Advantages of the factorization

Some advantages of the use of the factorization method for the assembly of stiffness matrices compared to the use of straightforward quadrature methods are:

- The matrix B^T, which is a discretization of the gradient operator, depends only on the grid geometry. Furthermore the matrix M is a discretization of matrix A, so during the computation of M one gets information about A. If matrix $H \equiv H(u)$ is non-linear then only M and not B will depend on this function u. Hence updating H only involves updating M
- According to theorem 7.3.1 the degree of the local basis functions and quadrature formulas may vary from element to element. Hence it is possible to use quadrature rules for elements lying near boundaries or internal layers which differ from those used for other parts of the domain
- For some specific choices of $\{\varphi_r^{(l)}\}_{r=1}^{T_l}$ and $\{\psi_k^{(l)}\}_{k=1}^{q_l}$ the element matrix B_l will be very simple, thereby considerably reducing the amount of work to be performed
- If M is invertible, then H^{-1} can be approximated by

$$(BB^T)^{-1} B M^{-1} B^T (BB^T)^{-1} .$$

This approximation is especially accurate when M is close to a constant times the identity matrix, see e.g. [4] and [3]. Some of the advantages are:

- for an arbitrary grid BB^T is nonsingular and spectrally equivalent to the stiffness matrix Λ on Ω if all weights are positive. Matrices Λ and BB^T have the same sparsity structure as H which means that no new pointer arrays have to be generated
- Matrix M^{-1} is easy to compute due its simple structure

- When solving linear systems with matrix BMB^T by iteration, the residuals can be computed in a stable way using differences, which implies that the resulting round-off errors in the computed solutions are bounded independent of the fineness of the grid. In the present formulation, the matrix vector multiplication with B and B^T are then performed as sum of differences which means that they can be computed with no or small roundoff errors (see [5]). Without this trick these errors increase as $O(N)$, $N \to \infty$, in a two-dimensional problem.

7.7 The Navier's system of equations

In this section the Navier's equations, which describe the deformation of solids or the flow of fluids, illustrate an extension of theorem 7.3.1 to systems of differential equations. Theorem 7.5.3 will be used to obtain condition numbers of some preconditioned versions.

In two space dimensions the *Navier's equations* can be written as the following linear system of differential equations:

$$\begin{aligned} -\frac{\partial}{\partial x^2}u_1 - \frac{1-\tilde{\nu}}{2}\frac{\partial}{\partial y^2}u_1 - \frac{1+\tilde{\nu}}{2}\frac{\partial}{\partial xy}u_2 &= f_1 \text{ on } \Omega \\ -\frac{1-\tilde{\nu}}{2}\frac{\partial}{\partial x^2}u_2 - \frac{\partial}{\partial y^2}u_2 - \frac{1+\tilde{\nu}}{2}\frac{\partial}{\partial xy}u_1 &= f_2 \text{ on } \Omega \\ \boldsymbol{u} &= 0 \text{ on } \Gamma \end{aligned} \tag{7.7.1}$$

where $\tilde{\nu} = \nu/(1-\nu)$ with ν the Poisson ratio (see for instance [7] and [20]) and $\boldsymbol{u} = [u_1, u_2]^T$. The scalar $\tilde{\nu} \in (0,1)$ is related to the properties of the material used. For almost inelastic (incompressible)

material $\tilde{\nu} \approx 1$. The variational formulation of (7.7.1) is: find $\boldsymbol{u} = [u_1, u_2] \in V_1 \otimes V_2$ such that

$$\int_\Omega (A\underline{\nabla}\boldsymbol{u})^T \underline{\nabla}\boldsymbol{v}\, d\boldsymbol{x} = 0 \quad \forall_{v \in V_1 \otimes V_2} \tag{7.7.2}$$

where $V_1, V_2 \subset \mathrm{H}_0^1(\Omega)$ are simultaneously the trial and testspaces and A is a symmetric but not diagonal matrix

$$\begin{bmatrix} a & 0 & 0 & c \\ 0 & b & 0 & 0 \\ 0 & 0 & b & 0 \\ c & 0 & 0 & a \end{bmatrix}$$

with $a = 1$, $b = (1 - \tilde{\nu})/2$ and $c = (1 + \tilde{\nu})/2$. Note that A is singular for $\tilde{\nu} = 1$.

Now discretize (7.7.2) by chosing $V_1 = V_2 = V$, the set spanned by the finite element basis functions $\{\varphi_i\}_{i=1}^N$, and a quadrature rule Q with positive weights. Define $\{w_i\}_{i=1}^{2N}$ to be a basis for $V \otimes V$ and the $2N$ by $2N$ stiffness matrix H by $[H]_{ij} = Q((A\underline{\nabla}w_j)^T \underline{\nabla}w_i)$. Then, in an analogous way to section 7.3, it can be shown that H can be factorized into a form BMB^T where B is an N by $4N'$ matrix and M is a $4N'$ by $4N'$ matrix

$$H = BMB^T, B = \left[B^{(1)}B^{(2)}B^{(1)}B^{(2)}\right]$$

$$M = \begin{bmatrix} M^{(1,1)} & 0 & 0 & M^{(1,4)} \\ 0 & M^{(2,2)} & 0 & 0 \\ 0 & 0 & M^{(2,2)} & 0 \\ M^{(1,4)} & 0 & 0 & M^{(1,1)} \end{bmatrix}$$

with $B^{(1)}, B^{(2)}, M^{(1,1)}, M^{(2,2)}$ and $M^{(1,4)}$ defined as in (7.3.4) respectively (7.3.5).

Finally, an estimate of the condition number of H, preconditioned by two particular stiffness matrices G, is presented. To this end define the

$$A(\theta) = \begin{bmatrix} a & 0 & 0 & \theta c \\ 0 & b & (1-\theta)c & 0 \\ 0 & (1-\theta)c & b & 0 \\ \theta c & 0 & 0 & a \end{bmatrix}$$

$$D = \mathrm{Diag}(a, b, b, a)$$

where $0 \leq \theta \leq 1$ and note that $A \equiv A(1)$.

Theorem 7.7.1 *Let $A, A(\theta)$, and D be defined as above. Then*

- *for $[H]_{ij} = Q((A\underline{\nabla}w_j)^T\underline{\nabla}w_i)$ and $[G]_{ij} = Q((I_4\underline{\nabla}w_j)^T\underline{\nabla}w_i) = LL^T$ one has*

$$\mathrm{Cond}(L^{-1}HL^{-T}) \leq \frac{3+\tilde{\nu}}{1-\tilde{\nu}}$$

- *for $[H]_{ij} = Q((A\underline{\nabla}w_j)^T\underline{\nabla}w_i)$ and $[G]_{ij} = Q((D\underline{\nabla}w_j)^T\underline{\nabla}w_i) = LL^T$ one has*

$$\mathrm{Cond}(L^{-1}HL^{-T}) \leq \frac{2}{1-\tilde{\nu}}.$$

Proof. First note that matrix A in (7.7.2) can be replaced by matrix $A(\theta)$ for all $\theta \in [0,1]$ without changing equation (7.7.2) because the homogeneous boundary conditions imply

$$\int_\Omega (\frac{\partial}{\partial xy}v)u\,d\boldsymbol{x} = \theta\int_\Omega \frac{\partial}{\partial x}v\frac{\partial}{\partial y}u\,d\boldsymbol{x} + (1-\theta)\int_\Omega \frac{\partial}{\partial y}v\frac{\partial}{\partial x}u\,d\boldsymbol{x} \quad \forall_{\theta\in[0,1]}$$

for all $u, v \in \mathrm{H}_0^1(\Omega)$. Now use theorem 7.5.3 to determine the condition number:

- Let matrix $Z = I_4$, then $G \equiv \Lambda = LL^T$. Consider the eigenvalue problem $A\boldsymbol{x} = \lambda I_4\boldsymbol{x}$. In this case it is easy to see that $\sigma = \{(1-\tilde{\nu})/2, (3+\tilde{\nu})/2\}$. According to theorem 7.5.3 hence $\mathrm{Cond}(L^{-1}HL^{-T}) \leq (3+\tilde{\nu})/(1-\tilde{\nu})$
- Now take matrix $Z = D$ and let $A = A(\theta)$. The solution of the generalized eigenvalue problem $A(\theta)\boldsymbol{x} = \lambda(\theta)D\boldsymbol{x}$ now yields $\sigma := \{a \pm \theta c, a \pm (1-\theta)c/b\}$. Minimization of $\lambda_{\max}(\theta)/\lambda_{\min}(\theta)$ for $\theta \in [0,1]$ yields, according to [7], $\theta = 2/(3-\tilde{\nu})$ whence $\mathrm{Cond}(L^{-1}HL^{-T}) \leq 2/(1-\tilde{\nu})$.

□

As can be seen it is possible to extend the theory of section 7.3 for systems of partial differential equations without any restrictions. In this case it is advisable to assemble H elementwise using the submatrices $B^{(p)}M^{(p,q)}B^{(q)^T}$ as in section 7.4.

7.8 Stiffness matrices and mixed variational formulation

Earlier methods (see [3] and [6]) to derive the BMB^T factorization of stiffness matrices used a mixed variational formulation of the system of coupled equations related to (7.2.4):

$$\begin{aligned} A^{-1}z - \underline{\nabla} u &= 0 \text{ on } \Omega \\ -\nabla \cdot z &= f \text{ on } \Omega \\ u &= 0 \text{ on } \Gamma_D \\ z^T n &= g \text{ on } \Gamma_N \end{aligned} \tag{7.8.1}$$

with $z = A\underline{\nabla} u$, A an n by n nonsingular matrix and n the outer normal of Ω.

Now consider a discretized mixed variational formulation of (7.8.1). Choosing as before a finite element space $V \subset H_0^1(\Omega)$, a quadrature rule Q as in (7.2.1) and a corresponding finite element space $M \subset L^2(\Omega)$ constructed as in section 7.3, this gives: find $u_h \in V$ and $z_h \in M^n$ such that

$$\begin{aligned} &Q((A^{-1}z_h)^T w) - Q(\underline{\nabla} u_h^T w) = 0 \quad \forall_{w \in M^n} \\ &-Q(z_h^T \underline{\nabla} v) = \int_\Omega f v \, dx + \oint_{\Gamma_N} g v \, ds \quad \forall_{v \in V} \end{aligned} \tag{7.8.2}$$

where h is the finite element grid parameter. If one takes

$$((\psi_1, 0, \ldots, 0), ., (\psi_{N'}, 0, \ldots, 0), (0, \psi_1, 0, \ldots, 0), ., (0, \ldots, 0, \psi_{N'}))$$

as a basis for M^n, and sets

$$\begin{cases} u_h = \sum_{i=1}^{N} \alpha_{i,h} \varphi_i & \alpha = [\alpha_1, \ldots, \alpha_N]^T \\ z_h = \sum_{i=1}^{n} [\beta_{i,h}^{(1)}, \ldots, \beta_{i,h}^{(N')}]^T \psi_i & \beta = [\beta_{i,h}^{(1)}, \ldots, \beta_{i,h}^{(N')}]^T, \end{cases}$$

then this leads to the system of equations

$$\begin{bmatrix} M^{-1} & -B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \beta \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ F \end{bmatrix}$$

or equivalently

$$BMB^T\alpha = F$$

where the block matrices M and B are defined as in theorem 7.3.1.

Note that the representation of matrix M depends on the order of the basis functions of M^n. If the basis for M^n above is permuted into

$$((\psi_1, 0, \ldots, 0), \cdots, (0, \ldots, 0, \psi_1), (\psi_{N'}, 0, \ldots, 0), \cdots, (0, \ldots, 0, \psi_{N'}))$$

then this leads to a block diagonal matrix $M = \text{Diag}(M^{(1)}, \ldots, M^{(N')})$ of order nN' with diagonal blocks $M^{(m)}$ of order n as defined in theorem 7.5.2. The block structure of M for both bases is shown in figure 7.6.

For nonsingular matrices A the factorization (theorem 7.3.1) and the mixed formulation coincide. This has already been proved for a positive definite diagonal matrix A in [3], [6] for the choice of:

- quadratic basis functions together with 3-point Gaussian quadrature on a triangulation of a two-dimensional domain
- linear basis functions in combination with 1-point Gaussian quadrature on a similar domain.

7.9 Numerical results

In this section a non-linear differential equation is studied to compare the computational work for the residual (gradient) and the Hessian with the use of the factorization method. Further the computational complexity for the various parts of a damped inexact Newton solution algorithm is examined to provide an estimate of the assembly costs for the Hessian in comparison with the total computational costs.

As a test problem consider the following non-linear differential equation which models a *subsonic* potential flow around an airplane wing (see fig. 7.7)

$$\begin{aligned} &-\nabla\cdot(\rho(|\underline{\nabla} u|^2)\underline{\nabla} u) = 0 \text{ on } \Omega \subset \mathbb{R}^2 \\ &\rho(|\underline{\nabla} u|^2) = \rho_\infty(1 - \tfrac{1}{5}(|\underline{\nabla} u|^2 - v_\infty^2))^{5/2} \end{aligned} \qquad (7.9.1)$$

with boundary conditions

$$\begin{aligned} \rho(|\underline{\nabla} u|^2)\underline{\nabla} u^T \boldsymbol{n} = &\quad 0 \quad && \text{on the wing} \\ u(\boldsymbol{x}) = &\quad v_\infty \cdot x && \text{at infinity.} \end{aligned}$$

Here ρ_∞ is the air density at infinity and $v_\infty = M_\infty$ is the wind velocity at infinity (see e.g. [9]), with M_∞ being the Mach number at infinity. For other types of domain, like windtunnel sections, see [15]. Note that Γ_N is the wing profile and that Γ_D is a circle around the airplane wing, at a suitable distance.

If this equation is linearized using a Newton method and if the resulting linear systems are discretized with the use of quadratic finite element basis functions $\{\varphi_i\}_{i=1}^N$ on triangles in combination with 3-point Gaussian quadrature, then the gradient $F(u)$ and its derivative, the Hessian $F'(u)$, are given by (see [8] and [9])

$$\begin{aligned} [F(u)]_i =& Q(\rho(|\underline{\nabla} u|^2)\underline{\nabla} u^T \underline{\nabla}\varphi_i) - \oint_{\Gamma_N} (\rho(|\underline{\nabla} u|^2)\underline{\nabla} u^T \boldsymbol{n})\varphi_i ds \\ [F'(u)]_{ij} =& Q(\rho(|\underline{\nabla} u|^2)\underline{\nabla}\varphi_j{}^T \underline{\nabla}\varphi_i + \\ & 2 \cdot \rho'(|\underline{\nabla} u|^2)\underline{\nabla} u^T \underline{\nabla}\varphi_j \underline{\nabla} u^T \underline{\nabla}\varphi_i) \end{aligned}$$

where $\rho'(\zeta) \equiv \frac{\mathrm{d}}{\mathrm{d}\zeta}\rho(\zeta)$, $\zeta = |\underline{\nabla} u|^2$. Note that for $\||\underline{\nabla} u|^2\|_{\infty,\Omega} < (5 + v_\infty^2)/6$ the Hessian is positive definite.

According to theorem 7.2.1 F' is a stiffness matrix. Rewriting F and F' into stiffness matrices yields

$$\begin{aligned} F(u) &= G(u)\boldsymbol{u} - \boldsymbol{f} \\ [G(u)]_{ij} &= Q((A\underline{\nabla}\varphi_j)^T \underline{\nabla}\varphi_i) \\ A &= \mathrm{Diag}(\rho(|\underline{\nabla} u|^2), \rho(|\underline{\nabla} u|^2)) \\ [\boldsymbol{f}]_i &= \oint_{\Gamma_N} (\rho(|\underline{\nabla} u|^2)\underline{\nabla} u^T \boldsymbol{n})\varphi_i ds \\ [F'(u)]_{ij} &= Q((\tilde{A}\underline{\nabla}\varphi_j)^T \underline{\nabla}\varphi_i) \end{aligned}$$

where

$$\tilde{A} = \begin{bmatrix} \rho(|\underline{\nabla} u|^2) + 2 \cdot u_x^2 \cdot \rho'(|\underline{\nabla} u|^2) & 2 \cdot u_x \cdot u_y \cdot \rho'(|\underline{\nabla} u|^2) \\ 2 \cdot u_x \cdot u_y \cdot \rho'(|\underline{\nabla} u|^2) & \rho(|\underline{\nabla} u|^2) + 2 \cdot u_y^2 \cdot \rho'(|\underline{\nabla} u|^2) \end{bmatrix}.$$

Note that G and F' are both stiffness matrices with bandwidth 19.

In order to simplify the comparison of the assemblage costs for the gradient and Hessian, assume that the evaluation of A at one quadrature point costs $d = 15$ flops and that the evaluation of $\tilde{A}$ costs $2d$ flops (evaluation of $\rho(|\underline{\nabla} u|^2)$ and $\rho'(|\underline{\nabla} u|^2)$). Further, for this choice of basis functions, the asymptotic relationship $2L = N$ will be used, which is true in the limit case where the grid parameter $h \to 0$. Therefore the additional matrix vector multiplication needed for the assembly of the gradient costs $19 \cdot 2L = 38L$ flops.

Now comparing the costs of the gradient assembly with (7.4.6) and (7.4.7) gives

$$\begin{cases} (21 \cdot \{2 + 3 \cdot 2 \cdot 2\} + 3d + 204)L & = 753L \quad \text{for } F \text{ with (4.7)} \\ (2 \cdot 27 + 3d + 204)L + 38L & = 341L \quad \text{for } F \text{ with (4.6)} \end{cases}$$

whence leading to a ratio of (4.6) : (4.7) = 1 : 2.2.

If the factorization method for the element matrix assembly for F as well as for F' is used, then this yields

$$\begin{cases} (2 \cdot 27 + 3d + 204)L + 38L & = 341L \quad \text{for } F \text{ with (4.6)} \\ (2 \cdot 27 + 45 + 6d + 204)L & = 393L \quad \text{for } F' \text{ with (4.6)} \end{cases}$$

leading to a ratio of $F : F' = 1 : 1.2$, which is fairly good compared to [8], where the ratio of assembly costs is about $F : F' = 1 : 5$.

Assembling Hessian and gradient simultaneously will reduce the overhead costs (except for the evaluation of ρ') hence further reducing the assembly costs for one Newton step:

$$\begin{cases} 341L + 393L & = 734L \quad \text{for } F \text{ and } F', \text{(4.6) apart} \\ 341L + (2 \cdot 27 + 45 + 3d)L & = 485L \quad \text{for } F \text{ and } F' \text{ simultaneous.} \end{cases}$$

This means that the costs for the calculation of the gradient assembly are only multiplied by a factor 1.4. Therefore one may conclude that the calculation of the Hessian costs almost nothing.

Table 7.9.1 Assembly costs.

task	#flops
(1) assembly F	$170N$
(2) assembly F' additional	$72N$
(3) factorization F'	$50N$
(4) one step ILU-CG	$43N$

Now consider the total costs for each damped inexact Newton step, with ILU-CG to solve the linearized systems (for the sparsity pattern of the Jacobian matrix, see fig. 7.8). The costs can be divided into those for the assembly of F and F', the ILU factorization of F' and for the solution of $F'x = F$ with the ILU-CG. Table 7.9.1 (see also [5], pp.384,385) shows the costs for the separate parts for the simultaneous assembly of F and F'.

Because one Newton iteration step consists of several ILU-CG iteration steps, one roughly obtains the approximate ratios

$$(1) : (2) : (3) : (4) = 4 : 2 : 1 : i$$

where i is the average number of PCG iterations per Newton step (note that backtracking can be neglected if one uses the *continuous method* in [2]). For an almost *transonic* problem with $v_\infty = 0.748$, using several coarse grids, obtained by uniform refinement from a coarse grid $Q^{(0)}$ as in chapter 5, this average number i is shown in table 7.9.2.

Table 7.9.2 # Newton steps.

Grid	N	#Newton steps	i
$Q^{(0)}$	561	4: 8,14,17,22	16
$Q^{(1)}$	1073	4: 3,37,58,79	45
$Q^{(2)}$	2145	3: 6,70,96	58
$Q^{(3)}$	4193	3: 10,117,149	92

Here the stopping criterions (see sections 5.9 and 8.2) are given by $\varepsilon_{\text{nonlinear}} = 10^{-10}$ and $\varepsilon_{\text{linear}}$, such that one has quadratic convergence (see section 8.2). On all grids $Q^{(k)}$, the starting residual $F(u_I^{(k)})$ of the interpolant of the previous solution is in the order of 10^{-1}. On the first grid, $u_I^{(k)}(x, y) = v_\infty^2 \cdot x$ is taken as a starting solution. See figs. 7.10, 7.12 resp. 7.9, 7.11 for the finite element grid and isoclines of the Mach number on the domain of the solution of (7.9.1) for $N = 4193$ on grid $Q^{(3)}$. In the third column of table 7.9.2, after the semicolon, the number of linear iterations for the separate Newton steps are listed.

Note in passing that there is no difficulty solving problems with Mach numbers smaller than but arbitrary close to 1 (subsonic case). Newtons method diverges however in the transonic case.

Note that it costs nothing to compute the eigenvalues of A and $\tilde{A}$ during the assembly of G and F' in this test example because

$$\sigma(A) = \{\rho(|\underline{\nabla} u|^2)\}$$
$$\sigma(\tilde{A}) = \{\rho(|\underline{\nabla} u|^2), \rho(|\underline{\nabla} u|^2) + 2 \cdot |\underline{\nabla} u|^2 \cdot \rho'(|\underline{\nabla} u|^2)\}\,.$$

This is of practical importance because if the smallest eigenvalue of $\tilde{A}$ becomes negative then the Hessian will eventually become indefinite if the grid will be refined up to a high level. This indefiniteness can cause the ILU-CG to crash and make the damped inexact Newton method useless.

Further, since the Hessian is not uniformly positive definite, note that a breakdown could occur if one of the subsequent iterands would become transonic due to a too large stepsize, determined by $\tau^{(k)}$. However, for the numerical test presented above where the solution is nearly transonic, but still subsonic, it is fairly easy to prevent the damped inexact Newton method from breaking down. In order to see this, note that the second eigenvalue in $\sigma(\tilde{A})$ is precisely equal to $\rho(x, |\underline{\nabla} u|^2)(1 - M^2(x, |\underline{\nabla} u|^2)$, where

$$M^2(x, |\underline{\nabla} u|^2) = -2|\underline{\nabla} u|^2 \rho'(x, |\underline{\nabla} u|^2)/\rho(x, |\underline{\nabla} u|^2)\,. \qquad (7.9.2)$$

Here, in computational fluid dynamics, the term M^2 is called the Mach number (squared). If this Mach number is uniformly less than one then the problem is said to be of subsonic nature – the problem is elliptic –, if it is locally greater than or equal to one, the problem becomes *transonic* – a parabolic layer arises – and if the Mach number is uniformly greater than one than the problem is said to be *supersonic* or hyperbolic. Before introducing the modification of the diffusion tensor to ensure that the Newton algorithm can not break down, consider the Mach number for some well-known examples.

Example 7.9.1 Consider the case where $\rho(|\underline{\nabla}_x u|^2) = f(x) \cdot (|\underline{\nabla}_x u|^2)^p$. Then

$$M^2(|\underline{\nabla}_x u|^2) = -2|\underline{\nabla}_x u|^2 \cdot \frac{f(x) \cdot p(|\underline{\nabla}_x u|^2)^{p-1}}{f(x) \cdot (|\underline{\nabla}_x u|^2)^p} = -2p \quad \forall_{p \in \mathbf{R}}$$

whence the Mach number squared is uniformly less than 1 for all $p > -\frac{1}{2}$. Note that the Ladyzhenskaya model (1.2.2) has a diffusion tensor of this type, possibly scaled by a constant.

Example 7.9.2 Consider the *minimal surface equation* tensor where

$$\rho(|\underline{\nabla}_x u|^2) = (1 + |\underline{\nabla}_x u|^2)^{-1/2}.$$

In this case the Mach number is equal to

$$M^2(|\underline{\nabla}_x u|^2) = -2|\underline{\nabla}_x u|^2 \cdot \frac{-\frac{1}{2}(1 + |\underline{\nabla}_x u|^2)^{-3/2}}{(1 + |\underline{\nabla}_x u|^2)^{-1/2}} = \frac{|\underline{\nabla}_x u|^2}{1 + |\underline{\nabla}_x u|^2} < 1$$

for all possible solutions u. Note that the Mach number in this case is not uniformly bounded below 1, and that equation (1.2.3) admits such a diffusion tensor.

Finally, the modification of the diffusion tensor ρ is investigated. The smallest eigenvalue of $\tilde{A}$, $\rho(\zeta) + 2\zeta\rho'(\zeta)$, is clearly greater than or equal to 0 if $\rho(\zeta) > 0$ and $\rho(\zeta) + 2\zeta\rho'(\zeta) \geq 0$. Now consider the positive function b: $[0, \infty) \mapsto [0, \infty)$ defined by

$$b(\zeta) = c_0(\zeta + c_1)^{-1/2}, \quad c_0, c_1 \in (0, \infty).$$

It is easily verified that $b(\zeta) + 2\zeta b'(\zeta) = b(\zeta) \cdot c_1/(\zeta + c_1) > 0$ for all $c_0, c_1 \in (0, \infty)$. However, this estimate is not uniform in ζ.

Now a breakpoint ζ_0 is chosen such that $M^2(\zeta_0) = 1 - \varepsilon$ for given $\varepsilon \in (0, 1)$ leading to the unique choice

$$\zeta_0 = (5 + v_\infty^2) \cdot \frac{1 - \varepsilon}{6 - \varepsilon}$$

where $\zeta_0 > v_\infty^2 \Leftrightarrow \varepsilon > 1 - v_\infty^2$. Then scalars c_0 and c_1 are chosen such that the graphs of ρ and b fit continuously differentiable at the breakpoint ζ_0. In order to determine these scalars note that

$$\begin{array}{llll} \rho(\zeta) & = g^p(\zeta) & b(\zeta) & = c_0 f^q(\zeta) \\ \rho'(\zeta) & = -\frac{1}{2} g^{p-1}(\zeta) & b'(\zeta) & = -\frac{1}{2} c_0 f^{q-1}(\zeta) \end{array}$$

where $g(\zeta) = 1 - \tilde{\gamma}(\zeta - v_\infty^2)$, $f(\zeta) = \zeta + c_1$, and $p = 1/(2\tilde{\gamma}) = \frac{5}{2}$ resp. $q = -\frac{1}{2}$. The continuously differentiability condition leads to the conditions

$$\rho(\zeta_0) = b(\zeta_0) \wedge \rho'(\zeta_0) = b'(\zeta_0)$$

and therefore to the conditions

$$\begin{aligned} g(\zeta_0) = f(\zeta_0) \quad &\Rightarrow c_1 = g(\zeta_0) - \zeta_0 = \frac{(5 + v_\infty^2)\varepsilon}{6 - \varepsilon} \\ c_0 f(\zeta_0)^q = g(\zeta_0)^p = f(\zeta_0)^p \quad &\Rightarrow c_0 = f^{p-q}(\zeta_0) = \left(\frac{5 + v_\infty^2}{6 - \varepsilon}\right)^3 \end{aligned}$$

Finally, this leads to a modified tensor $\hat{\rho}$ defined by

$$\begin{aligned} \hat{\rho}(\zeta) &= \rho(\zeta) \quad \text{for } \zeta \in [0, \zeta_0) \\ \hat{\rho}(\zeta) &= b(\zeta) \quad \text{for } \zeta \in [\zeta_0, \infty). \end{aligned}$$

Using the tensor $\hat{\rho}$, a solution with Mach number uniformly less than $1 - \varepsilon$ will be a solution of the original problem with diffusion ρ and a breakdown can not occur. As the modification is easy to compute, it is an easy way to avoid a breakdown of the Newton solution algorithm.

In the airplain wing example ε is chosen to be 0. 005, whence the breakpoint $\zeta_0 = 0.9228$. On all grids, the computed solutions remained below this threshold, whence every solution is a subsonic solution of equation (7.9.1), without diffusion tensor modification.

7.10 Conclusions

It has been demonstrated that for many practically important examples like the electromagnetic field equations or the potential equations for aerodynamics the assembly of the Hessian matrix which occurs in a damped Newton algorithm is not more expensive than the assembly of the gradient, if the factorization method presented in this chapter is used. Hence in those cases there is no need to avoid updating the Hessian each iteration step or to use other techniques to assemble the Hessian such as those described in the introduction.

It can even be stated that the computation of the Hessian costs almost nothing except for the calculation of the derivative of a function

as the residual has to be calculated at each iteration step in any case. The extra work needed for the Hessian is often so small that, when the gradient has already been computed, it amounts to about the computing time needed to store its entries.

Use of the factorization method provides also good information about positive definiteness of the stiffness matrix. Also more insight in the effect of the use of quadrature formulas in finite element methods is provided.

7.11 References

[1] Argyris J.H. and Brønlund O.E., *The natural factor formulation of the stiffness for the matrix displacement method,* Computer Methods in Applied Mechanics and Engineering, 5(1975), 97-119

[2] Axelsson O., *On global convergence of iterative methods,* **in** Iterative Solution of Nonlinear Systems of Equations, 1-19 LNM#953, (Ansorge R., Meis Th. and Törnig W. eds.), Springer Verlag, 1982

[3] Axelsson O., *A mixed variable finite element method for the efficient solution of nonlinear diffusion and potential flow equations,* **in** Advances in Multi-grid Methods (Braess D. et al eds.), 1-11, Vieweg Verlag, Braunschweig - Wiesbaden, 1985

[4] Axelsson O., *Numerical algorithms for indefinite problems,* **in** Elliptic Problem Solvers II, (Birkhoff G. and Schoensstadt A. eds.), Academic Press, 1984

[5] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[6] Axelsson O. and Gustafsson I., *An efficient finite element method for nonlinear diffusion problems,* Report 84.06R, ISSN0347-0946

[7] Axelsson O. and Gustafsson I., *Iterative methods for the solution of the Navier's equations of elasticity,* Computer Methods in Applied Mechanics and Engineering, 15(1978), 241-258

[8] Axelsson O. and Nävert U., *On a graphical package for nonlinear partial differential equation problems,* **in** Information Processing 77, IFIP, 103-108, North-Holland , 1977

[9] Bristeau M.O., Glowinski R., Periaux J., Perrier P. and Pironneau O., *On the numerical solution of nonlinear problems in fluid*

dynamics by least square and finite element methods. (I) Least squares formulations and conjugate gradient solutions of the continuous problems, Computer Methods in Applied Mechanics and Engineering, 17/18(1979), 619-657. *(II) Application to transonic flow simulations*, Computer Methods in Applied Mechanics and Engineering, 51(1985), 363-394

[10] Broyden C.G., *The convergence of an algorithm for solving sparse nonlinear systems,* Mathematics of Computation, 25(1971), 285-294

[11] Dembo R.S., Eisenstat S.C. and Steihaug T., *Inexact Newton methods,* SIAM Journal on Numerical Analysis, 19(1982), 400-408

[12] Duvenant, *High degree efficient symmetric gauss quadrature rules for the triangle,* International Journal for Numerical Methods in Engineering, 21(1985), 1129-1148

[13] Haegemans A. and Piessens R., *Cubature formulas of degree 7 for symmetric planar regions,* Journal of Computational and Applied Mathematics, 1(1975), 79-83

[14] Hemker P.W. and Spekreijse S.P., *Multiple grid and Osher's scheme for the efficient solution of the steady Euler equations,* Applications in Numerical Mathematics, 2(1986), 475-493

[15] Koren B., *Euler flow solutions for a transonic windtunnel section,* CWI report NM-R8601, Amsterdam, 1986

[16] Kron G., *Solution of complex nonlinear plastic structures by the method of tearing,* Journal of Aerodynamic Science 23(1956), no. 6

[17] Kron G., *Tensor Analysis and Networks,* Wiley, New York, 1939

[18] Langefors B., *Algebraic methods for the numerical analysis of built-up systems,* SAAB TN 38 (1957), SAAB Aircraft Company, Linköping, Sweden

[19] Mizukami A., *Some integration formulas for a four-node isoparametric element,* Computer Methods in Applied Mechanics and Engineering, 59(1986), 111-121

[20] Nečas J. and Hlaváček L., *Mathematical Theory of Elastic and Elastico-plastic Bodies: an Introduction,* Elsevier (Studies in Applied Mechanics 3), Amsterdam, 1981

[21] Periaux J., *Three-dimensional analysis of compressible potential*

flows with the finite element method, International Journal for Numerical Methods in Engineering, 9(1975), 775-831

[22] Strang G., *Introduction to Applied Mathmatics,* Wellesley Cambridge Press, Wellesley, Mass., 1986

[23] Strang G. and Fix G.S., *An analysis of the finite element method,* Prentice Hall, Englewood Heights, New Jersey, 1973

[24] Stroud A.H., *Approximate Calculation of Multiple Integrals,* Prentice-Hall (Series in Automatic Computation), NewYork, 1971

[25] Zienkiewicz O., *The Finite Element Method in Engineering Science,* 3^{rd} edition, Mc Graw-Hill, New York, 1977

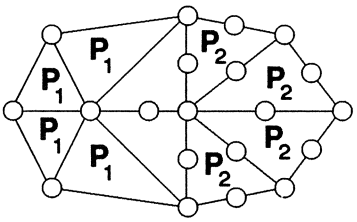


Fig. 7.1 An example of a piecewise polynomial basis.

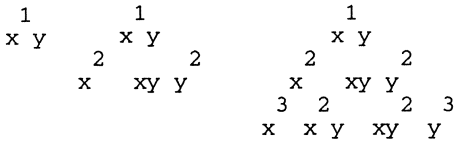


Fig. 7.2 Basis function spans.

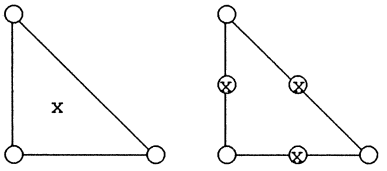


Fig. 7.3 Quadrature rules on a triangle.

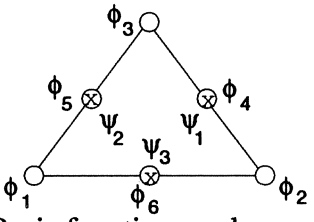


Fig. 7.4 Basis functions and quadrature points.

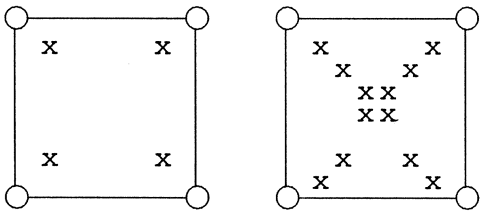


Fig. 7.5 Quadrature rules on a square.

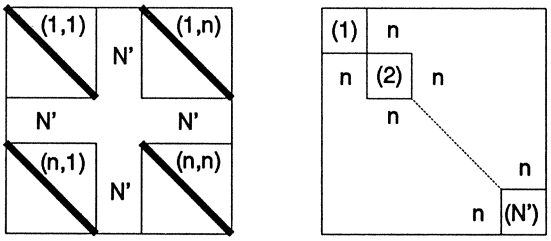


Fig. 7.6 Matrix M on first and on second basis.

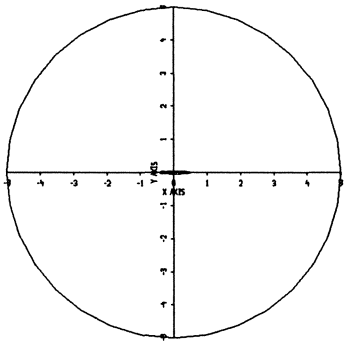


Fig. 7.7 The computational domain.

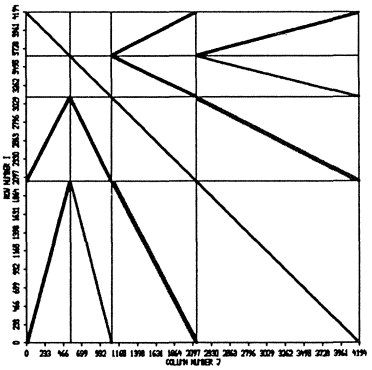


Fig. 7.8 Standard nodal sp. pattern.

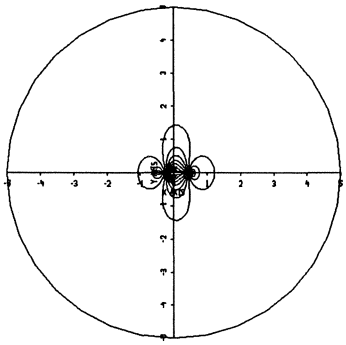


Fig. 7.9 Isoclines of the solution on $Q^{(3)}$.

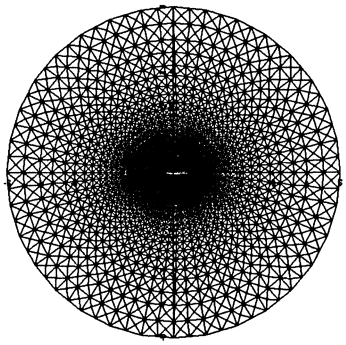


Fig. 7.10 The refined grid $Q^{(3)}$.

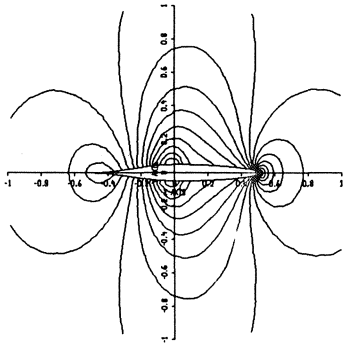


Fig. 7.11 Magnification fig. 7.9.

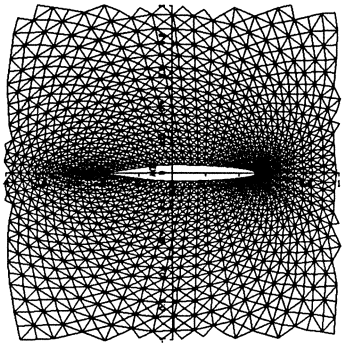


Fig. 7.12 Magnification fig. 7.10.

8 Non-linear iterative solution methods

An unpublished extension to all previous chapters presenting the damped inexact Newton algorithm and iterative solution methods used for the numerical tests presented. Also the computation of the Jacobian matrix for all partial diffential equations studied in this thesis is presented.

8.1 The Jacobian matrix

The Jacobian matrix, as used for the derivation of the discretization error estimates and the damped inexact Newton solution method, turns out to be easy to determine for all differential equations considered in this thesis. After the examination of this Jacobian matrix the damped inexact Newton method using it and the iterative solution methods are presented. The formulas and solution methods presented are the basis for all numerical test in the previous chapters.

Lemma 8.1.1 *Let $\underline{\nabla}_x$ denote the gradient in space dimension and $\underline{\nabla}$ the gradient in space and time. Further let $F(\boldsymbol{x}, u, |\underline{\nabla}_x u|^2)$ for $u \in \mathrm{H}^1_\gamma(Q)$ be defined by*

$$\langle F(\boldsymbol{x}, u, |\underline{\nabla}_x u|^2), v\rangle = \int_Q (E(\boldsymbol{x}, u, |\underline{\nabla}_x u|^2)\underline{\nabla}_x u)^T \underline{\nabla}_x v \; d\boldsymbol{x}dt + \int_Q f(\boldsymbol{x}, u, \underline{\nabla} u)v \; d\boldsymbol{x}dt + \oint_{\Gamma_N} vh \, ds$$

for all functions $v \in \mathrm{H}_0^1(Q)$*, where* E *is a diagonal matrix and the function* h *describes the Neumann boundary conditions. Then the Jacobian matrix is given by*

$$\begin{aligned}\langle F'(x, u, |\underline{\nabla}_x u|^2)w, v\rangle =& \langle \frac{\partial F}{\partial w}(x, u, |\underline{\nabla}_x u|^2), v\rangle \\ =& \int_Q \left[\{E + 2E'\underline{\nabla}_x u \underline{\nabla}_x u^T\}\underline{\nabla}_x w\right]^T \underline{\nabla}_x v + \\ & w\left[(\frac{\partial E}{\partial u})\underline{\nabla}_x u\right]^T \underline{\nabla}_x v + [w\frac{\partial}{\partial u}f + \\ & \underline{\nabla} w^T \frac{\partial f}{\partial \underline{\nabla} u}] \cdot v \; dx dt\end{aligned}$$

for all functions $u \in \mathrm{H}_\gamma^1(Q)$ *and all* $v, w \in \mathrm{H}_0^1(Q)$.

Proof. This follows straighty using elementary Banach-space analysis concerning partial derivatives of functionals and exploiting the definition (1.3.4) and lemma 3.3.1. □

Note that the Jacobian matrix is no longer symmetric in v, w in the case that E depends on u or f depends on $\underline{\nabla}_x u$. For numerical tests considering static two-dimensional partial differential equations, the integration in time, which is part of the definition of F and its derivative, is simply omitted. The integrals involved are approximated elementwise (for all triangles in a grid $\mathcal{Q}$ covering Q) with the use of a second degree Gaussian quadrature rule, as can be found in chapter 7. It should be noted that for all linear partial differential problems studied, the use of linear or quadratic basis functions in combination with this quadrature rule leads to exact integration, i.e., integration without quadrature errors.

8.2 The damped inexact Newton method

The *damped inexact Newton method (DIN)* used for all numerical test is introduced in [5]. There it is shown that the method converges for all finite dimensional non-linear systems of equations $F(x) = 0$ with *uniformly positive definite* Jacobian matrix F'. It is closely related to the simpler damped Newton method, which can be formulated in pseudo programming code by

$k = 0$
While $\|F(\boldsymbol{x}^{(k)})\| > \varepsilon_{\text{nonlinear}}$
Do
$\boldsymbol{d}^{(k)} := \{F'(\boldsymbol{x}^{(k)})\}^{-1} F(\boldsymbol{x}^{(k)})$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \tau^{(k+1)} \boldsymbol{d}^{(k)}$
$k := k + 1$
Od

for a predetermined precision $\varepsilon_{\text{nonlinear}}$. Here the parameter $\tau^{(k+1)}$ is called the damping parameter and $\|\cdot\|$ denotes a norm. In the case $\tau^{(k)} = 1$ the method reduces to the classical *Newton method* where the correction $\boldsymbol{d}^{(k)} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ has to be computed exactly or for instance to machine precision accurate.

The damped inexact Newton method is called damped since the damping parameter $\tau^{(k)}$ is involved and it is said to be inexact since the subsequently computed corrections $\boldsymbol{d}^{(k)}$ are only computed to a certain predetermined precision, as can be seen in the following pseudo code describing the method

$\tau^{(0)} = 1$; $k = 0$
While $\|F(\boldsymbol{x}^{(k)})\| > \varepsilon_{\text{nonlinear}}$
Do
$\tau^{(k+1)} := \min(1, 2\tau^{(k)})$
Determine $\boldsymbol{d}^{(k)}$ such that
$\|F'(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} + F(\boldsymbol{x}^{(k)})\| < \varepsilon_{\text{linear}}^{(k)}$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \tau^{(k+1)} \boldsymbol{d}^{(k)}$
While $\|F(\boldsymbol{x}^{(k+1)})\| / \|F(\boldsymbol{x}^{(k)})\| > (1 - \gamma\tau^{(k+1)})$
Do
$\tau^{(k+1)} := \tau^{(k+1)}/2$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \tau^{(k+1)} \boldsymbol{d}^{(k)}$
Od
$k := k + 1$
Od

for given $\boldsymbol{x}^{(0)}$. Here $\varepsilon_{\text{linear}}^{(k)} = \rho^{(k)} \|F(\boldsymbol{x}^{(k)})\|$ with $\rho^{(k)}$ is a forcing sequence. In all numerical tests $\rho^{(k)}$ is chosen to be $\rho^{(k)} = \min\{\frac{1}{10}, \|F(\boldsymbol{x}^{(k)})\|\}$. The scalar γ, which is an arbitrary scalar to be chosen in the interval $(0, 1)$, is set to $\gamma = \frac{8}{10}$ for all tests. Note that the damping parameter $\tau^{(k)}$ is determined automatically. This, together with the fact that no estimates involving the norm of the Jacobian matrix are needed, is the reason that

this method has been used. Other methods can for instance be found in [2].

One can show (see [5]) that the above choice for the forcing sequence guarantees the existence of a positive $K \in \mathbb{N}$ such that $\tau^{(k)} = 1$ for all $k \geq K$. Hence, for k large enough, the DIN method reduces to an undamped inexact method. In combination with the choice of the forcing sequence above, this will lead to quadratic convergence for $k \to \infty$, i.e.,

$$\|\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}\| \leq c\|\boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)}\|^2$$

for a certain positive scalar c. Note that the iterands $\boldsymbol{x}^{(k+1)}$ in the damped inexact Newton method could have been defined differently, with the use of

$$\|F'(\boldsymbol{x}^{(k)})(\boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}) + \tau^{(k)}F(\boldsymbol{x}^{(k)})\| < \varepsilon_{\text{linear}}^{(k)},$$

as presented in section 1.6. This approach simplifies the notations involved since no separate search direction $\boldsymbol{d}^{(k)}$ is needed. However, it is not used for the actual computations, since for the possible case of a very small damping parameter, the vector $\tau^{(k)}F(\boldsymbol{x}^{(k)})$ gets too small.

For all numerical tests presented, including the linear problems, the DIN method is used for the solution of the partial differential equation, discretized with the use of finite element basis functions where F above denotes the gradient vector and F' its related Jacobian matrix. In order to simplify the notations for the presentation of the iterative solution methods in the next section, let for given $\boldsymbol{x}^{(k)}$, the matrix $A = F'(\boldsymbol{x}^{(k)})$ and vector $\boldsymbol{b} = -F(\boldsymbol{x}^{(k)})$ be defined by

$$[A]_{ij} = \langle F'(\boldsymbol{x}^{(k)})\varphi_j, \varphi_i \rangle \text{ and } [b]_i = \langle F(\boldsymbol{x}^{(k)}), \varphi_i \rangle,$$

using the finite element basis as in chapter 5, the Gaussian quadrature rule as in (7.4.1) and the definition of F and F' given in lemma 8.1.1. Note that this notation implies that $\left|F'(\boldsymbol{x}^{(k)})\boldsymbol{d}^{(k)} + F(\boldsymbol{x}^{(k)})\right| = \left|\boldsymbol{r}^{(k)}\right|$, where $\boldsymbol{r}^{(k)}$ denotes the *residual*. This means that the iterative solution methods to be presented will have a stopping criterion $\|\boldsymbol{r}^{(k)}\| < \varepsilon_{\text{linear}}^{(k)}$.

The matrix A is stored in the special row-wise ordered manner as introduced in section 5.6. The node numbering, determining the sparsity pattern of the matrix and the vector, satisfies (5.4.2), in

order to guarantee a block structured matrix in the case that a hierarchical finite element basis representation is used. Further, assuming that $\boldsymbol{x}^{(0)} \in \mathcal{H}_\gamma(\mathcal{Q})$ (see (5.9.1)) satisfies the – inhomogeneous – Dirichlet boundary conditions, the search direction must satisfy homogeneous boundary conditions, i.e., $\boldsymbol{d}^{(k)} \in \mathcal{H}_0(\mathcal{Q})$. In order to guarantee this the matrix A and vector $\boldsymbol{b}$ are always modified as follows. Let φ_i be a node defined at a Dirichlet boundary point, then for all j

$$[A]_{ij} = [A]_{ji} = [\boldsymbol{b}]_i = 0 \text{ and } [A]_{ii} = 1$$

Note that this procedure in some cases also has to be followed in order to prevent the matrix from being singular. This would for instance be the case if F is the Laplacian functional and a standard nodal finite element basis is used for the construction of the matrix A. Then, without the modification above, A maps the vector $[1, \ldots, 1]^T$, onto 0, i.e., the unmodified matrix A is singular.

8.3 The solution of systems of linear equations

A linear system of equations $A\boldsymbol{x} = \boldsymbol{b}$ as obtained in the previous section, emanating from the discretization of a partial differential equation, is usually sparse and of large order. Iterative solution methods, exploiting these properties, are mostly much cheaper solution methods than the classical Gaussian elimination method. Main reason is the fact that the latter method causes a large amount of *matrix fill in* (see [4]) and therefore demands the whole matrix to be stored in the computer memory. Using an iterative solution method, only non-zero entries of the matrix have to be stored, and the matrix A^{-1} will not be computed.

In order to speed up the convergence rate of an iterative method a preconditioner $C = LU \approx A$ is constructed by the *incomplete Gaussian factorization ILU* (see for instance [4], pages 40, 41). Here incomplete means that during the row-wise ordered elimination without pivoting, corrections are neglected if they do not correspond to couplings (i, j) of the sparsity patterns J, presented in section 5.5. The elimination takes into account the node numbering – reflecting the level of refinement (5.4.2) – by starting to eliminate using the nodes of the highest level, implying that the resulting preconditioner LU depends on the

numbering of the nodes. Tests in [6] using a standard nodal finite element basis have shown the influence from several numbering strategies on the rate of convergence.

There exist many other possible preconditioning techniques for symmetric positive definite matrices A emanating from elliptic differential equations, but here the incomplete LU preconditioner is used since for time-dependent problems the related A is not symmetric. However, in the case of a positive definite matrix A one could consider also the elementwise factorization technique proposed in [7].

All numerical test performed use one of the described iterative solution methods below. The initial approximation $\boldsymbol{x}^{(0)}$ is always taken to be equal to $C^{-1}\boldsymbol{b}$. The stopping criterion used by all iterative methods is $\|\boldsymbol{r}^{(k)}\| < \varepsilon_{\text{linear}}$ for a given precision $\varepsilon_{\text{linear}}$, where $\|\cdot\|$ denotes the Euclidian norm. This differs from the proposed criteria in the literature (see e.g. [4] and [11]), for comparison purposes. As all norms on a finite dimensional vector space are equivalent, there is no specific reason to choose the Euclidian norm in the stopping criterion. For example, the max-norm $\|\cdot\|_\infty$ could have been used as a cheap alternative. Note that the stopping criterion does not take into account the type of finite element basis used.

The first and oldest method to be considered is the *preconditioned conjugate gradient PCG* solution method as described in e.g. [4]. This method in general can only be used to solve symmetric positive definite linear systems of equations and is given by the following pseudo code

$\boldsymbol{r}^{(0)} := A\boldsymbol{x}^{(0)} - \boldsymbol{b}$
$\boldsymbol{d}^{(0)} := -C^{-1}\boldsymbol{r}^{(0)}$
$k := 0$
While $\|\boldsymbol{r}^{(k)}\| > \varepsilon_{\text{linear}}$
Do
  $\alpha = -(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})$
  $\beta = \alpha/(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})$
  $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \beta\boldsymbol{d}^{(k)}$
  $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \beta\boldsymbol{d}^{(k)}$
  $\gamma = (\boldsymbol{r}^{(k+1)}, C^{-1}\boldsymbol{r}^{(k+1)})/\alpha$
  $\boldsymbol{d}^{(k+1)} = \beta\boldsymbol{d}^{(k)} - C^{-1}\boldsymbol{r}^{(k+1)}$
  $k := k + 1$

Od

for all numerical tests performed.

As the time-dependent partial differential equations give rise to non-symmetric systems of equations other iterative solution methods are necessary. For the solution of nonsymmetric positive definite problems the more recently developed preconditioned *generalized conjugate gradient least squares GCGLS* method and *conjugate gradient squared CGS* method have been used. The GCGLS method can be described by (see e.g. [1])

$\boldsymbol{r}^{(0)} := A\boldsymbol{x}^{(0)} - \boldsymbol{b}$
$\boldsymbol{p}^{(0)} := C^{-1}\boldsymbol{r}^{(0)}$
$\boldsymbol{d}^{(0)} := -\boldsymbol{p}^{(0)}$
$k := 0$
While $\|\boldsymbol{r}^{(k)}\| > \varepsilon_{\text{linear}}$
Do
$\alpha^{(k)} := -(\boldsymbol{r}^{(k)}, A\boldsymbol{d}^{(k)})/(A\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})$
$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} + \alpha^{(k)}\boldsymbol{d}^{(k)}$
$\boldsymbol{r}^{(k+1)} := \boldsymbol{r}^{(k)} + \alpha^{(k)}A\boldsymbol{d}^{(k)}$
$\boldsymbol{p}^{(k+1)} := C^{-1}\boldsymbol{r}^{(k+1)}$
$\boldsymbol{h}^{(k+1)} := A^T A\boldsymbol{p}^{(k+1)}$
$\beta_{k-j}^{(k)} := (\boldsymbol{h}^{(k+1)}, \boldsymbol{d}^{(k-j)})/(A\boldsymbol{d}^{(k-j)}, A\boldsymbol{d}^{(k-j)}), \quad j = 0, \ldots, k$
$\boldsymbol{d}^{(k+1)} := -\boldsymbol{p}^{(k+1)} + \sum_{j=1}^{k} \beta_{k-j}^{(k)}\boldsymbol{d}^{(k-j)}$
$k := k + 1$
Od

For practical reasons a truncated version of this algorithm is used, where only the last l search directions are used to determine the new search direction. The summation term in the algorithm above then ranges from $j = \max(0, k - l + 1)$ to $j = k$. One can prove, if the symmetric part is positive definite (see [1], [3]), that the residuals $\boldsymbol{r}^{(k)}$ converge monotonically to zero.

The CGS solution method (see [11]) is given by the code

$\boldsymbol{p}^{(0)} := 0; \boldsymbol{q}^{(0)} := 0; \alpha^{(0)} := 1;$
$\boldsymbol{y}^{(0)} := L^{-1}\boldsymbol{b}$
$\boldsymbol{r}^{(0)} := \boldsymbol{b} - AU^{-1}\boldsymbol{y}^{(0)}$
$\boldsymbol{c}^{(0)} := L^{-1}\boldsymbol{r}^{(0)}$

$\hat{\boldsymbol{r}}^{(0)} := \boldsymbol{c}^{(0)}$
$k := 0$
While $\|\boldsymbol{r}^{(k)}\| > \varepsilon_{\text{linear}}$
Do
$\alpha^{(k+1)} := (\hat{\boldsymbol{r}}^{(0)}, \boldsymbol{c}^{(k)})$
$\beta := \alpha^{(k+1)} / \alpha^{(k)}$
$\boldsymbol{u}^{(k+1)} := \boldsymbol{c}^{(k)} + \beta \boldsymbol{q}^{(k)}$
$\boldsymbol{c}^{(k+1)} := \boldsymbol{q}^{(k)} + \beta \boldsymbol{p}^{(k)}$
$\boldsymbol{p}^{(k+1)} := \boldsymbol{u}^{(k+1)} + \beta \boldsymbol{c}^{(k+1)}$
$\boldsymbol{c}^{(k+1)} := L^{-1} A U^{-1} \boldsymbol{p}^{(k+1)}$
$\gamma := \alpha^{(k+1)} / (\hat{\boldsymbol{r}}^{(0)}, \boldsymbol{c}^{(k+1)})$
$\boldsymbol{q}^{(k+1)} := \boldsymbol{u}^{(k+1)} - \gamma \boldsymbol{c}^{(k+1)}$
$\boldsymbol{c}^{(k+1)} := \boldsymbol{u}^{(k+1)} + \boldsymbol{q}^{(k+1)}$
$\boldsymbol{y}^{(k+1)} := \boldsymbol{y}^{(k)} + \gamma \boldsymbol{c}^{(k+1)}$
$\boldsymbol{r}^{(k+1)} := \boldsymbol{r}^{(k)} - \gamma A U^{-1} \boldsymbol{c}^{(k+1)}$
$\boldsymbol{c}^{(k+1)} := L^{-1} \boldsymbol{r}^{(k+1)}$
$k := k + 1$
Od
$\boldsymbol{x} := U^{-1} \boldsymbol{y}^{(k+1)}$

for arbitrary $\hat{\boldsymbol{r}}^{(0)}$. In all tests performed $\hat{\boldsymbol{r}}^{(0)} := L^{-1} \boldsymbol{r}^{(0)}$. Note that the algorithm above computes the solution $\boldsymbol{x}$ on a transformed bases (transformation with L^{-1}).

There is no proof of convergence available for this method, possibly due to the fact that breakdown can occur for $(\hat{\boldsymbol{r}}^{(0)}, \boldsymbol{c}^{(k)}) = 0$ for certain k. Contrary to the GCGLS method, the residuals $\boldsymbol{r}^{(k)}$ do not converge monotonically to zero in general. Prior to convergence, they can vary several orders of magnitude (early literature can be found in [10]). However, recently, v.d. Vorst [12] introduced the related Bi-CGSTAB iterative solver which behaves more stable, also for problems which contain a moving shock. The convergence rate is approximately that of the CGS method.

Recently several parallel finite element methods were developed by Layton, Rabier and Maubach (see for instance [8] and [9]). Depending on the choice of the finite element basis, these methods are element-wise parallel and can solve non-linear partial differential equations with constraints in parallel.

8.4 References

[1] Axelsson O., *A generalized conjugate gradient, least square method,* Numerische Mathematik, 51(1987), 209-227

[2] Axelsson O., *On global convergence of iterative methods,* **in** Iterative Solution of Nonlinear Systems of Equations, 1-19 LNM#953, (Ansorge R., Meis Th. and Törnig W. eds.), Springer Verlag, 1982

[3] Axelsson O., *A restarted version of a generalized preconditioned conjugate gradient method,* Communications in Applied Numerical Methods, 4(1988), 521-530

[4] Axelsson O. and Barker V.A., *Finite Element Solution of Boundary Value Problems,* Academic Press, Orlando, Florida, 1984

[5] Dembo R.S., Eisenstat S.C. and Steihaug T., *Inexact Newton methods,* SIAM Journal on Numerical Analysis, 19(1982), 400-408

[6] Duff I.S. and Meurant G.A., *The effect of ordering on preconditioned conjugate gradients,* BIT, 29(1989), 635-657

[7] Kaasschieter E.F., *A general finite element preconditioning for the conjugate gradient method,* BIT, 29(1989), 824-849

[8] Layton W., Maubach J.M. and Rabier P., *Parallel algorithms for maximal monotone operators of local type,* accepted by Numerische Mathematik, 1994

[9] Layton W., Maubach J.M. and Rabier P., *Robust methods for highly non-symmetric problems,* **in** Proceedings of the International Conference on Domain Decomposition Methods, Penn State University, Pennsylvania, 1993

[10] Markham G., *The CG-Squared method for solving asymmetric systems of linear equations,* Note Tprd/L/Apm/012, CERL 1983

[11] Sonneveld P., *CGS, a fast Lanczos-type solver for non-symmetric linear systems,* SIAM Journal on Scientific and Statistical Computing, 10(1989), 36-52

[12] Vorst H.A. van der, *Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems,* internal report of the Mathematical Institute, University of Utrecht, The Netherlands, 1990

Index

The index table provides for most of the definitions to be found inside this textbook the page number of their introduction and in some instances some additional references.