A Unifying Theory for Nonlinear Additively and Multiplicatively Preconditioned Globalization Strategies: Convergence Results and Examples From the Field of Nonlinear Elastostatics and Elastodynamics

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

Vorgelegt von
Christian Groß
aus
Remagen

Bonn, Juli 2009

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Rheinischen Friedrich-Wilhelms Univerversität Bonn

1. Gutachter: Prof. Dr. Rolf Krause
2. Gutachter: Prof. Dr. Helmut Harbrecht
Tag der Promotion: 11.09.2009

Diese Arbeit ist mit Unterstützung der von der Deutschen Forschungsgemeinschaft getragenen Bonn International Graduate School (BIGS) und des SFB 611 entstanden.

Mano Brangutei
Für meine Liebste

Abstract

The solution of nonlinear programming problems is of paramount interest for various applications, such as for problems arising from the field of elasticity. Here, the objective function is a smooth, but nonlinear and possibly nonconvex functional describing the stress-strain relationship for material classes. Often, additional constraints are added to model, for instance, contact. The discretization of the resulting partial differential equations, for example with Finite Elements, gives rise to a finite dimensional minimization problem of the kind

$$u \in \mathcal{B} \subset \mathbb{R}^n : J(u) = \min! \tag{M}$$

where $n \in \mathbb{N}$, and $J : \mathbb{R}^n \to \mathbb{R}$, sufficiently smooth. The set of admissible solutions $\mathcal{B}$ is given by $\mathcal{B} = \{u \in \mathbb{R}^n \mid \underline{\phi}_i \leq u_i \leq \overline{\phi}_i \text{ for all } i = 1, \ldots, n\}$ where $\underline{\phi}, \overline{\phi} \in \mathbb{R}^n$.
The solution of such a minimization problem can be carried out with various numerical methods. From an analytical point of view it is of interest under which assumptions a numerical solution strategy computes a (local) solution of the minimization problem. Here, basically two classes of globalization strategies, Linesearch and Trust-Region methods, exist which are able to solve (M) even if J is nonconvex. Though, the interest of a user lies in the efficiency and robustness of the employed tool. In fact, it is of great importance that a solution is, independent of the employed parameters, rapidly carried out.
In particular, a modern nonlinear solution strategy must necessarily be able to be applied for (massive) parallel computing. The first step would, indeed, be employing parallelized linear algebra for the Trust-Region and Linesearch strategy. But, to guarantee convergence, traditional solution strategies damp the computed Newton corrections which might slow down the convergence.
Therefore, different extensions for the traditional schemes were developed, such as the two (additive) schemes PARALLEL VARIABLE DISTRIBUTION (PVD) [FM94], PARALLEL GRADIENT DISTRIBUTION (PGD) [Man95] and the (multiplicative) schemes MG/OPT [Nas00], recursive Trust-Region methods (RMTR) [GST08, GK08b] and recursive Linesearch methods (MLS) [WG08]. Both, the nonlinear additive and multiplicative scheme, aim at a solution of related but "smaller" minimization problems to compute corrections or search directions. In particular, the paradigm of the PVD and PGD schemes is to asynchronously compute solutions of local minimization problems which are combined to a global correction. The recombination process itself is the solution of another nonlinear programming problem. The multiplicative schemes, in contrast, aim at a solution of coarse level problems starting from a projection of the current fine level iterate. As numerical examples in [GK08b, GMS$^+$09] and [WG08] have shown, combining multiplicative schemes with a "global" smoothing step yields clearly improved rates of convergence with little computational overhead.
In the present thesis we will show that these additive and multiplicative schemes can be regarded as a nonlinear right preconditioning of a globalization strategy. Moreover, novel, generalized nonlinear additive and multiplicative frameworks are introduced which fit into the nonlinear preconditioning context. In numerous examples, we comment on the relationship to state-of-the-art domain decomposition frameworks such as hierarchical and vertical decompositions and explain how these decompositions fit into the presented context. In a second step, Trust-Region and Linesearch variants of the preconditioning frameworks are presented and first–order convergence is shown.

As it turns out, the presented multiplicative Trust-Region concept is based on the RMTR framework employed in [GK08b] extending it to more arbitrary domain decompositions. On the other hand, the multiplicative Linesearch methods are based on the MLS scheme in [WG08]. Here, the original assumptions are weakend allowing for the solution of non-smooth nonlinear programming problems. Moreover, we present a novel nonlinear additive preconditioning framework, along with actual Trust-Region and Linesearch implementations. As it turns out, well-balanced a priori and a posteriori strategies and a novel subset objective function which allow for straight-forwardly implementing the presented frameworks and showing first–order convergence. As will be highlighted, these novel additive preconditioning strategies are perfectly suited to be employed for massive parallel computing. Furthermore, remarks on second–order convergence are stated.
To motivate the presented solution strategies, systems of PDEs and equivalent minimization problems arising from the field of elasto-statics and elasto-dynamics are introduced. Moreover, we will show that – after discretization – the resulting objective functions satisfy the assumptions stated for showing convergence of the respective globalization strategies. Furthermore, various numerical examples employing these objective functions are presented showing the efficiency and robustness of the presented nonlinear preconditioning frameworks. Comments on the computation times, the number of iterations, the computation of search directions, and the actual implementation of the frameworks are stated.

Danksagung

An dieser Stelle möchte ich mich bei meinem Erstbetreuer Rolf Krause bedanken, der mir vorgeschlagen hat, dieses außerordentliche interessante, breite und anspruchsvolle Forschungsthema zu bearbeiten. Zudem stand er mir oft mit Rat und Tat zur Seite und hat sich immer darum bemüht, dass ich meine Ergebnisse auch in frühen Stadien meines Forschungsprojektes in (Konferenz-) Vorträgen darstelle. Desweiteren danke ich Helmut Harbrecht für die reibungslose Übernahme der Betreuerpflichten an der Universität Bonn, seine Unterstützung und sein ausführliches Feedback. Auch möchte ich mich sehr bei Andreas Weber für die Förderung noch während meines Diplomstudiums danken.
Besonderer Dank gilt meinen Kollegen Thomas Dickopf und Mirjam Walloth, die immer ein offenes Ohr für oftmals technische Fragen hatten. Aufgrund ihrer überaus aufgeschlossenen Einstellung wurde oft aus einer Idee ein mathematisch korrektes Resultat. Gleichsam danke ich Johannes Steiner und Britta Joswig für die Wirbelgeometrie, die sie erstellt haben und die ich im Abschnitt 5.6.8 verwenden durfte. Auch danke ich allen Kollegen am INS und am ICS, insbesondere Dorian Krause für das schnelle Bereitstellen des Servers in Lugano.
Ich danke in besonderem Maße der Bonn International Graduate School, die mir nicht nur ein großzügiges Promotionsstipendium gewährt hat, sondern auch viele Konferenzteilnahmen und einen Aufenthalt an der Columbia University in the City of New York zu großen Teilen finanziert hat. Für das Bereitstellen einer hervorragenden Infrastruktur danke ich besonders dem Institut für Numerische Simulation der Rheinischen Friedrich-Wilhelms Universität Bonn und dem Institute of Computational Science der Università della Svizzera italiana in Lugano.
Die Begebenheiten, die zu dieser wissenschaftlichen Arbeit geführt haben sind vielfältig. Jedoch haben gerade die frühen Weichenstellungen in ganz besonderem Maße dazu geführt, dass ich studiert habe und diese Arbeit nun geschrieben habe. Daher möchte ich mich ganz besonders bei meinen wichtigsten Förderern und Vorbildern, meinen Eltern Elisabeth und Wolfgang Groß und meinem Bruder Thomas, bedanken. Zu guter Letzt möchte ich mich ganz herzlich bei meiner Frau Rimante für das viele Zuhören, das gute Zureden und das Tolerieren überlanger Arbeitstage bedanken.

Contents

1 Introduction **1**
- 1.1 The Nonlinear Model Problem . . . 4
- 1.2 The Constitutional Equations and their Discretization . . . 5
  - 1.2.1 Kinematics and Conservation Laws . . . 6
  - 1.2.2 Elastodynamic and Elastostatic Model Problems in H^1 . . . 7
- 1.3 Discretization . . . 9
  - 1.3.1 Temporal Discretization . . . 10
  - 1.3.2 Spatial Discretization . . . 12

2 State of the Art Globalization Strategies **15**
- 2.1 The "Traditional" Trust-Region Framework . . . 16
  - 2.1.1 Assumptions on J and the Trust-Region Model . . . 16
  - 2.1.2 Decrease Ratio and Trust-Region Update . . . 17
  - 2.1.3 Constraints and Scaling Functions . . . 18
  - 2.1.4 Convergence to First–Order Critical Points . . . 19
  - 2.1.5 Second–Order Convergence . . . 23
- 2.2 The "Traditional" Linesearch Framework . . . 23
  - 2.2.1 Assumptions on the Objective Function . . . 24
  - 2.2.2 Assumptions on the Search Direction . . . 24
  - 2.2.3 The Armijo Condition as Step Length Control . . . 25
  - 2.2.4 Convergence to First–Order Critical Points . . . 27
  - 2.2.5 Second–Order Convergence . . . 28

3 A Generic Nonlinear Preconditioning Framework **31**
- 3.1 The Concept behind Nonlinearly Preconditioned Globalization Strategies . . . 32
  - 3.1.1 Nonlinear Right Preconditioning . . . 32
  - 3.1.2 Nonlinear Additive and Multiplicative Update Operators . . . 34
  - 3.1.3 Decomposition of the $\mathbb{R}^n$ and Construction of the Transfer Operators . . . 35
  - 3.1.4 The Transfer Operators . . . 35
  - 3.1.5 Example: a Multilevel Decomposition of Finite Element Spaces . . . 38
  - 3.1.6 Example: (Non-) Overlapping Domain Decomposition Methods . . . 40
- 3.2 Abstract Formulation of the Nonlinear Additive Preconditioning Operator . . . 41
  - 3.2.1 Derivation of the Additive Subset Objective Function . . . 41
  - 3.2.2 Example: The Forget-Me-Not Approach . . . 42
  - 3.2.3 The Nonlinear Additive Update and Preconditioning Operators . . . 43
  - 3.2.4 Example: Parallel Variable Distribution . . . 44
  - 3.2.5 The Construction of the Subset Obstacles in the Additive Setting . . . 45
- 3.3 Abstract Formulation of the Nonlinear Multiplicative Preconditioning Operator . . . 47
  - 3.3.1 Derivation of the Multiplicative Subset Objective Function . . . 47
  - 3.3.2 The Nonlinear Multiplicative Update and Preconditioning Operator . . . 48

3.3.3 Example: A Multiplicative Algorithm of Gauß-Seidel type 49
3.3.4 Example: A Multilevel V-Cycle Algorithm 50
3.3.5 The Construction of the Subset Obstacles in the Multiplicative Setting 51

4 Nonlinear Additively Preconditioned Globalization Strategies 53
4.1 Nonlinear Additively Preconditioned Trust–Region Methods 53
4.1.1 The APTS Framework 54
4.1.2 Convergence to First-Order Critical Points 57
4.2 Nonlinear Additively Preconditioned Linesearch Methods 62
4.2.1 The APLS Framework 63
4.2.2 A Modified Armijo Condition for the Additive Context 64
4.2.3 Convergence to First–Order Critical Points 67
4.3 A Remark on Parallel Communication 70
4.4 A Remark on Second-Order Convergence 71

5 Nonlinear Multiplicatively Preconditioned Globalization Strategies 73
5.1 Nonlinear Multiplicatively Preconditioned Trust-Region Methods 74
5.1.1 The MPTS Framework 74
5.1.2 Convergence to First-Order Critical Points 77
5.2 Combined Nonlinearly Preconditioned Trust-Region Methods 83
5.3 Nonlinear Multiplicatively Preconditioned Linesearch Methods 85
5.3.1 The MPLS Framework 85
5.3.2 A Modified Armijo Condition 86
5.3.3 Convergence to First–Order Critical Points 90
5.4 Combined Nonlinearly Preconditioned Linesearch Methods 94
5.5 A Remark on Second-Order Convergence 97
5.6 Non-Linear Elasto-Static PDEs 98
5.6.1 Visualization 102
5.6.2 The Nonlinear Update Operator 103
5.6.3 Unconstrained Minimization Problem: Compression of a Cube 106
5.6.4 Unconstrained Minimization Problem: Simulation of a Can 110
5.6.5 Unconstrained Minimization Problem: Simulation of an Iron wheel 114
5.6.6 Constrained Minimization Problem: Contact with a Small Obstacle 116
5.6.7 Constrained Minimization Problem: Simulation of a Can 119
5.6.8 Constrained Minimization Problem: Simulation of an Intervertebral Disk . . 122
5.7 Non-Linear Elasto-Dynamic PDEs 124
5.7.1 Example: Dynamic Simulation of a Can 124
5.7.2 Example: Dynamic Simulation of a Hollow Geometry 126

6 Appendix: Implementational Aspects 129
6.1 NLSolverLib 129
6.2 Asynchronous Linear Solvers 132
6.3 IOLib 132
6.4 InterpreterLib 132

Bibliography 139

1 Introduction

Ever since 1958 till the beginning of this millenium, the number of transistors placed on an integrated circuit has doubled every two years, yielding extremely fast computers. In particular, at the end of the 1990s, the computational power of the TOP 500 computers, the 500 fastest, civil used computers, was just under 50,000 Gflops. Today, the TOP 500 computers achieve a peak performance of 25,400,000 Gflops [TOP08], an annual increase of 2800%. Though, recently, this increase is in major parts due to the massive parallelization of computers, rather than due to the acceleration of individual processors. Therefore, in order to harness the computational power of modern supercomputers, algorithms must be developed and implemented with the capability to run in parallel.
In case of Finite Elements for the discretization of problems arising from the field of elasticity, the parallelization affects the linear algebra, linear solvers, often the geometry and, therefore, quadrature rules and the assembling processes. As it turns out, most of the affected routines can run in parallel with little parallel communication, such as, for instance, the quadrature. In contrast, the iterative solution of linear systems of equations makes much parallel communication necessary since locally computed solutions must be recombined to a global solution, for instance, to compute updated residuals.

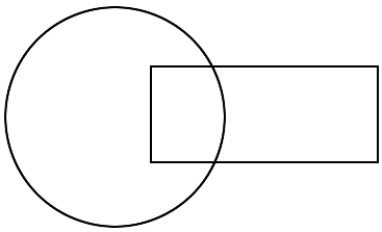

Figure 1.1: Domain decomposition methods go back to the 1870s, when H.A. Schwarz proposed an alternating domain decomposition method [Sch90]. In this original domain decomposition of H.A. Schwarz the domain is decomposed into an overlapping rectangle and a circle.

As a matter of fact, parallelized linear algebra enables scientists to compute the solution of highly complex problems, such as large-scale nonlinear and possibly nonconvex minimization problems arising, for instance, from the field of *nonlinear* elasticity. As it turns out, if the objective function, in this case the *stored energy function*, is highly nonlinear but convex, Newton's method is able to compute a solution of the minimization problem. But, in the case of nonconvex objective functions, the same holds only if the initial iterate is *sufficiently good*. In this case, it suffices to employ a state–of–the–art parallelized linear solver to compute Newton corrections. But, generally it is unknown whether the initial iterate is sufficiently good or not. Therefore, one must employ a *globalization strategy* – e.g., Trust-Region or Linesearch strategies – to ensure convergence to critical points.
Both strategies, Trust-Region and Linesearch strategies, combine the computation of quasi-Newton corrections, and the computation of adequate damping parameters to ensure convergence to critical points. The damping parameters themselves depend on the "quality" of the *search direction*, e.g., the Newton corrections, and the local nonlinearity of the objective function. In turn, in regions with strong nonlinearities of the objective function often the damping parameters must be chosen

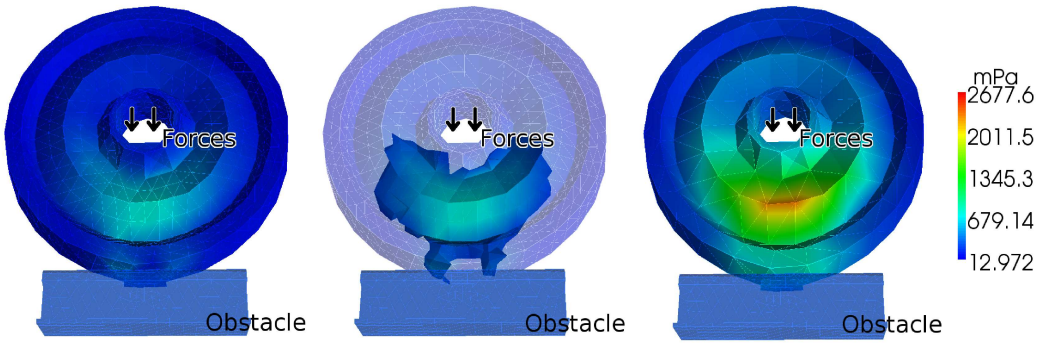

Figure 1.2: Different Scales: A minimization problem arising from nonlinear elasticity, where for given boundary values energy optimal displacements are computed. The colors represent the von-Mises stresses (cf., Section 5.6.1) within the deformed configuration. *Left:* here we visualize the von-Mises stresses on the finest scale which obviously, vary in different parts of the geometry. Therefore, we visualize in the *middle figure* the strongest local stresses on the fine scale. *Right:* here, we show the coarse scale von-Mises stresses which look similar to the fine scale stresses. The geometry is from [NZ01].

sufficiently small to ensure an actual decrease of the objective function, even for sufficiently good search directions. As it turns out, this problem increases with the number of unknowns since the step–length depends on the strongest *local* nonlinearity. This particularly means that even if nonlinearities occur only locally or in certain spectra they govern the whole solution process of the minimization problem.
Thus, in the last decades, two different approaches emerged to bypass this problem by attacking nonlinearities

- on different scales
- locally w.r.t. the domain

To handle nonlinearities on different spectra, in the early 1980s, A. Brandt introduced the FULL APPROXIMATION SCHEME (FAS) [Bra81], the first nonlinear multigrid method. Here, the restricted "fine scale" gradients are combined with the gradient of an arbitrarily chosen nonlinear "coarse level" objective function. One important difference to linear multigrid strategies is that due to the nonlinearity of the resulting coarse level problem, the choice of an initial iterate influences the resulting coarse level correction. Though, due to the method's formulation, convergence may only be proven for convex minimization problems or for sufficiently well chosen initial iterates.
To overcome this problem S. Nash introduced in 2000 the MG/OPT method, a reformulation of the FAS scheme which combines a new objective function with a globalization strategy such as a Linesearch strategy [Nas00]. By now, several Trust-Region (called RMTR) and further Linesearch (called MLS) implementations of the MG/OPT framework have been introduced by S. Gratton et al. [GST08, GMTWM08], Z. Wen and D. Goldfarb [WG08] and C. Groß and R. Krause [GK08b, GK08c]. Similarly to S. Nash's approach, the MLS strategy and the RMTR strategies deterministically compute initial iterates on the coarse levels. In fact, it is proposed to employ the restriction operator to compute an approximation to the fine level iterate. Also damped restriction operators were proposed to improve the rates of convergence [GMS$^+$09] which slightly affects the analysis of the RMTR method. But, as it turns out in the case of nonlinear elasticity [GK08b] the L^2-projection seems to yield better coarse level corrections and faster convergence than employing the restriction operator.

The analysis of both, the MLS and the RMTR strategy, is based on the fact that an interpolated coarse level correction can be regarded as a search direction for the fine-level problem. In turn, this enables the respective authors to prove convergence under modest assumptions. Though, in order to derive a multiplicative framework which is also suited alternating domain decomposition methods, in the present thesis, we will generalize the recursive Trust-Region scheme in [GK08b] to a multiplicative Trust-Region framework. Moreover, the multiplicative Linesearch scheme in this thesis will generalize the MLS method to the non–smooth context. In order to prove convergence of this scheme, we show that the assumptions for the MLS method can be weakend by introducing different control strategies.

On the other hand, in the 1990s, frameworks for asynchronous and nonlinear globalization strategies called PARALLEL VARIABLE DISTRIBUTION (PVD) and PARALLEL GRADIENT DISTRIBUTION (PGD) were introduced by M. C. Ferris and O. L. Managsarian [FM94, Man95]. Therefore, both approaches asynchronously solve local minimization problems and recombine the computed corrections employing a set of damping parameters. The computation of the damping parameters, though, is the result of the solution of another possibly nonconvex minimization problem. Both frameworks, the PVD and PGD framework, are globalization strategies which, in addition, can be employed to resolve local nonlinearities. Moreover, X.-C. Cai and D. E. Keyes introduced in 2002 the ADDITIVE PRECONDITIONED INEXACT NEWTON (ASPIN) method [CK02], a *nonlinear additive Schwarz method*, based on a left preconditioning of the first–order conditions. An important feature of the ASPIN method is an alternative recombination step, which is carried out by solving a linear system of equations. But, similarly to the full approximation scheme, convergence of the ASPIN method may only be proven for sufficiently good initial iterates [CK02, AMPS08].

In fact, the asynchronous solution of local nonlinear minimization problems enables the respective method to resolve local nonlinearities without being governed by a global step–length constraint. But, moreover, these additive frameworks are good starting points for the derivation of nonlinear additively (right) preconditioned globalization strategies which aim at the *massive parallel* solution of nonlinear minimization problems. Since, as far as it is possible to avoid computing a set of damping parameters, the ASPIN method and (for certain configurations) the PVD/PGD algorithms reduce the overall parallel communication, as it is desirable for parallel solution strategies.

In order to avoid the expensive computation of global damping parameters, we will consider the additively computed correction as a search direction in the context of the global minimization problem. This point of view allows for deriving easy implementable standard Trust-Region and Linesearch control strategies reducing the set of damping parameters to one damping parameter or one Trust-Region radius. Along with an, in the additive context, novel objective function this results in a novel additive preconditioning framework. Moreover, under modest assumptions, we are able to prove convergence of the presented additively preconditioned Trust-Region and Linesearch strategies to first–order critical points.

Finally, we will introduce novel combined preconditioned Linesearch and Trust-Region strategies which employ both approaches, the additive and multiplicative approaches, within one preconditioning framework. Both methods are formulated based on the about to be presented multiplicative and additive schemes which enables us to straight-forwardly prove convergence to first–order critical points. As it will turn out, in numerous computed examples, carried out within a Finite Element framework, these combined preconditioned globalization strategies are considerably faster than the traditional schemes. Similarly, also the pure multiplicative and additive schemes yield in most computed examples faster convergence to critical points, than the traditional schemes. Here, we implemented exemplarily a nonlinear multigrid method as multiplicative and a nonlinear non-overlapping domain decomposition method as additive scheme. Moreover, we will comment on employing dif-

ferent decompositions frameworks within the concept of nonlinear preconditioning.

Overview

In the next chapter, Chapter 2, we will present standard implementations and convergence proofs for Trust-Region and Linesearch strategies. This has two purposes. On the one hand, we can compare the presented strategies to the novel preconditioned globalization strategies. On the other hand, we employ parts of the respective convergence proofs within our analysis of the additive and multiplicative preconditioning strategies.
In Chapter 3, we will introduce the abstract nonlinear right preconditioning frameworks. Here, we will present the additive and multiplicative domain decomposition frameworks along with the transfer operators. These are, in turn, employed in the multiplicative and additive context. Also the subset objective functions and assumptions on the subset constraints will be formulated in this chapter along with the additive and multiplicative update operators. Together, this allows for proving that the formally defined nonlinear preconditioning operators yield admissible search directions.
Actual nonlinear additively preconditioned globalization strategies will be presented in Chapter 4. Here the objective function and transfer operators as presented in Chapter 3 will be employed within an additive Trust-Region framework, the APTS method, and an additive Linesearch framework, the APLS method. Both presented novel additive preconditioning strategies will be analyzed in this section and convergence to first–order critical points will be shown. In Section 4.3, we will also comment on the overall parallel communication within the nonlinear additively preconditioned globalization strategies.
In Chapter 5, we will present and analyze multiplicative Trust-Region and Linesearch variants. Here, the RMTR strategy in [GK08b] will be extended to a more general multiplicative framework resulting in the nonlinear multiplicatively preconditioned Trust-Region strategy (MPTS) method. Moreover, we will present the nonlinear multiplicatively preconditioned Linesearch strategy (MPLS) which extends the approach of Z. Wen and D. Goldfarb [WG08] to the non-smooth case. Moreover, we will introduce and analyze novel combined multiplicatively and additively preconditioned Trust-Region and Linesearch methods.
Finally, in Chapter 5.5 and Chapter 6, we will explain the actual implementation of the respective nonlinear additive and multiplicative preconditioners and present numerical results. The implemented nonlinear solvers of the present thesis are part of the NLSOLVERLIB toolbox extending OBSLIB++ [Kra07b] and UG [BBJ+97]. In Chapter 5.5, the particular example problems and their solutions will be introduced. Moreover, we will present numerous comparisons considering the rates of convergence of

- additive, multiplicative, combined and standard Trust-Region strategies
- additive, multiplicative, combined and standard Linesearch strategies

1.1 The Nonlinear Model Problem

In the present thesis, we present several globalization frameworks which aim at the solution of the following non–linear, box-constrained minimization problem

$$u \in \mathcal{B} \subset \mathbb{R}^n : J(u) = \min! \tag{M}$$

where $n \in \mathbb{N}$, and $J : \mathbb{R}^n \to \mathbb{R}$. The objective function J is supposed to be at least continuously differentiable, but is neither assumed to be quadratic nor convex. Here, $\mathcal{B}$ denotes a set of admissible solutions with

$$\mathcal{B} = \{u \in \mathbb{R}^n \mid \underline{\phi}_i \leq u_i \leq \overline{\phi}_i \text{ for all } i = 1, \ldots, n\}$$

and $\underline{\phi}, \overline{\phi} \in \mathbb{R}^n$.

As pointed out in the introduction, the solution of nonlinear programming problems is usually carried out employing globalization strategies. Globalization strategies itself are strategies which provably compute sequences of iterates converging to critical points. As it turns out, "globalization" and "global convergence" refers to the independence of the convergence from the first iterate's quality. In other words, in the present thesis, we present different strategies, which compute critical points without assumptions on the initial iterate (except, that J is defined and the initial iterate is admissible).

As a consequence of the possible nonconvexity of the objective function, the about to be presented nonlinearly preconditioned globalization strategies will only aim at the computation of a first–order critical point which satisfies the following *first–order sufficiency conditions*

$$\overline{u} \in \mathcal{B} : \begin{cases} \nabla J(\overline{u})_i = 0 & \text{if } \underline{\phi}_i < u_i < \overline{\phi}_i \\ \nabla J(\overline{u})_i \leq 0 & \text{if } u_i = \overline{\phi}_i \\ \nabla J(\overline{u})_i \geq 0 & \text{if } u_i = \underline{\phi}_i \end{cases} \tag{1.1.1}$$

In fact, such a first–order critical point may be a local minimizer or a stationary point and can be computed under modest assumptions. In contrast, globalization strategies with second–order convergence properties state more restrictive assumptions on the search directions and on the objective functions (see, for instance, [CL96, CL94, CGT00]). In turn, we will see that due to the formulation of the nonlinearly preconditioned globalization strategies, only a global smoothing step may ensure convergence to such a point, but not the multiplicative or additive strategy, cf. Section 4.4 and Section 5.5.

Obviously, due to its general formulation, many classes of minimization problems may be covered by the model problem (M) and therefore, may be solved by the algorithms presented in this thesis, for instance problems arising from the field of nonlinear solid mechanics.

Due to its ability to predict internal stresses within solid materials, in the last decades numerical simulations of elastic materials became increasingly important. In particular, engineers employ elastostatic and elastodynamic simulations to cheaply verify designs in the load case. To this end, in the next section we will present nonlinear material laws which will serve as the objective functions within our numerical examples in Chapter 5.5.

1.2 The Constitutional Equations and their Discretization

In the context of continuum mechanical simulations, in contrast to molecular dynamics, the atomistic structure of a solid is neglected and just approximated within the model and its assumptions. Therefore, currently, *continuum mechanical* simulations enable engineers to simulate the behavior of solids on much larger time and length scales than employing *molecular dynamical* simulations. In the present thesis, we aim at the (dynamic) continuum mechanical simulation of a solid's reaction to large external forces basically following the monographs [Cia88, EGK08].

Therefore, in the presented model, all physical quantities, for instance, mass, linear momentum, velocity and energy, are considered as mean values. The body itself is a (time dependent) domain

$\Omega(t) \subset \mathbb{R}^3$ with Lipschitz continuous boundary, given in its reference configuration, i.e., in the undeformed state. Therefore, the objective is to find an energy optimal *deformation* subject to given external and internal forces. This *elastic (internal) energy* is, in turn, been given as a material law relating stresses to strains, such as, for instance, Hooke's law. Hooke's law, describes a linear stress strain relationship yielding a quadratic and coercive energy function, whose minimization can be carried out employing state–of–the–art linear and iterative solvers. On the other hand, various nonlinear material laws exist, which incorporate a nonlinear stress strain relationship implying, in turn, a nonlinear, possibly non-convex energy function.

1.2.1 Kinematics and Conservation Laws

In the context of elasticity theory, a deformation is a continuously differentiable, orientation preserving and invertible mapping $\boldsymbol{\varphi} : [t_0, t_{\text{end}}] \times \Omega(t_0) \to \mathbb{R}^3$. In turn, the current position of each point $\boldsymbol{x} \in \Omega_0 = \Omega(t_0)$ is given by $\boldsymbol{\varphi}(t, \boldsymbol{x}) \in \Omega(t)$. This sought-after deformation is supposed to be sufficiently smooth in order to solve the about to be presented systems of PDEs, for instance $\boldsymbol{\varphi}(t, \cdot) \in H^1(\Omega)$ for all $t \in [t_0, t_{\text{end}}]$.
Following [EGK08], the sought-after deformation $\boldsymbol{\varphi}$ is the result of the *conservation of impulse*, subject to a given elastic material law, and two kinds of force densities acting on the body $\Omega(t)$,

- volume force densities $\boldsymbol{F} : [t_0, t_{\text{end}}] \times \Omega(t) \to \mathbb{R}^3$, like for instance gravity
- surface force densities $\boldsymbol{f} : [t_0, t_{\text{end}}] \times \Gamma_N(t) \subset \partial\Omega(t) \to \mathbb{R}^3$, which will constitute the *Neumann* boundary conditions

In many cases also additional, pointwise constraints are added, for instance to model contact between Ω and a rigid obstacle, or constraints to the solution which gives rise to

$$\boldsymbol{\varphi}(t, \cdot) \cdot \boldsymbol{n}(t, \cdot) \leq \mathrm{Id} \cdot \boldsymbol{n}(t, \cdot) + \phi(t, \cdot) \text{ a.e. on } \Gamma_C(t) \subset \partial\Omega(t)$$

where $\phi : [t_0, t_{\text{end}}] \times \Gamma_C(t) \to \mathbb{R}^3$ and $\boldsymbol{n}(t, \boldsymbol{x})$ is the outer normal at $\boldsymbol{x} \in \Gamma_C$. Moreover, often also fixed displacements are applied to the volume Ω constituting *Dirichlet* boundary conditions, i.e.,

$$\boldsymbol{u}(t, \cdot) = \boldsymbol{g}(t, \cdot) \text{ a.e. on } \Gamma_D(t) \subset \partial\Omega(t)$$

where $\boldsymbol{g} : [t_0, t_{\text{end}}] \times \Gamma_D(t) \to \mathbb{R}^3$.
Given, the boundary conditions, we may now consider the conservation of impulse. In the reference configuration[1], the conservation of impulse is given by

$$\frac{d}{dt} \int_\Omega \rho \dot{\boldsymbol{u}} d\boldsymbol{x} = \int_\Omega \rho \boldsymbol{F} d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{f} \, ds_{\boldsymbol{x}} \tag{1.2.1}$$

where $\boldsymbol{u} = \boldsymbol{\varphi} + \mathrm{Id}$ are called *displacements*. Here, we introduced the mass density $\rho \in [t_0, t_{\text{end}}] \times L^2(\Omega)$. To obtain a better understanding how external forces induce internal stresses, we will follow [Cia88] and introduce a stress tensor $\hat{\boldsymbol{T}} : \Omega \to \mathbb{M}^3$, where $\mathbb{M}^3$ is the set of all 3×3 matrices. In particular, $\hat{\boldsymbol{T}}$ is an elastic *response function* which describes the stress-strain relationship for the material which is Ω made of. If one assumes that Cauchy's axiom (cf. Axiom 2.2-1 [Cia88]) holds,

[1] We implicitly consider all variables in the occurring PDEs as Lagrange variables and, thus, drop the time-dependency of the domain and its boundaries. Another point of view is to consider all variables as called *Euler* variables, variables defined in the (sought-after) deformed state.

(1.2.1) becomes

$$\frac{d}{dt}\int_\Omega \rho \dot{\boldsymbol{u}} d\boldsymbol{x} = \int_\Omega \rho \boldsymbol{F} d\boldsymbol{x} - \int_{\Gamma_N} \hat{\boldsymbol{T}} \cdot \boldsymbol{n}\, ds_{\boldsymbol{x}}$$

If we assume that ρ is constant in time, i.e, $\dot{\rho} = 0$, we may apply the divergence theorem giving rise to

$$\int_\Omega \left(\rho \ddot{\boldsymbol{u}} - \rho \boldsymbol{F} - \text{div}\, \hat{\boldsymbol{T}}\right) d\boldsymbol{x} = 0$$

Since this equality must also be satisfied on each subset Ω' of Ω we can deduce that

$$\rho \ddot{\boldsymbol{u}} - \rho \boldsymbol{F} - \text{div}\, \hat{\boldsymbol{T}} = 0 \text{ in } \Omega \tag{1.2.2a}$$
$$\hat{\boldsymbol{T}} \cdot \boldsymbol{n} = \boldsymbol{f} \text{ a.e. on } \Gamma_N \tag{1.2.2b}$$
$$\boldsymbol{u} = \boldsymbol{g} \text{ a.e. on } \Gamma_D \tag{1.2.2c}$$
$$\boldsymbol{u} \cdot \boldsymbol{n} \leq \phi \text{ a.e. on } \Gamma_C \tag{1.2.2d}$$

holds.

1.2.2 Elastodynamic and Elastostatic Model Problems in H^1

To obtain a complete description of the PDE, we must introduce a constitutional law for the response function. In our context, the context of large deformations, we are interested in the material's response on large deformations and we will, therefore, employ the theory of hyperelastic materials. For hyperelastic materials the following relationship holds

$$\hat{\boldsymbol{T}}(\boldsymbol{x}, \nabla \boldsymbol{\varphi}) = \frac{\partial}{\partial \boldsymbol{C}} \tilde{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{C})$$

Where $\tilde{\boldsymbol{W}} : \overline{\Omega} \times \mathbb{S}^3_> \to \mathbb{R}$ is a continuously differentiable *stored energy function* and $\boldsymbol{C}$ is the *right Cauchy-Green strain tensor* given by

$$\boldsymbol{C} = (\nabla \boldsymbol{u} + I)^T (\nabla \boldsymbol{u} + I) = \nabla \boldsymbol{\varphi}^T \nabla \boldsymbol{\varphi} \in \mathbb{S}^3_>$$

with $\mathbb{S}^3_>$ is the set of all symmetric positive definite 3×3 matrices. Now we can combine the boundary conditions and the initial conditions with the derived constitutional law and obtain the following system of PDEs

$$\rho \ddot{\boldsymbol{u}} + \text{div} \frac{\partial}{\partial \boldsymbol{C}} \tilde{\boldsymbol{W}} - \boldsymbol{F} = 0 \qquad \text{a.e. in } \Omega \tag{1.2.3a}$$
$$\hat{\boldsymbol{T}} \cdot \boldsymbol{n} = \boldsymbol{f} \qquad \text{a.e. on } \Gamma_N \subset \partial\Omega \tag{1.2.3b}$$
$$\boldsymbol{u} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C \subset \partial\Omega \tag{1.2.3c}$$
$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{a.e. on } \Gamma_D \tag{1.2.3d}$$

Here, we suppose that $\Gamma_C \cup \Gamma_D \cup \Gamma_N = \partial\Omega$, $\Gamma_C, \Gamma_N, \Gamma_D$ are pairwise disjoint and $\phi \in L^2(\Gamma_C)$. The initial displacements $\boldsymbol{u}_0 \in H^1(\Omega)$ and velocity $\dot{\boldsymbol{u}}_0 \in L^2(\Omega)$ are assumed to be given a priori which gives rise to the following initial conditions

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0(\boldsymbol{x}) \qquad \text{in } \Omega \tag{1.2.4a}$$
$$\dot{\boldsymbol{u}}(\boldsymbol{x}, 0) = \dot{\boldsymbol{u}}_0(\boldsymbol{x}) \qquad \text{a.e. in } \Omega \tag{1.2.4b}$$

For the analysis presented in the remaining sections of this chapter, it will not make a difference if ρ is chosen constant in space. Therefore, in the remainder, we will assume without loss of generality that $\rho \equiv 1$.
Moreover, if in the *static border case* the time derivatives vanish, the system of PDEs (1.2.3) becomes

$$\operatorname{div} \frac{\partial}{\partial \boldsymbol{C}} \tilde{\boldsymbol{W}} - \boldsymbol{F} = 0 \qquad \text{a.e. in } \Omega \tag{1.2.5a}$$
$$\hat{\boldsymbol{T}} \cdot \boldsymbol{n} = \boldsymbol{f} \qquad \text{a.e. on } \Gamma_N \subset \partial\Omega \tag{1.2.5b}$$
$$\boldsymbol{u} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C \subset \partial\Omega \tag{1.2.5c}$$
$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{a.e. on } \Gamma_D \subset \partial\Omega \tag{1.2.5d}$$

Here, we also assume that Γ_D, Γ_N and Γ_C are pairwise disjoint and that $\Gamma_D \cup \Gamma_N \cup \Gamma_C = \partial\Omega$.

An Elasto-Static Minimization Problem and Existence of Solutions

If we now assume that $\boldsymbol{f}$ and $\boldsymbol{F}$ are independent of $\boldsymbol{u}$, Theorem 4.1-1 [Cia88] gives us that (1.2.5) is formally equivalent to the following constrained minimization problem: find a $\boldsymbol{u} \in H^1(\Omega)$ which solves

$$J(\boldsymbol{u}) = \min! \tag{1.2.6a}$$
$$\boldsymbol{u} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C \subset \partial\Omega \tag{1.2.6b}$$
$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{a.e. on } \Gamma_D \subset \partial\Omega \tag{1.2.6c}$$

Here, we have introduced the *nonlinear energy functional* consisting of the elastic (internal) energy and external work as

$$J(\boldsymbol{u}) = \int_\Omega \left(\tilde{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{C}) - \rho \boldsymbol{F} \cdot \boldsymbol{u} \right) d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f} \cdot \boldsymbol{u} \, ds_{\boldsymbol{x}}$$

Remark 1.2.1. *Here $J(\boldsymbol{u}) = \min!$ denotes the local minimizer over the set of all $\tilde{u}$ satisfying (1.2.6b) and (1.2.6c). For the ease of notation, in the remainder of this thesis we will keep to this notation and will not explicitly highlight, that we are interested in a local solution.*

Since the energy J is arbitrarily nonlinear and, thus, possibly non-convex, solutions of the minimization problem (1.2.6) generally cannot be shown to exist. Therefore, we follow [Cia88] and J. Ball [Bal77] and introduce stronger assumptions on the stored energy function $\tilde{\boldsymbol{W}}$ to ensure the existence of minimizers.

Definition 1.2.2. *A stored energy function $\tilde{\boldsymbol{W}}$ is* **polyconvex** *if there exists for each $\boldsymbol{x} \in \overline{\Omega}$ a convex function $\mathbb{W} : \mathbb{M}^3 \times \mathbb{M}^3 \times (0, \infty) \to \mathbb{R}$ such that*

$$\tilde{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{C}) = \widehat{\boldsymbol{W}}(\boldsymbol{x}, \nabla\boldsymbol{\varphi}) = \mathbb{W}(\boldsymbol{x}, \nabla\boldsymbol{\varphi}, \operatorname{Cof}\nabla\boldsymbol{\varphi}, \det\nabla\boldsymbol{\varphi}) \qquad \forall \boldsymbol{\varphi} \in \mathbb{M}^3_+$$

where $\operatorname{Cof} A = \det A A^{-T}$. We will call a polyconvex stored energy function **coercive** *if there exists an $\alpha > 0$ and an $\beta \in \mathbb{R}$ such that*

$$\tilde{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{C}) = \widehat{\boldsymbol{W}}(\boldsymbol{x}, \nabla\boldsymbol{\varphi}) \geq \alpha \left(\|\nabla\boldsymbol{\varphi}\|^2 + \|\operatorname{Cof}\nabla\boldsymbol{\varphi}\|^2 + (\det\nabla\boldsymbol{\varphi})^2 \right) + \beta \text{ a.e. in } \Omega \text{ and all } \nabla\boldsymbol{\varphi} \in \mathbb{M}^3_+$$

Now, we may cite Theorem 7.8-1 [Cia88] which provides the existence of minimizers for problem (1.2.6) under certain assumptions on the stored energy function and the respective problem setting.

Theorem 1.2.3. *Assume that $\widehat{\boldsymbol{W}}$ is a polyconvex, coercive stored energy function with*

$$\lim_{\det(\mathrm{I}+\nabla \boldsymbol{u})\searrow 0} \widehat{\boldsymbol{W}}(\boldsymbol{x}, \mathrm{I}+\nabla \boldsymbol{u}) = \infty$$

Let $\Gamma_D, \Gamma_N, \Gamma_C$ be disjoint, relatively open subsets of $\partial\Omega$, $\Gamma_D \neq \emptyset$ and $\partial\Omega - \Gamma_D \cup \Gamma_C \cup \Gamma_N = \emptyset$. Assume that the following set of energy-admissible solutions

$$\begin{aligned} \Phi = \{\boldsymbol{u} \in H^1(\Omega) \quad | \quad & \mathrm{Cof}(\mathrm{I}+\nabla \boldsymbol{u}) \in L^2(\Omega), 0 < \det(\mathrm{I}+\nabla \boldsymbol{u}) \in L^2(\Omega), \\ & \boldsymbol{u} = \boldsymbol{g} \qquad \text{a.e. on } \Gamma_D, \\ & \boldsymbol{u} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C\} \end{aligned}$$

is non-empty. Let

$$L(\boldsymbol{u}) = -\int_\Omega \boldsymbol{F} \cdot \boldsymbol{u}\, d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f} \cdot \boldsymbol{u}\, ds_{\boldsymbol{x}}$$

be in $(H^1(\Omega))'$. Moreover, assume that there exists an $\tilde{\boldsymbol{u}} \in \Phi$ such that

$$\inf_{\tilde{\boldsymbol{u}} \in \Phi} I(\tilde{\boldsymbol{u}}) = \inf_{\tilde{\boldsymbol{u}}} \int_\Omega \widehat{\boldsymbol{W}}(\boldsymbol{x}, \mathrm{I}+\nabla \tilde{\boldsymbol{u}})\, d\boldsymbol{x} + L(\tilde{\boldsymbol{u}}) < +\infty$$

then there exists at least one function such that

$$\inf_{u \in \Phi} I(\boldsymbol{u}) \in (-\infty, +\infty)$$

In fact, this theorem directly applies to elastostatic problems of the kind (1.2.6) and provides sound assumptions on the problem itself, such as the relationship between constraints and Dirichlet values, and assumptions on the surface forces.

1.3 Discretization

A key role in the solution of the system of PDEs in (1.2.3) and (1.2.6) is their discretization. As it turns out, we will apply the globalization strategies presented in the following chapters to solve discretized minimization problems instead of the respective original system.
The discretization of a dynamic system of PDEs like (1.2.3) is usually carried out in two steps:

- discretization in time
- discretization in space

Here, one distinguishes between the discretization in time prior to space, called *Rothe's method*, or vice versa, called *method of lines*. In the method of lines the spatial discretization is chosen fixed for all time steps, which does not allow for a better resolution of time dependent spatial phenomena. In contrast, Rothe's method allows for choosing different spatial discretization depending on the current time step. Moreover, discretizing (1.2.3) in time gives rise to spatial minimization problems. In turn, in each time step, under similar assumptions as for Theorem 1.2.3, we are able to prove the existence of solutions for these minimization problems.

1.3.1 Temporal Discretization

Our aim is the computation of displacements $\boldsymbol{u}$ and velocities $\dot{\boldsymbol{u}}$ which solve the system of PDEs (1.2.3) at time t. To this end we follow [DKE08] and employ *Newmark's Scheme* [New59] to discretize (1.2.3) in time. Since the acceleration is already given by Newton's law, this integration scheme enables us to derive additional equations for the velocities and displacements. In fact, integrating the acceleration term twice yields the sought-after equations (for a complete introduction see, for instance, [Wri08]). Moreover, Newmark's scheme allows for introducing a contact stabilization to avoid artificial oscillations at the contact boundary [DKE08]. The basic principle of the contact stabilized Newmark method is to compute an additional predictor step. This means, that one employs the displacements and velocities of the previous timestep and the obstacle to compute predicted displacements, already satisfying the contact conditions (1.2.3c). In a second step, one employs these predicted displacements to compute the actual displacements.
For the ease of notation, we now denote by $\boldsymbol{u}^i$ the temporal discretized displacements. But note that the discretized solution generally does not satisfy $\boldsymbol{u}^i = \boldsymbol{u}(t_i)$ where $t_i = t_0 + i\tau$ and $\tau > 0$. Similarly, $\dot{\boldsymbol{u}}^i$ denotes the temporal discretization of the velocity at t_i.
In order to derive the contact stabilized scheme, we introduce for a given predictor $\boldsymbol{u}_{\text{pred}}^{i+1} \in L^2(\Omega)$ the following functional

$$\mathcal{I}^{i+1}(\boldsymbol{u}) = \frac{1}{2}(\boldsymbol{u}, \boldsymbol{u})_{L^2(\Omega)} - (\boldsymbol{u}, \boldsymbol{u}_{\text{pred}}^{i+1})_{L^2(\Omega)}$$

where $(\cdot,\cdot)_{L^2(\Omega)}$ denotes the L^2 scalar product. The temporal discretized energy functional is given by

$$J^i(\boldsymbol{u}) = \int_\Omega \left(\widehat{\boldsymbol{W}}(I + \nabla \boldsymbol{u}) - \boldsymbol{F}^i \cdot \boldsymbol{u} \right) \, d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f}^i \cdot \boldsymbol{u} \, ds_{\boldsymbol{x}}$$

where $\boldsymbol{F}^i = \boldsymbol{F}(t_i)$, $\boldsymbol{f}^i = \boldsymbol{f}(t_i)$. Now, we can introduce the contact stabilized Newmark method as the following system of PDEs: Find $\boldsymbol{u}^{i+1} \in H^1(\Omega)$, $\boldsymbol{u}_{\text{pred}}^{i+1}$ and $\dot{\boldsymbol{u}}^{i+1} \in L^2(\Omega)$ such that

$$\left((\frac{1}{2}\boldsymbol{u}_{\text{pred}}^{i+1} - \boldsymbol{u}^i - \tau \dot{\boldsymbol{u}}^i), \boldsymbol{u}_{\text{pred}}^{i+1} \right)_{L^2(\Omega)} = \min! \tag{1.3.1a}$$

$$\mathcal{I}^{i+1}(\boldsymbol{u}^{i+1}) + \frac{\tau^2}{2}(1-2\beta) J^i(\boldsymbol{u}^i) + \tau^2 \beta J^{i+1}(\boldsymbol{u}^{i+1}) = \min! \tag{1.3.1b}$$

$$\dot{\boldsymbol{u}}^i + \tau \left((1-\gamma) \frac{\partial}{\partial \boldsymbol{u}} J^i(\boldsymbol{u}^i) + \gamma \frac{\partial}{\partial \boldsymbol{u}} J^{i+1}(\boldsymbol{u}^{i+1}) \right) = \dot{\boldsymbol{u}}^{i+1} \tag{1.3.1c}$$

$$\boldsymbol{u}_{\text{pred}}^{i+1} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C \tag{1.3.1d}$$

$$\boldsymbol{u}^{i+1} \cdot \boldsymbol{n} \leq \phi \qquad \text{a.e. on } \Gamma_C \tag{1.3.1e}$$

$$\boldsymbol{u}^{i+1} = \boldsymbol{g} \qquad \text{a.e. on } \Gamma_D \tag{1.3.1f}$$

where $\gamma, 2\beta \in [0,1]$. The initial conditions are given by

$$\boldsymbol{u}^0(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x}) \qquad \text{a.e. in } \Omega \tag{1.3.2a}$$

$$\dot{\boldsymbol{u}}^0(\boldsymbol{x}) = \dot{\boldsymbol{u}}_0(\boldsymbol{x}) \qquad \text{a.e. in } \Omega \tag{1.3.2b}$$

In the unconstrained case, Newmark's scheme becomes *energy, linear momentum and angular momentum* preserving if one chooses the constants β, γ as $2\beta = \gamma = \frac{1}{2}$ (cf., for instance [ST92]). As a matter of fact, for $2\beta \geq \gamma \geq \frac{1}{2}$ the scheme becomes unconditionally stable. Thus, in order to make

the scheme in the unconstrained case unconditionally stable, one has to solve an arbitrary nonlinear, possibly non-convex, minimization problem in each time step.

Existence of Solutions for the Discretized Dynamical Problem

We will now consider the existence of a local minimizer for the temporally discretized, constrained minimization problem (1.3.1b). The key concept of this theorem is to reorder the objective function in (1.3.1b) to obtain the actual nonlinear energy function H, a linear form $L^i$and a constant part J^i and to prove that under certain assumptions the total energy can be bounded.

Theorem 1.3.1. *Assume that $\widehat{\boldsymbol{W}}$ is a polyconvex, coercive stored energy function with*

$$\lim_{\det(\mathrm{I}+\nabla \boldsymbol{u})\searrow 0} \widehat{\boldsymbol{W}}(\boldsymbol{x}, \mathrm{I}+\nabla \boldsymbol{u}) = \infty$$

Let $\Gamma_N, \Gamma_C, \Gamma_D$ be disjoint, relatively open subsets of $\partial\Omega$, and $\partial\Omega - \Gamma_C \cup \Gamma_N \cup \Gamma_D = \emptyset$. Assume that for a given obstacle $\phi \in L^2(\Gamma_C)$ the set of energy admissible solutions

$$\begin{aligned} \Phi = \{\boldsymbol{u} \in H^1(\Omega) \quad | \quad & \mathrm{Cof}(\mathrm{I}+\nabla \boldsymbol{u}) \in L^2(\Omega), 0 < \det(\mathrm{I}+\nabla \boldsymbol{u}) \in L^2(\Omega), \\ & \boldsymbol{u}\cdot\boldsymbol{n} \leq \phi \qquad \textit{a.e. on } \Gamma_C \\ & \boldsymbol{u} = \boldsymbol{g} \qquad \textit{a.e. on } \Gamma_D\} \end{aligned}$$

is non-empty. For $\tau > 0$, define the modified energy by

$$H(\boldsymbol{u}) = \int_\Omega \left(\frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u} + \tau^2 \beta \widehat{\boldsymbol{W}}(I+\nabla \boldsymbol{u}) \right) d\boldsymbol{x}$$

Let $\boldsymbol{u}^i \in H^1(\Omega)$, $\boldsymbol{u}^{i+1}_{pred}$, $\boldsymbol{F}^i$, $\boldsymbol{F}^{i+1} \in L^2(\Omega)$, $\boldsymbol{f}^i$, $\boldsymbol{f}^{i+1} \in L^2(\Gamma_N)$, and $J^i(\boldsymbol{u}^i) \in \mathbb{R}$ be given and define

$$\begin{aligned} L^i(\boldsymbol{u}) \quad = \quad & \int_\Omega \left(-\boldsymbol{u}^{i+1}_{pred}\cdot\boldsymbol{u} - \left(\frac{\tau^2}{2}(1-2\beta)\boldsymbol{F}^i + \tau^2\beta\boldsymbol{F}^{i+1} \right)\cdot\boldsymbol{u} \right) d\boldsymbol{x} - \\ & \int_{\Gamma_N} \left(\frac{\tau^2}{2}(1-2\beta)\boldsymbol{f}^i + \tau^2\beta\boldsymbol{f}^{i+1} \right)\cdot\boldsymbol{u} ds_{\boldsymbol{x}} \end{aligned}$$

Therefore, L^i is a linear form on $H^1(\Omega)$. Suppose that there exists an $\tilde{\boldsymbol{u}}$ such that

$$\inf_{\tilde{\boldsymbol{u}}} I(\tilde{\boldsymbol{u}}) = H(\tilde{\boldsymbol{u}}) + L^i(\tilde{\boldsymbol{u}}) + \frac{\tau^2}{2}(1-2\beta)J^i(\boldsymbol{u}^i) < +\infty$$

then there exists at least one function $\boldsymbol{u} \in \Phi$ such that

$$\inf_{\boldsymbol{u}\in\Phi} I(\boldsymbol{u}) \in (-\infty, +\infty)$$

Proof. Due to the coercivity of $\widehat{\boldsymbol{W}}$ we obtain that there exist constants $\alpha > 0$, $\gamma > 0$, $c_0 \in \mathbb{R}$ such that

$$\widehat{\boldsymbol{W}}(\boldsymbol{x}, \nabla \boldsymbol{u}) \geq \alpha \|I + \nabla \boldsymbol{u}\|^2_{L^2(\Omega)} + \gamma \mathrm{vol}\Omega + c_0$$

(cf. Theorem 7.7.-1 (i) [Cia88]). Moreover, employing the triangle-inequality we obtain

$$c_1 + (\boldsymbol{u}, \boldsymbol{u})_{L^2(\Omega)} \geq \|\mathrm{Id}\|^2_{L^2(\Omega)} + \|\boldsymbol{u}\|^2_{L^2(\Omega)} \geq \|\mathrm{Id} + \boldsymbol{u}\|^2_{L^2(\Omega)}$$

where $c_1 > 0$. Now, we employ $L^i(\boldsymbol{u}) = L^i(\boldsymbol{u} + \mathrm{Id}) - L^i(\mathrm{Id})$ and the continuity of L^i and obtain

$$\begin{aligned} I(\boldsymbol{u}) &= H(\boldsymbol{u}) + L^i(\boldsymbol{u}) + \frac{\tau^2}{2}(1-2\beta)J^i(\boldsymbol{u}^i) \\ &= \frac{1}{2}(\boldsymbol{u},\boldsymbol{u})_{L^2(\Omega)} + \int_\Omega \tau^2\beta\widehat{\boldsymbol{W}}(I+\nabla\boldsymbol{u})\,d\boldsymbol{x} + L^i(\boldsymbol{u}+\mathrm{Id}) - L^i(\mathrm{Id}) + \frac{\tau^2}{2}(1-2\beta)J^i(\boldsymbol{u}^i) \\ &\geq \alpha_2\frac{1}{2}|I+\nabla\boldsymbol{u}|^2_{H^1(\Omega)} + \frac{1}{2}\|\mathrm{Id}+\boldsymbol{u}\|^2_{L^2(\Omega)} - \|L^i\|^2_{L^2(\Omega)}\|\mathrm{Id}+\boldsymbol{u}\|_{L^2(\Omega)} + c_2 \end{aligned}$$

where $c_2 = -\frac{1}{2}c_1 + \frac{\tau^2}{2}(1-2\beta)J^i(\boldsymbol{u}^i) - L^i(\mathrm{Id}) + \tau^2\beta(\gamma\mathrm{vol}\Omega + c_0)$ and $\alpha_2 = \tau^2\beta\alpha$. This, in particular, yields

$$I(\boldsymbol{u}) \geq c_3\|I+\nabla\boldsymbol{u}\|^2_{H^1(\Omega)} - c_4\|\mathrm{Id}+\boldsymbol{u}\|_{L^2(\Omega)} + c_2 \tag{1.3.3}$$

where $c_3 > 0$, $c_4 \in \mathbb{R}$. Thus, we obtain for each sequence $(\boldsymbol{\varphi}^k)_k$ with $\boldsymbol{\varphi}_k \in \Phi$ and $\lim\limits_{k\to\infty}\|\boldsymbol{\varphi}^k\|_{H^1(\Omega)} = \infty$ (if such a sequence exists) that

$$\liminf_{k\to\infty}\frac{I(\boldsymbol{\varphi}^k)}{\|\boldsymbol{\varphi}^k\|_{H^1(\Omega)}} > 0$$

Thus, any minimizing sequence of the total energy is necessarily bounded in Φ. Now the further proof is the one of Theorem 7.7.-1 (iii-vi) [Cia88]. □

As it turns out, if one assumes that $\mathcal{B}$ is given like in (M) the proof of Theorem 7.8-2 [Cia88] directly applies to our Newmark scheme, since no "directions where the body can escape" occur[2]. On the other hand, even if such a direction exists, the L^2 scalar product in (1.3.1a) directly yields the strong coercivity result in (1.3.3), even if no Dirichlet values are set. Thus, we have just proven that for dynamical (contact) problems the assumption that no escape direction exists is not necessary.

1.3.2 Spatial Discretization

Several spatial discretization schemes, such as Finite Volumes (see, e.g., [Bra07]), Finite Elements (see, e.g., [Bra07, Lev02]), Finite Differences and Wavelets (see, e.g., [Dah97]) as well as Meshfree Methods (see, e.g., [Liu03]), are frequently used to solve PDEs on finite domains such as Ω. Today, Finite Elements and Finite Differences particularly prevailed as state-of-the-art discretization techniques for problems arising from the field of elasticity. Finite Elements and its variants, moreover, prevailed as a spatial discretization technique for complex, possibly CAD based, geometries, which may not be accurately enough resolved by Finite Differences.

The paradigm of Finite Elements is to discretize and approximate the domain Ω by a three dimensional, polyhedral meshed domain Ω_h. In many examples, Ω is a CAD based, polyhedral geometry. In this case one can assume that $\Omega_h = \Omega$ and that Ω is polygonally bounded. In order to (adaptively) resolve time dependent spatial phenomena, Rothe's method enables us to choose the piecewise polynomial Finite Element *basis functions* time dependently as $\boldsymbol{\lambda}^i_p : \Omega \to \mathbb{R}^3$ yielding the Finite Element space $\mathcal{X}^i = \mathrm{span}\{\lambda^i_p\}_p$.

[2] This is a theoretical problem for traction problems with unilateral boundary conditions of place. If the applied forces press Ω against the obstacle, a solution exists. Forces pulling Ω away yield the insolvability of the problem.

With the definition of the Finite Element basis, each element $\boldsymbol{u}_h \in \mathcal{X}^i$ can be represented as

$$\boldsymbol{u}_h(\boldsymbol{x}) = \sum_p u_p^i \boldsymbol{\lambda}_p^i(\boldsymbol{x})$$

where the coefficient vector is given by $u_h^i = (u_p^i)_p \in \mathbb{R}^{n_i}$ and $n_i = \dim \mathcal{X}^i$. In other words, in each time-step t_i there exists a coordinate isomorphism $X^i : \mathbb{R}^{n_i} \to \mathcal{X}^i$ given by

$$X^i = \left(\boldsymbol{\lambda}_0^i, \ldots, \boldsymbol{\lambda}_{n_i}^i\right)$$

Similarly, one may also define spatial discretized velocities $\dot{u}_h^i$ and predictors $u_{h,\text{pred}}^i$.

Constraints and Nodal Basis Functions

For applying nonsmooth iterative solvers, like for instance the preconditioned projected cg method, to solve PDEs or optimization problems subject to constraints it is preferable that the constraints are pointwise. In our case, we have to deal with generally coupled constraints given by (1.3.1d) and (1.3.1e). Therefore, to apply the preconditioned projected cg method, we must alter the standard discretization slightly. In particular, at the contact boundary Γ_C the (three) nodal basis functions must be chosen such that one basis function directs in direction $\boldsymbol{n}$ and the other basis function are orthogonal to the first one [Kra01]. Therefore, in the remainder we will assume that one is able to choose the discretization such that

$$\boldsymbol{u}_h \cdot n = \boldsymbol{u}_h \leq \boldsymbol{g}_h \text{ on } \Gamma_C$$

for a given $\boldsymbol{g}_h \in \mathcal{X}$ holds.

The Temporal and Spatial Discretized Minimization Problems

Employing this isomorphism, we can now reformulate the temporal discretized system of PDEs (1.3.1a) into a fully discretized finite dimensional system of PDEs: find $u_h^{i+1}, u_{h,\text{pred}}^{i+1}, \dot{u}_h^i \in \mathbb{R}^n$ which solve

$$\frac{1}{2}\left(u_{h,\text{pred}}^{i+1}\right)^T M^{i+1} u_{h,\text{pred}}^{i+1} - (u_h^i - \tau \dot{u}_h^i)^T M_i^{i+1} u_{h,\text{pred}}^{i+1} = \min! \tag{1.3.4a}$$

$$\left(\frac{1}{2}u_h^{i+1} - u_{h,\text{pred}}^{i+1}\right)^T M^{i+1} u_h^{i+1} + \frac{\tau^2}{2}(1-2\beta) J(X^i u_h^i) - 2\beta J(X^{i+1} u_h^{i+1}) = \min! \tag{1.3.4b}$$

$$X^i \dot{u}_h^i + \tau\left((1-\gamma)\frac{\partial}{\partial \boldsymbol{u}} J^i(X^i u_h^i) + \gamma \frac{\partial}{\partial \boldsymbol{u}} J^{i+1}(X^{i+1} u_h^{i+1})\right) = X^{i+1}\dot{u}_h^{i+1} \tag{1.3.4c}$$

$$X^{i+1} u_{h,\text{pred}}^{i+1} \leq \phi_h^{i+1} \quad \text{on } \Gamma_C \tag{1.3.4d}$$

$$X^{i+1} u_h^{i+1} \leq \phi_h^{i+1} \quad \text{on } \Gamma_C \tag{1.3.4e}$$

$$X^{i+1} u_h^{i+1} = \boldsymbol{g}_h^{i+1} \quad \text{on } \Gamma_D \tag{1.3.4f}$$

where $\boldsymbol{g}_h^{i+1} : X_{i+1} \to \mathbb{R}^3$ is given and $\phi_h^{i+1} \in \mathcal{X}_{i+1}$ is an approximation to ϕ in $\mathcal{X}_{i+1}$. In general one might define both functions as the solution of

$$\begin{aligned} \boldsymbol{g}_h^{i+1} : (\boldsymbol{g}_h^{i+1}, \lambda_p^{i+1})_{L^2(\Omega)} &= (\boldsymbol{g}(t_{i+1}), \lambda_p^{i+1})_{L^2(\Omega)} \qquad \forall p = 1, \dots, n_{i+1} \\ \phi_h^{i+1} : (\phi_h^{i+1}, \lambda_p^{i+1})_{L^2(\Omega)} &= (\phi, \lambda_p^{i+1})_{L^2(\Omega)} \qquad \forall p = 1, \dots, n_{i+1} \end{aligned}$$

Moreover, we employed the mass matrices

$$M^{i+1} = (X^{i+1})^T X^{i+1} = \left(\int_\Omega \boldsymbol{\lambda}_p^{i+1}(\boldsymbol{x}) \boldsymbol{\lambda}_q^{i+1}(\boldsymbol{x}) \, d\boldsymbol{x} \right)_{pq} \in \mathbb{R}^{n_{i+1} \times n_{i+1}} \tag{1.3.5a}$$

$$M_i^{i+1} = (X^i)^T X^{i+1} = \left(\int_\Omega \boldsymbol{\lambda}_p^{i}(\boldsymbol{x}) \boldsymbol{\lambda}_q^{i+1}(\boldsymbol{x}) \, d\boldsymbol{x} \right)_{pq} \in \mathbb{R}^{n_{i} \times n_{i+1}} \tag{1.3.5b}$$

For given $\boldsymbol{u}^0, \dot{\boldsymbol{u}}^0 \in \mathbb{R}^n$ the initial conditions are

$$u_h^0 = u^0 \qquad \text{on } \mathbb{R}^n \tag{1.3.6a}$$

$$\dot{u}_h^0 = \dot{u}^0 \qquad \text{on } \mathbb{R}^n \tag{1.3.6b}$$

As it turns out, the computation of the mass matrices changes slightly if the mass ρ is not constant in space which can easily be implemented in practice. On the other hand, the computation of M_i^{i+1} is more challenging in practice, particularly for strongly changing Finite Element spaces. Here, one must put effort in computing the quadrature in (1.3.5).
In the absence of time, the discretization of the resulting minimization problem (1.2.6) can be carried out straight-forwardly. We suppose that a Finite Element space $\mathcal{X} \subset H^1(\Omega)$ along with the isomorphism $X : \mathbb{R}^n \to \mathcal{X}$ is given. Then the discretized version of (1.2.6) is given by: find a $u \in \mathbb{R}^n$ such that

$$J(Xu) = \min! \tag{1.3.7a}$$

$$Xu \leq \phi_h \qquad \text{on } \Gamma_C \subset \partial\Omega \tag{1.3.7b}$$

$$Xu = g_h \qquad \text{on } \Gamma_D \subset \partial\Omega \tag{1.3.7c}$$

for given $g_h, \phi_h \in \mathcal{X}$. Here, the constraints should be understood pointwise.
In many cases, $\hat{\boldsymbol{T}}$ is chosen by means of Hooke's law yielding a strictly convex quadratic optimization problem in (1.3.4b) and (1.3.7a), respectively. But, in our context, even if the response function $\hat{\boldsymbol{T}}$ is chosen based on a polyconvex and coercive stored energy function, the resulting minimization problems are generally non-convex. Thus, the solution of these problems must be carried out employing a globalization strategy to provably succeed in computing a local minimizer. To this end, in the following chapters we will present different traditional globalization strategies and introduce novel nonlinearly preconditioned globalization strategies which allow for the solution of nonlinear programming problems like (1.3.4b) and (1.3.7a).

2 State of the Art Globalization Strategies

The solution of minimization problems like problem (M) is usually carried out employing iterative schemes like, for instance, Newton's method. Though, for a nonconvex objective function J Newton's method is not globally convergent, which means that in order to prove convergence the initial iterate must be assumed to be sufficiently close to a local solution. Therefore, to ensure convergence for arbitrary initial iterates the solution of (M) must be carried out employing a globalization strategy. In contrast to the direct application of Newton's method, a globalization strategy aims at a scaling of the current Newton correction's length to enforce convergence. But, in fact, computing and scaling a correction based on Newton's method does not suffice to generate *descending search directions*. Therefore, two major problems had to be attacked to develop a globally convergent solution strategy for problem (M):

- the computation of search directions
- the computation of damping parameters

such that the resulting correction, i.e., the rescaled search direction, induces a sufficient decrease of the objective function J.
The first contribution attacking the problem of computing "good" search directions was made by K. Levenberg in the context of the solution of nonlinear least squares problems [Lev44]. Also D. D. Morrisson [Mor60] considered the solution of quadratic minimization problems which in his formulation were surprisingly solved subject to fixed step–length constraints. Further contributions in this field were made by D. W. Marquardt [Mar63], R. E. Griffith and R. A. Stewart [GS61]. Griffith and Stewart proposed to successively solve linear problems based on the objective function.
In the early stages of globalization strategies the correction's step–length and the model were connected to each other by means of a damping of the Hessian. A first update strategy for the damping parameter (which was by then chosen fixed) was provided in [GQT66] which was further developed by M. Powell [Pow70] and D. Winfield [Win73]. Basically, Goldfeldt et al. [GQT66] developed the update strategy which is employed in today's Trust-Region methods. Here, the quadratic model is employed to compute a *predicted reduction* which is compared to the *actual reduction* of the objective function. Moreover, since the late 1960s and early 1970s quadratic model functions for the successive computation of search directions prevailed and the *Cauchy point*[1] was developed as a measure for sufficient decrease [Pow70].
On the other hand, in the 1940s H. B. Curry [Cur44], and in the 1960s A. A. Goldstein [Gol62] and L. Armijo [Arm66] formulated assumptions on a damping parameter to enforce convergence of the steepest descent method. In their articles, Goldstein and Armijo proposed criteria for controlling the step–length and were, due to their formulation of a sufficient decrease, able to show convergence to first–order critical points.
As it turns out, both frameworks, the Linesearch and Trust-Region framework, are closely related to each other. The fact that Linesearch methods can be regarded as a special case of Trust-Region

[1] The Cauchy point (2.1.10) is the optimally scaled negative gradient which may be employed as sufficiently good search direction.

methods was published in [SSB85] and [Toi88]. However, both strategies particularly differ in the step–length control strategy. We will see that, on the one hand, Trust-Region methods employ an a priori control of the step–length and an a posteriori acceptance criterion. On the other hand, Linesearch strategies employ an a posteriori step–length control and accept each correction.
In this chapter, we will present non–smooth Trust-Region and Linesearch frameworks for the solution of pointwise constrained minimization problems of the kind (M) and will prove the convergence of the respective strategy. These strategies will serve as a basis of the preconditioned Trust-Region and Linesearch methods, introduced in Chapter 4 and Chapter 5. Therefore, our analysis of this chapter will also be employed for showing local convergence properties of the preconditioned globalization strategies.

2.1 The "Traditional" Trust-Region Framework

Similar to Newton's method, a Trust-Region method is an iterative solution strategy, which in each iteration computes a *correction* and chooses whether to apply the correction or not. The correction itself is the (approximate) solution of a constrained quadratic minimization problem, a second–order approximation to the expected reduction of the objective function J from (M). The quadratic function, called *Trust-Region model*, consists of the gradient and the Hessian, or in general, a symmetric matrix which approximates or replaces the Hessian. This is in particular reasonable, if the Hessian is dense or negative definite. For instance, the BFGS method [Bro70, Fle70, Gol70, Sha70] is a frequently used strategy to directly approximate the inverse of the Hessian. Here, expensive quadrature can be avoided by an update just based on the computed gradients.
As it will turn out, not every computed correction, even if it is an exact solution of the quadratic minimization problem, is added to the iterate. In particular, only if the ratio between the actual decrease and the decrease predicted by the Trust-Region model is sufficiently large, the correction is applied. This in turn, is an a posteriori control which guarantees convergence to first–order critical points, i.e., the solution of (1.1.1) . All together this yields Algorithm 1.

2.1.1 Assumptions on J and the Trust-Region Model

Surprisingly, the convergence theory of Trust-Region methods can be carried out with modest assumptions [CGT00]. In the present thesis, we will follow T. Coleman and Y. Li [CL96] and suppose that the gradient of the objective function $J : \mathbb{R}^n \to \mathbb{R}$ in (M) is bounded on a compact level set. Moreover, one has to state assumptions on the Hessian, or its approximation.

($A_{tr}1$) For the given initial iterate $u_0 \in \mathcal{B}$ from (M), we assume that the level set

$$\mathcal{L} = \{u \in \mathcal{B} \mid J(u) \leq J(u_0)\}$$

is compact.

($A_{tr}2$) We assume that J is continuously differentiable on $\mathcal{L}$. Moreover, on $\mathcal{L}$, the gradients are assumed to be bounded by a constant $C_g > 0$, i.e., $\|\nabla J(u)\|_2 \leq C_g$ for all $u \in \mathcal{L}$.

($A_{tr}3$) There exists a constant $C_B > 0$ such that for all iterates $u \in \mathcal{L}$ and for each symmetric matrix $B(u)$ approximating $\nabla^2 J(u)$ the inequality $\|B(u)\|_2 \leq C_B$ is satisfied.

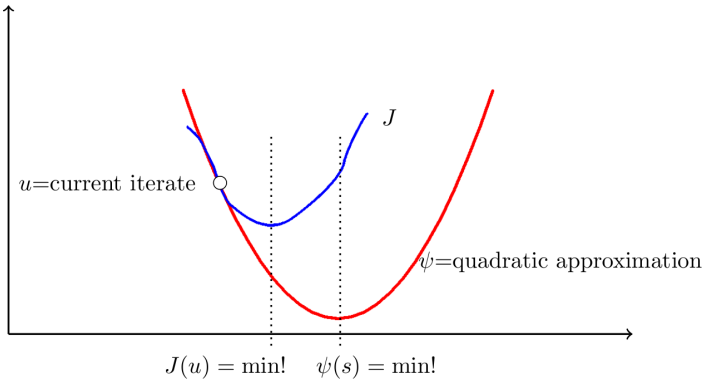

Figure 2.1: A comparison of a highly nonlinear objective function and the quadratic model (illustrated is $\psi(s) + J(u)$). Obviously, the minimizer of $\psi(s)$ will not induce a decrease of J, i.e., $J(u) \leq J(u+s)$. Though, an actual decrease will be computed, if the Trust-Region radius is chosen small enough.

Now, for a given iterate u_i the *Trust-Region model* is given as $\psi_i : \mathbb{R}^n \to \mathbb{R}$ with

$$\psi_i(s) = \langle g_i, s \rangle + \frac{1}{2}\langle s, B_i s \rangle \tag{2.1.1}$$

where $g_i = \nabla J(u_i)$ and $B_i = B(u_i)$. A Trust-Region correction $s_i \in \mathbb{R}^n$ is then computed by means of

$$\min_{s \in \mathbb{R}^n} \psi_i(s), \qquad \text{s.t. } \|s\|_\infty \leq \Delta_i \text{ and } u_i + s \in \mathcal{B} \tag{2.1.2}$$

where $\Delta_i \in \mathbb{R}^+$ is called Trust-Region radius and $\mathcal{B}$ is the set of admissible solutions for problem (M). As it will turn out, s_i is not necessarily an exact solution of (2.1.2). It can rather be an approximation to the minimizer, as long as s_i satisfies a sufficient decrease condition like condition (2.1.8). This is reasonable, since, in fact, from a numerical point of view, the exact solution itself would due to rounding errors generally be impossible. However, if B_i is positive definite, the exact solution of $\psi_i(s) = \min!$ is a Quasi-Newton step.

2.1.2 Decrease Ratio and Trust-Region Update

Since, on the one hand the correction s_i is computed approximately and, on the other, J is arbitrarily non–linear, one has to control the "quality" of s_i. There we define the *actual* and the *predicted reduction* as

$$\begin{aligned} \text{ared}_i(s_i) &= J(u_i) - J(u_i + s_i) \\ \text{pred}_i(s_i) &= -\psi_i(s_i) \end{aligned}$$

Now, the decrease ratio ρ_i is defined by

$$\rho_i = \frac{\text{ared}_i(s_i)}{\text{pred}_i(s_i)} \tag{2.1.3}$$

Note that, in fact, if $B_i = \nabla^2 J(u_i)$, $\text{pred}_i(s)$ measures $J(u_i) - J(u_i + s)$ employing a second–order Taylor approximation of $J(u_i + s)$. Moreover, if J is a quadratic function and $B_i = \nabla^2 J(u_i)$, the predicted and actual reduction are the same, i.e., $\text{pred}_i(s) = \text{ared}_i(s)$ and, thus, $\rho_i = 1$. In this case,

Algorithm: Trust-Region Solver

Input: $m, n \in \mathbb{N}$, $\mathcal{B}$, $J : \mathbb{R}^n \to \mathbb{R}$, $u_0 \in \mathbb{R}^n$, $\Delta^0 \in \mathbb{R}^+$
Constants: $\gamma_1, \gamma_2, \eta \in (0,1)$, $m \in \mathbb{N} \cup \{\infty\}$

$i = 0$
do {
  generate ψ_i by means of (2.1.1)
  solve problem (2.1.2) approximately such that (2.1.7) holds, and obtain $s_i \in \mathbb{R}^n$

  compute ρ_i according to (2.1.3)

  if ($\rho_i \geq \eta$)
    $u_{i+1} = u_i + s_i$
  else
    $u_{i+1} = u_i$

  compute Δ_{i+1} by means of (2.1.4)

  if ($i > m$)
    return u_{i+1}

  $i = i + 1$
}

Algorithm 1: Trust-Region Algorithm

it is mandatory to solve (2.1.2) sufficiently good.
The Trust-Region radius Δ_i is updated based on the decrease ratio ρ_i, i.e.,

$$\Delta_{i+1} \in \begin{cases} (\Delta_i, \gamma_2 \Delta_i] & \text{if } \rho_i \geq \eta \\ [\gamma_1 \Delta_i, \Delta_i) & \text{if } \rho_i < \eta \end{cases} \tag{2.1.4}$$

where $1 > \eta > 0$, as well as $\gamma_2 > 1 > \gamma_1 > 0$ are assumed to be given a priori and fixed for the whole computation. In our examples, we use $\gamma_1 \Delta_i$ and $\gamma_2 \Delta_i$ for computing the new radius. However, in other works, for instance in [CL96], more complex update strategies are proposed, in order to reflect the complexity of (M).
In a last step, if

$$\rho_i \geq \eta \tag{2.1.5}$$

holds, a correction is added to the current iterate. In this case, a correction is called *successful*. Otherwise, the correction will be rejected and the Trust-Region radius decreased. These four steps are summed up in Algorithm 1.

2.1.3 Constraints and Scaling Functions

To measure the first–order sufficient conditions in a constrained context, we follow [CL96] where it was shown that the first–order–sufficient conditions of

$$u \in \mathcal{B} : J(u) = \min!$$

are equivalent to

$$u \in \mathcal{B} : D(u)\nabla J(u) = 0$$

where $D(u) \in \mathbb{R}^{n\times n}$ is a diagonal scaling matrix defined as

$$D(u)_{ii} = \begin{cases} (\overline{\phi} - u)_i & \text{if } (\nabla J(u))_i < 0 \\ (u - \underline{\phi})_i & \text{if } (\nabla J(u))_i \geq 0 \end{cases} \tag{2.1.6}$$

For the sake of notational simplicity, we define $\hat{g}_i = D(u_i)\nabla J(u_i)$.
In the next section, we show that Algorithm 1 computes a sequence of iterates converging to a first–order critical point. To derive this asymptotic convergence result, we suppose that an infinite sequence of iterates is computed by the Trust-Region algorithm, i.e., $m = \infty$.
Moreover, Algorithm 1 will also be employed as an embedded solver in Chapter 4 and Chapter 5. Here, in each call of the Trust-Region algorithm, a limited number of Trust-Region steps will be computed.

2.1.4 Convergence to First–Order Critical Points

To ensure convergence to first–order critical points, we search for a lower bound of the actual decrease of the objective function depending only on the first–order conditions and the Trust-Region radius. Since the acceptance criterion already bounds the actual decrease by the predicted one, it suffices to assume that for each computed correction s_i the following *sufficient decrease condition*

$$\mathrm{pred}_i(s_i) \geq \beta\|\hat{g}_i\|_2 \min\{\|\hat{g}_i\|_2, \Delta_i\} \tag{2.1.7}$$

for $\beta > 0$ holds. If, in turn, this condition is satisfied, we obtain for each successful correction

$$J(u_i) - J(u_i + s_i) \geq \eta\mathrm{pred}_i(s_i) \geq \eta\beta\|\hat{g}_i\|_2 \min\{\|\hat{g}_i\|_2, \Delta_i\} \tag{2.1.8}$$

Therefore, in order to prove a sufficient decrease of the objective function, it suffices to assume that the quadratic minimization problem is solved sufficiently accurate. In particular, a correction may satisfy the sufficient decrease condition, even if the quadratic minimization problem (2.1.2) is solved approximately. In fact, the following Cauchy condition can be employed to test whether a correction induces a sufficient decrease or not:

$$\psi_i(s_i) \quad \leq \quad \tilde{\beta}\psi_i(s_i^C) \qquad \text{w.r.t. } \|s_i\|_\infty \leq \Delta_i \text{ and } u_i + s_i \in \mathcal{B} \tag{2.1.9}$$

The Cauchy point $s_i^C \in \mathbb{R}^n$ is the solution of

$$\min_{\{s=-t(D_i)^2 g_i | t\geq 0\}} \{\psi_i(s) : \|s\|_\infty \leq \Delta_i,\ u_i + s \in \mathcal{B}\} \tag{2.1.10}$$

Here, $\tilde{\beta} > 0$ is an a priori chosen constant and $D_i = D(u_i)$ (cf., [CL96] and equation (2.1.11)). Since an accepted Cauchy point induces a sufficient decrease of the objective function, as will be proven in the following lemma, each correction satisfying (2.1.9) does also. Hence, to check whether a computed correction satisfies this criterion or not, one only[2] must compute the Cauchy point as the solution of a scalar constrained minimization problem. Exemplary this is done in a case differentia-

[2] In the case of dense matrices, this might still be expensive since a matrix-vector multiplication usually takes $O(n^2)$ operations

tion in the proof of the following lemma.

Lemma 2.1.1. *Let assumptions ($A_{tr}1$), ($A_{tr}2$) and ($A_{tr}3$) hold. Then for all i, with $u_i \in \mathcal{B}$ such that $\|\hat{g}_i\|_2 > 0$ and all s_i satisfying (2.1.9) it holds*

$$\mathrm{pred}_i(s_i) \geq c\|\hat{g}_i\|_2 \min\{\|\hat{g}_i\|_2, \Delta_i\} \tag{2.1.11}$$

where $c = \tilde{\beta}\min\{1, \frac{1}{2C_O^2 C_B}, \frac{1}{2C_O^2 C_B}, \frac{1}{2C_O}\}$ and $C_O > 0$.

Proof. This proof will be carried out as follows: we will estimate the ψ-value reduction, implied by the Cauchy condition (2.1.9), by a case differentiation.

Criterion (2.1.9) now implies,

$$-\mathrm{pred}_i(s_i) = \psi_i(s_i) \leq \tilde{\beta} \min_{\tau\in[0,\tau^+]} \varphi(\tau)$$

where $\varphi(\tau) = \psi_i(-\tau(D_i)^2 g_i)$ and $\tau^+ = \min\{\tau_{\mathcal{B}}, \tau_\Delta\}$. The constant $\tau_{\mathcal{B}}$ is given by

$$\begin{aligned} \tau_{\mathcal{B}} = \max\{\tau > 0 \quad : \quad & \overline{\phi}_k - u_i + \tau(D_i)^2 g_i \geq 0, \\ & \underline{\phi}_k - u_i - \tau(D_i)^2 g_i \geq 0\} \end{aligned} \tag{2.1.12}$$

and $\|(D_i)^2 g_i\|_\infty \leq \|(D_i)\|_\infty \|\hat{g}_i\|_\infty$, we obtain for τ_Δ

$$\tau_\Delta = \frac{\Delta_i}{\|(D_i)^2 g_i\|_\infty} \geq \frac{\Delta_i}{C_O\|\hat{g}_i\|_\infty} \geq \frac{\Delta_i}{C_O\|\hat{g}_i\|_2}$$

where $C_O > 0$ such that $\forall i : C_O \geq \overline{\phi}_i - \underline{\phi}_i$. Now, we estimate $\tau_{\mathcal{B}}$,

$$\begin{aligned} \tau_{\mathcal{B}} &= \min\{\min_{l:(g_i)_l<0} \tfrac{(\overline{\phi}_k - u_i)_l}{-((D_i)^2 g_i)_l}, \min_{l:(g_i)_l>0} \tfrac{(\underline{\phi}_k - u_i)_l}{((D_i)^2 g_i)_l}\} \\ &= \min_{l:(g_i)_l\neq 0} \tfrac{(D_i)_{ll}}{|((D_i)^2 g_i)_l|} \\ &\geq \tfrac{1}{\|\hat{g}_i\|_\infty} \geq \tfrac{1}{\|\hat{g}_i\|_2} \end{aligned} \tag{2.1.13}$$

Next, we employ $\varphi(\tau) = -\kappa_1\tau + \frac{1}{2}\kappa_2\tau^2$ and

$$\kappa_1 = \langle (D_i)^2 g_i, g_i\rangle = \|\hat{g}_i\|_2^2, \qquad \kappa_2 = \langle (D_i)^2 g_i, B_i (D_i)^2 g_i\rangle$$

This yields to the following case differentiation.

1) If $\tau^* < \tau^+$ is the minimizer of $\varphi(\tau)$ then we directly obtain $\kappa_2 > 0$ and after differentiation $\tau^* = \kappa_1/\kappa_2$. This yields

$$\varphi(\tau^*) = -\frac{\kappa_1^2}{2\kappa_2} \leq -\frac{1}{2}\frac{\|\hat{g}_i\|_2^4}{\|(D_i)^2\|_2\|B_i\|_2\|\hat{g}_i\|_2^2} \leq -\frac{1}{2}\frac{\|\hat{g}_i\|_2^2}{C_O^2 C_B}$$

2a) If $\tau^* = \tau_\Delta$ and $\kappa_2 > 0$ then we have that $\kappa_1/\kappa_2 \geq \tau_\Delta$ and thus

$$\varphi(\tau^*) = -\kappa_1\tau_\Delta + \frac{\kappa_2}{2}\tau_\Delta^2 \leq -\frac{\kappa_1}{2}\tau_\Delta \leq -\frac{1}{2}\frac{\|\hat{g}_i\|_2^2}{C_O\|\hat{g}_i\|_2}\Delta_i \leq -\frac{1}{2C_O^2}\|\hat{g}_i\|_2\Delta_i$$

2b) If $\tau^* = \tau_\Delta$ and $\kappa_2 \leq 0$ then we have

$$\varphi(\tau^*) = -\kappa_1 \tau_\Delta \leq -\frac{\kappa_1}{2}\tau_\Delta \leq -\frac{1}{2}\frac{\|\hat{g}_i\|_2^2}{C_O\|\hat{g}_i\|_2}\Delta_i \leq -\frac{1}{2C_O^2}\|\hat{g}_i\|_2\Delta_i$$

3) For $\tau^* = \tau_\mathcal{B}$ with analogous arguments it holds

$$\varphi(\tau^*) \leq -\frac{\kappa_1}{2}\tau_\mathcal{B} \leq -\frac{\|\hat{g}_i\|_2}{2C_O}$$

Gathering these results yields (2.1.11).

□

As Taylor's theorem shows, a quadratic or linear approximation to a function becomes asymptotically exact. Therefore, in the following lemma we prove that the predicted reduction becomes sufficiently accurate if Δ_i becomes sufficiently small.

Lemma 2.1.2. *Let assumptions ($A_{tr}1$), ($A_{tr}2$) and ($A_{tr}3$) hold. Suppose, moreover that $\|\hat{g}_i\|_2 \geq \varepsilon > 0$ and Δ_i is sufficiently small. Then we obtain for the decrease ratio induced by a correction s_i computed in Algorithm 1*

$$\rho_i \geq \eta$$

Proof. Exploiting ($A_{tr}1$), ($A_{tr}2$) and the mean value theorem yields for sufficiently small Δ_i

$$J(u_i + s_i) - J(u_i) = \langle \overline{g}_i, s_i \rangle$$

with $\overline{g}_i = \nabla J(u_i + \tau s_i)$ where $\tau \in (0,1)$. Using the definition of the decrease ratio and ψ_i, as well as ($A_{tr}2$) and ($A_{tr}3$) yields

$$\begin{aligned}
|\mathrm{pred}_i(s_i)||\rho_i - 1| &= |J(u_i + s_i) - J(u_i) - \langle g_i, s_i\rangle - \tfrac{1}{2}\langle s_i, B_i s_i\rangle| \\
&\leq |\tfrac{1}{2}\langle s_i, B_i s_i\rangle| + |\langle \overline{g}_i - g_i, s_i\rangle| \\
&\leq \tfrac{1}{2}C_B\|s_i\|_2^2 + \|\overline{g}_i - g_i\|_2\|s_i\|_2 \\
&\leq \tfrac{n}{2}C_B\|s_i\|_\infty^2 + n\|\overline{g}_i - g_i\|_2\|s_i\|_\infty \\
&\leq \tfrac{n}{2}C_B(\Delta_i)^2 + n\|\overline{g}_i - g_i\|_2\Delta_i
\end{aligned}$$

Note that $\Delta_i \neq 0$ for $i \in \mathbb{N}$ and that we may employ that (2.1.7) holds and obtain $\mathrm{pred}_i(s_i) > 0$. Multiplication with $(\Delta_i)^{-1}$ yields

$$\begin{aligned}
(\Delta_i)^{-1}\beta\varepsilon\min\{\varepsilon, \Delta_i\}|\rho_i - 1| &\leq (\Delta_i)^{-1}|\mathrm{pred}_i(s_i)||\rho_i - 1| \\
&\leq \tfrac{nC_B}{2}\Delta_i + n\|\overline{g}_i - g_i\|_2
\end{aligned}$$

Now, we can conclude that if we reduce Δ_i, the right hand side of this inequality converges to zero and since $\|s_i\|_2$ converges to zero, $(u_i)_i$ converges in $\mathcal{L}$. Therefore, we obtain for $\Delta_i \to 0$ that $|\rho_i - 1| \to 0$ and, in turn,

$$\rho_i \geq \eta$$

for sufficiently small Δ_i. □

Finally, we will prove convergence of a subsequence of the iterates to first–order critical points.

Theorem 2.1.3. *Let assumptions ($A_{tr}1$), ($A_{tr}2$) and ($A_{tr}3$) hold. Then we obtain that the sequence of iterates generated by Algorithm 1 has the property*

$$\liminf_{i\to\infty} \|\hat{g}_i\|_2 = 0$$

Proof. Assume that the proposition does not hold, i.e., there exists an $\varepsilon > 0$ and an index ν_0 such that $\|\hat{g}_i\|_2 > \varepsilon$ for all $i \geq \nu_0$. If this is the case, the sequence of Trust-Region radii converges to zero: if there are only finitely many successful corrections the update criterion (2.1.4) directly implies $\Delta_i \to 0$. If there are infinitely many successful corrections, we have due to (2.1.8)

$$J(u_i) - J(u_{i+1}) \geq \eta\beta\varepsilon \min\{\varepsilon, \Delta_i\}$$

for each successful step. On the other hand, the levelset $\mathcal{L}$ is compact, we obtain

$$J(u_i) - J(u_{i+1}) \to 0$$

which implies $\Delta_i \to 0$.

Now we use Lemma 2.1.2 and obtain that for sufficiently small Δ_i every correction must be successful, which contradicts that $\Delta_i \to 0$ and proves the proposition. □

Theorem 2.1.4. *Let assumptions ($A_{tr}1$), ($A_{tr}2$) and ($A_{tr}3$) hold. Then we obtain that the sequence of iterates generated by Algorithm 1 converges to a first–order critical point, i.e.,*

$$\lim_{i\to\infty} \|\hat{g}_i\|_2 = 0$$

Proof. We follow the proof of Theorem 6.6 [UUH99] which was carried out by contradiction. Due to Theorem 2.1.3 we know that there exists a sequence $(u_{U_j})_j \subset (u_i)_i$ such that

$$\|\hat{g}_{U_j}\|_2 \leq \varepsilon_2$$

Now assume that there exists a subsequence of $(u_i)_i$, such that

$$\|\hat{g}_i\|_2 \geq \varepsilon_1 > 0 \text{ for all } L_j \leq i < U_j \tag{2.1.14}$$

where $L_j, U_j \in \mathbb{N}$ and $\varepsilon_1 > \varepsilon_2$.

Since $\hat{g}_{U_j} \neq \hat{g}_{U_j-1}$, the previous correction must be successful. We employ (2.1.7) and obtain that the actual reduction is bounded away from zero by a term depending on $\|\hat{g}_i\|_2^2$ and Δ_i. In particular, we obtain for all successful corrections in $L_j \leq i < U_j$

$$J(u_{i-1}) - J(u_i) \geq \eta\beta\varepsilon_1 \min\{\varepsilon_1, \Delta_{i-1}\} \tag{2.1.15}$$

Since $\mathcal{L}$ is compact, $J(u_{i-1}) - J(u_i) \to 0$. This implies that Δ_i must converge to zero for all i with $L_j \leq i < U_j$. Therefore we obtain for sufficiently large i

$$\begin{aligned} J(u_{i-1}) - J(u_i) \geq \eta\beta\varepsilon_1\Delta_{i-1} &\geq \eta\beta\varepsilon_1\|u_i - u_{i-1}\|_\infty \\ &\geq \eta\beta\frac{\varepsilon_1}{\sqrt{n}}\|u_i - u_{i-1}\|_2 \end{aligned}$$

Now, from the triangle inequality we obtain

$$J(u_{L_j}) - J(u_{U_j}) \geq \eta\beta\frac{\varepsilon_1}{\sqrt{n}}\|u_{L_j} - u_{U_j}\|_2$$

and, therefore, $\|u_{L_j} - u_{U_j}\|_2 \to 0$. But,

$$\|\hat{g}_{L_j} - \hat{g}_{U_j}\|_2 \geq \|\hat{g}_{L_j}\|_2 - \|\hat{g}_{U_j}\|_2 \geq \varepsilon_1 - \varepsilon_2 > 0 \tag{2.1.16}$$

Note, due to the definition of the scaling matrix $D(u)$ and the assumptions ($A_{tr}1$) and ($A_{tr}2$), we obtain that $\hat{g}$ is uniformly continuous on $\mathcal{L}$. Thus, equation (2.1.16) contradicts the uniform continuity of $\hat{g}$ and assumption (2.1.14) must be wrong. □

2.1.5 Second–Order Convergence

As we have seen in the previous section, the proposed Trust-Region algorithm aims at the computation of first–order critical points, which might, indeed, be no local minimizers. To achieve convergence to second–order critical points, i.e., points $\overline{u}$ which satisfy

$$\begin{aligned} \langle D(\overline{u})s, \nabla^2 J(\overline{u})D(\overline{u})s\rangle &\geq 0 \\ D(\overline{u})\nabla J(\overline{u}) &= 0 \end{aligned}$$

for all $s : \overline{u} + s \in \mathcal{B}$, more restrictive assumptions on the corrections are necessary. In particular, convergence to a second–order critical point can be proven if the quadratic minimization problems (2.1.2) are solved exact and if B_i converges to $\nabla^2 J(u_i)$ [CL96].
Though, obviously, the exact solution of the quadratic model problems is considerably more restrictive than the by now stated assumptions. Moreover, if the Hessian is indefinite, arbitrarily non-convex constrained quadratic minimization problems must be solved which usually makes the application of direct solvers necessary. For an overview of solution strategies for indefinite quadratic minimization problems we refer to [CGT00] and to [YZ01].

2.2 The "Traditional" Linesearch Framework

In the previous section, we have seen that solving non-convex minimization problems by means of a Trust-Region strategy makes the solution of problems of the type (2.1.2) necessary. In case of a positive definite matrix $B(u)$, this minimization problem reduces to a pointwise constrained system of linear equations which might easily be solved employing a projected cg–method in combination with a good preconditioner. But, as pointed out in the previous section, if the Hessian or $B(u)$ is arbitrarily indefinite, the solution of (2.1.2) is computationally expensive. Moreover, if the decrease ratio is not sufficiently good, the correction is discarded and another correction must be computed on basis of altered constraints.
Thus, to save computation time, it often is preferable to somehow damp and apply all computed corrections. Therefore, in each Linesearch step a *search direction* s_i is computed - often as the solution of a quadratic minimization problem - and rescaled employing a *step–length parameter* $\alpha_i \in (0, 1]$. The next iterate is then given as

$$u_{i+1} = u_i + \alpha_i s_i$$

In contrast to Trust-Region methods, the step–length parameter α_i is computed independent from the

solution of the quadratic minimization. But, to ensure convergence the step–length parameter must be chosen such that it satisfies a decrease condition of Armijo–type.

2.2.1 Assumptions on the Objective Function

As in the Trust-Region setting, some modest assumptions on the objective function J from (M) must be stated. But, since no decrease ratio is computed, it is not necessary to formulate assumptions on the matrices B_i. On the other hand, since the acceptance criterion is of Armijo-type it is necessary to suppose that the gradients are Lipschitz continuous on $\mathcal{L}$ (cf., also [NW06, JS04])

($A_{ls}1$) For the given initial iterate $u^0 \in \mathcal{B}$, we assume that the level set

$$\mathcal{L} = \{u \in \mathcal{B} \mid J(u) \leq J(u^0)\}$$

is compact.

($A_{ls}2$) We assume that J is continuously differentiable on $\mathcal{L}$ and that the gradient is Lipschitz continuous with a constant $L_g > 0$, i.e., $\|\nabla J(u) - \nabla J(u+s)\|_2 \leq L_g \|s\|_2$ for all $u \in \mathcal{L}$.

2.2.2 Assumptions on the Search Direction

As we have seen, in the Trust-Region framework assumptions on the correction's quality are stated in form of the sufficient decrease condition and the acceptance criterion. Both criteria ensure a sufficient decrease of the objective function, whenever a correction is accepted. Similarly, in the Linesearch framework, a sufficient decrease of the objective function will be obtained if

- the search direction is a sufficiently good descent direction
- the search direction is scaled such that the Armijo condition is satisfied

As it turns out, the following descent condition is too weak to prove convergence of the resulting Linesearch method

$$\langle s_i, \hat{g}_i \rangle < 0 \tag{2.2.1}$$

where $\hat{g}_i = D(u_i)\nabla J(u_i)$ as defined in Section 2.1.3. Thus, we introduce the more restrictive condition

$$\|s_i\|_\infty^2 \leq \beta_{ls} \|\hat{g}_i\|_\infty^2 \tag{2.2.2a}$$

$$-\langle s_i, g_i \rangle \geq \eta_{ls} \|\hat{g}_i\|_2^2 \tag{2.2.2b}$$

where $\beta_{ls} > \eta_{ls} > 0$ are some positive constants. Similar to the Trust-Region constraint, equation (2.2.2a) leads to an additional condition for the computation of s_i.

A Practicable Decrease Criterion

Similarly to the argumentation for Trust-Region methods, (2.2.2b) is at least for the Cauchy point $s_i^C = -\tau(D_i)^2 g_i$ satsified which is the solution of the following problem. Find a $\tau > 0$ such that

$$-\langle -\tau(D_i)^2 g_i, g_i \rangle = \max! \qquad \text{w.r.t. } u_i - \tau(D_i)^2 g_i \in \mathcal{B} \text{ and } \|\tau(D_i)^2 g_i\|_\infty^2 \leq \beta_{ls} \|\hat{g}_i\|_\infty^2 \tag{2.2.3}$$

In the following lemma we will show that this Cauchy point satisfies the descent condition (2.2.2b). In turn, if a computed search direction does not satisfy (2.2.2b), one might simply substitute this direction by the Cauchy point in order to obtain a valid search direction.

Lemma 2.2.1. *Suppose that assumptions ($A_{ls}1$) and ($A_{ls}2$) hold. Then there exists an $\eta_{ls} > 0$ such that for all $u_i \in \mathcal{B}$ with $\|\hat{g}_i\|_2 > 0$ the Cauchy point s_i^C from (2.2.3) satisfies inequality (2.2.2b).*

Proof. In this proof we estimate the maximal possible step–length τ. Obviously, we obtain by construction $\tau = \min\{\tau_{\mathcal{B}}, \tau_{\Delta}\}$. Similarly to Lemma 2.1.1, we define

$$\begin{aligned} \tau_{\mathcal{B}} &: \max\{\tau > 0 : u_i - \tau(D_i)^2 g_i \in \mathcal{B}\} \\ \tau_{\Delta} &: \max\{\tau > 0 : \|\tau(D_i)^2 g_i\|_\infty^2 \leq \beta_{ls}\|\hat{g}_i\|_\infty^2\} \end{aligned}$$

Now, we employ that there exists an $C_O > 0$ such that $(\overline{\phi} - \underline{\phi})_i^2 \leq C_O$ and obtain

$$\tau_{\Delta} = \beta_{ls}\frac{\|\hat{g}_i\|_\infty^2}{\|(D_i)^2 g_i\|_\infty^2} \geq \beta_{ls}\frac{\|\hat{g}_i\|_\infty^2}{\|D_i\|_\infty^2\|D_i g_i\|_\infty^2} \geq \beta_{ls}\frac{\|\hat{g}_i\|_\infty^2}{C_O\|\hat{g}_i\|_\infty^2} \geq \frac{\beta_{ls}}{C_O}$$

If we suppose that ($A_{ls}1$) and ($A_{ls}2$) hold, we obtain furthermore that there exists a constant $c_g > 0$ such that $\|\hat{g}(u)\|_2 \leq c_g$ for all $u \in \mathcal{L}$. Moreover, equation (2.1.13) gives rise to

$$\tau_{\mathcal{B}} \geq \frac{1}{\|\hat{g}_i\|_2} \geq \frac{1}{c_g}$$

Both together now yields

$$-\langle -\tau(D_i)^2 g_i, g_i\rangle = \tau\langle D_i g_i, D_i g_i\rangle \geq \min\{\frac{1}{c_g}, \frac{\beta_{ls}}{C_O}\}\|\hat{g}_i\|_2^2$$

Now choosing $\eta_{ls} = \min\{\frac{1}{c_g}, \frac{\beta_{ls}}{C_O}\}$ gives rise to

$$-\langle s_i^C, g_i\rangle \geq \eta_{ls}\|\hat{g}_i\|_2^2$$

which concludes the proof. □

In fact, employing the negative gradient as search direction goes back to the famous work of L. Armijo [Arm66], who has proven convergence of a steepest descent method in an unconstrained setting. Though, the steepest descent method employs a first–order approximation on the actual decrease in J and therefore, a condition like

$$\min_{s\in\mathbb{R}^n} \psi_i(s), \qquad \text{such that (2.2.2) and } u_i + s_i \in \mathcal{B} \text{ hold} \tag{2.2.4}$$

would generally yield better search directions. Here, ψ_i is the quadratic model from (2.1.1).

2.2.3 The Armijo Condition as Step Length Control

The second step in computing the actual correction is to ensure a decrease of the objective function. Therefore, we employ the following Armijo condition for the identification of an appropriate step–length α_i, i.e.,

$$J(u_i + \alpha_i s_i) \leq J(u_i) + \rho_A \alpha_i \langle s_i, \nabla J(u_i)\rangle \tag{2.2.5}$$

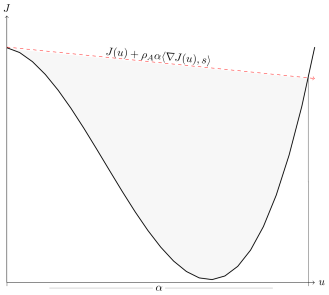

Figure 2.2: This figure illustrates the Armijo condition. As we can see, the ray $J(u_i) + \rho_A\alpha\langle s_i, \nabla J(u_i)\rangle$ starts at $J(u_i)$ and points in direction $\rho_A\langle s_i, \nabla J(u_i)\rangle$. Since $\rho_A \in (0, 1)$, there always exists an environment around u_i where the Armijo condition is satisfied.

where $\rho_A \in (0, 1)$. Note that also unacceptable small step–lengths satisfy (2.2.5) which might lead to extremely slow convergence, or, even worse, to an undesirable stall. Thus, often the Armijo condition is extended to the *Wolfe conditions* by adding the following curvature criterion

$$\langle s_i, \nabla J(u_i + \alpha_i s_i)\rangle \geq \rho_W \langle \nabla J(u_i), s_i\rangle$$

with $\rho_A < \rho_W \in (0, 1)$. A different approach to ensure sufficient progress is to employ the following *backtracking algorithm*, Algorithm 2, to compute a step–length satisfying (2.2.5).

Backtracking Algorithm

Input: $s_i \in \mathbb{R}^n$
Constants: $\alpha_0 \in (0, 1]$, $\tau, \rho_A \in (0, 1)$
Output: Step length α_i

```
i = 0
do {
    if (u_i + α_i s_i satisfies (2.2.5)) {
        return α_i
    } else {
        α_{i+1} = τ α_i
        i = i + 1
    }
}
```

Algorithm 2: Backtracking Algorithm

As it turns out, under the presented assumptions, the backtracking algorithm terminates always after a fixed number of iterations. To this end, we will first show that for sufficiently small α the Armijo condition is satisfied and that along with the assumptions (2.2.2) the number of backtracking iterations just depends on the constants η_{ls}, τ (from Algorithm 2), L_g and β_{ls}.

Lemma 2.2.2. *Assume that ($A_{ls}1$) and ($A_{ls}2$) hold. Then the Armijo condition is satisfied for all $\alpha \leq \hat{\alpha}_i$ where*

$$\hat{\alpha}_i = \frac{2(\rho_A - 1)\langle s_i, g_i\rangle}{L_g \|s_i\|_2^2}$$

Moreover, the backtracking algorithm terminates with a step–length α_i satisfying

$$\min\{\alpha_0, 2\tau\hat{\alpha}_i\} \leq \alpha_i \leq \min\{\alpha_0, 2\hat{\alpha}_i\}$$

where, by definition, $\alpha_0 \leq 1$.

Proof. Since ∇J is Lipschitz continuous, we might apply Taylor's theorem and obtain

$$J(u_i + \alpha s_i) \leq J(u_i) + \alpha\langle s_i, g_i\rangle + \frac{1}{2}L_g\alpha^2\|s_i\|_2^2$$

Now we exploit $\rho_A \in (0,1)$ and obtain for all $\alpha \leq \hat{\alpha}_i$ (by substituting α in the quadratic part) the inequality

$$J(u_i + \alpha s_i) \leq J(u_i) + \alpha\langle s_i, g_i\rangle + \alpha(\rho_A - 1)\langle s_i, g_i\rangle = J(u_i) + \alpha\rho_A\langle s_i, g_i\rangle$$

Thus, the Armijo condition is satisfied for such α. Since $\tau\alpha < \alpha$, we obtain due to the formulation of Algorithm 2 that it terminates for α_i with

$$\min\{\alpha_0, 2\tau\hat{\alpha}_i\} \leq \alpha_i \leq \min\{\alpha_0, 2\hat{\alpha}_i\}$$

□

Next we prove that each step–length parameter α_i, computed in Algorithm 2, is bounded from below by a constant.

Lemma 2.2.3. *Assume that ($A_{ls}1$), ($A_{ls}2$) and (2.2.2) hold. Then we obtain for each α_i computed in Algorithm 2,*

$$\alpha_i \geq \min\left\{\alpha_0, \frac{2\eta_{ls}\tau(1-\rho_A)}{nL_g\beta_{ls}}\right\}$$

Proof. Combining Lemma 2.2.2 with (2.2.2) yields

$$\begin{aligned}
\alpha_i &\geq \min\{\alpha_0, 2\tau\hat{\alpha}_i\} \geq \min\left\{\alpha_0, \frac{2(\rho_A-1)\langle s_i, g_i\rangle}{nL_g\|\hat{s}_i\|_\infty^2}\right\} \\
&\geq \min\left\{\alpha_0, \frac{2(\rho_A-1)\langle s_i, g_i\rangle}{nL_g\beta_{ls}\|\hat{g}_i\|_\infty^2}\right\} \geq \min\left\{\alpha_0, \frac{2(1-\rho_A)\|\hat{g}_i\|_2^2}{nL_g\beta_{ls}\|\hat{g}_i\|_2^2}\right\} \\
&\geq \min\left\{\alpha_0, \frac{2\eta_{ls}(1-\rho_A)}{nL_g\beta_{ls}}\right\}
\end{aligned}$$

□

2.2.4 Convergence to First–Order Critical Points

In this section, we combine the backtracking algorithm and the assumptions on the search direction s_i to a Linesearch algorithm for a solution of problem (M), Algorithm 3.

Theorem 2.2.4. *Suppose ($A_{ls}1$) and ($A_{ls}2$) hold. Then the Linesearch algorithm, Algorithm 3, computes a sequence of iterates converging to a first–order critical point for problem (M).*

Linesearch Algorithm

Input: $u_0 \in \mathbb{R}^n$, $\mathcal{B} \subset \mathbb{R}^n$, $J : \mathbb{R}^n \to \mathbb{R}$
Constants: $\eta_{ls} > 0, \beta_{ls} > 0$

i = 0
do {
compute a search direction s_i satisfying (2.2.2)
call Algorithm 2 with s_i and receive a step–length α_i
set $u_{i+1} = u_i + \alpha_i s_i$
$i = i + 1$
}

Algorithm 3: Linesearch Algorithm

Proof. Due to our assumptions, Lemma 2.2.3 holds and, thus, each α computed in Algorithm 2 is bounded from below, i.e.,

$$\alpha \geq \min\{\alpha_0, 2\tau\alpha_{\min}\} = \min\left\{\alpha_0, \frac{2\eta_{ls}\tau(1-\rho_A)}{nL_g\beta_{ls}}\right\}$$

Moreover, we obtain from the Armijo condition and from (2.2.2a) the following sufficient decrease condition

$$J(u_i) - J(u_{i+1}) \geq -\alpha_i\rho_A\langle s_i, g_i\rangle \geq \alpha_i\rho_A\|\hat{g}_i\|_2^2$$

Since ($A_{ls}1$) implies that $J(u_i)$ converges in $\mathcal{L}$ and since $\alpha_i \geq \min\{\alpha_0, 2\tau\alpha_{\min}\}$ (in all iterations), we obtain

$$\|\hat{g}_i\|_2^2 \to 0$$

which proves the proposition. □

Note that similarly to Trust-Region methods, the combination of step–length control (in this case: the Armijo condition as acceptance criterion), assumptions on the objective function and quality of the search directions yield a sufficient decrease condition. Moreover, the compactness of $\mathcal{L}$ and the boundedness of α, i.e., the "sufficient progress" of the resulting Linesearch algorithm, provide the convergence to first–order critical points.

2.2.5 Second–Order Convergence

Also for Linesearch methods second–order convergence results for problems of the kind (M) have been derived. In this section we briefly sketch the approach of T. Coleman and Y. Li [CL94]. To make a Linesearch algorithm second–order convergent, one must (just like for Trust-Region methods) sharpen the assumption on the search direction. In [CL94], the search direction must be the exact solution of the following quadratic, constrained minimization problem:

$$\min\{\psi_i(s^C) : u_i + s^C \in \mathcal{B}, \|s^C\|_2 \leq \Delta\} \tag{2.2.6}$$

where $\Delta > 0$ is a fixed constant (replacing (2.2.2a)) and for $u_k \to \overline{u}$ also $B_k \to \nabla^2 J(\overline{u})$. Due to the absence of the a posteriori control structures of Trust-Region methods, which can be employed to control the second–order behavior of the objective function, the step–length parameter is computed

by means of the following second–order Armijo condition

$$J(u_i + \alpha_i s) \leq J(u_i) + \rho_A \left(\langle \nabla J(u_i), \alpha_i s \rangle + \frac{\alpha_i^2}{2} \langle s, \nabla^2 J(u_i) s \rangle \right)$$

As it turns out, the combination of the modified Armijo condition with the new search direction now suffices to prove convergence to second–order critical points.

3 A Generic Nonlinear Preconditioning Framework

As we have seen in the previous chapter, in order to ensure convergence, Trust-Region and Linesearch strategies damp or rescale the respective search directions, depending on the nonlinearity of the objective function. In turn, also the convergence rates depend on the nonlinearity of the objective function. This effect generally holds, even if the search directions are the exact solutions of the quadratic minimization problems. Moreover, the (global) rescaling depends on the strongest nonlinearity on the computational domain which often slows down the convergence of large-scale optimization problems. Therefore, it would be desirable to avoid a global rescaling and to adaptively compute search directions within a nonlinear preconditioning step.
Nonlinear preconditioning follows the Krylov-Schwarz paradigm for the iterative solution of linear systems of equations, where additive or multiplicative preconditioners are combined with a Krylov space method. Analogously, in the context of nonlinear fluid dynamics, X.-C. Cai and D.E. Keyes proposed in [CK02] the PRECONDITIONED INEXACT NEWTON method (PIN). The PIN method is a combination of locally applied Newton methods and a global recombination step which together constitute a *nonlinear (left) preconditioner*. But, due to the method's formulation, one cannot determine whether or not the nonlinear preconditioning yields corrections which cause a decrease of the objective function.
Eight years earlier, in 1994 M. Ferris and O. Mangasarian introduced the PARALLEL VARIABLE DISTRIBUTION (PVD) [FM94] which can also be regarded as a preconditioned globalization strategy (for a brief outline of the PVD we refer to Section 3.2.4). In fact, this minimization algorithm asynchronously computes solutions of local minimization problems which are in a second step combined to a global correction. Moreover, the formulation based on the global objective function allows for proving convergence. In order to avoid the exact minimization of the local objective functions without loosing convergence properties, M. Solodov introduced an inexact version of the PVD [Sol97]. Though, a crucial point of the PVD is the computation of a good *set of damping parameters* employed to combine the asynchronously computed corrections. As pointed out in [FR99], these damping parameters can be the solution of another possibly nonconvex constrained minimization problem or simply the best subset correction. The first approach can be realized by employing a filter–based Linesearch strategy (cf., for instance, [WB06]) or a SQP Trust-Region approach (cf., for instance, [WT02]). By simply choosing the best correction, one disposes all but one correction.
Similar problems also arised in the beginning of the development of nonlinear multigrid methods. Due to the formulation of the FAS strategy [Bra81], convergence from arbitrary starting points could not be ensured, too. Only the reformulation based on the objective function and the introduction of control strategies within the MG/OPT strategy [Nas00] allowed for proving convergence.
Influenced by these concepts, we will introduce nonlinear additive and multiplicative preconditioning frameworks employing particular

- domain decompositions
- subset objective functions and obstacles

As pointed out before, these general and – in the additive context – novel frameworks will be exploited to formulate actual Trust-Region and Linesearch implementations in the following chapters.

3.1 The Concept behind Nonlinearly Preconditioned Globalization Strategies

The first concept of a nonlinear preconditioning operator was the PIN strategy presented in [CK02]. Here, the operator is formulated by means of the following optimization problem

$$\overline{u} \in \mathbb{R}^n : G(\nabla J(\overline{u})) = 0 \tag{3.1.1}$$

where G should be easy to implement and speed up the iterative solution process for the original optimization problem

$$\overline{u} \in \mathbb{R}^n : \nabla J(\overline{u}) = 0$$

where $J : \mathbb{R}^n \to \mathbb{R}$ is the objective function from problem (M). In fact, this strategy can be regarded as *left preconditioning*. Here the original minimization problem is obviously changed to a different problem. To show that this minimization problem is equivalent to the original minimization problem, restrictive assumptions on the initial iterate and the problem itself must be stated, as done in [CK02].

3.1.1 Nonlinear Right Preconditioning

Therefore, we will regard the preconditioning ansatz in the present thesis as a *right preconditioning* which acts on the pre-image. For linear systems of equations, right preconditioning reads as follows. For a given $u \in \mathbb{R}^n$ find an $s \in \mathbb{R}^n$ such that

$$AM(u+s) - b = 0 \tag{3.1.2a}$$
$$\overline{u} = M(u+s) \tag{3.1.2b}$$

In the nonlinear case, we are interested in the computation of a critical point $\overline{u}$ such that $\nabla J(\overline{u}) = 0$ which gives rise to the following unconstrained problem. For a given $u \in \mathbb{R}^n$ find an $s \in \mathbb{R}^n$ such that

$$\nabla J(\mathcal{F}(u+s)) = 0 \tag{3.1.3a}$$
$$\overline{u} = \mathcal{F}(u+s) \tag{3.1.3b}$$

where $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a *nonlinear update operator*. Obviously, if both equations hold, $\overline{u}$ is a critical point for J. Therefore, if $J \in C^2(\mathbb{R}^n)$ and $\mathcal{F} \in C^1(\mathbb{R}^n)$ we can apply Newton's method to equation (3.1.3a) which gives rise to the following iterative scheme

$$\nabla^2 J(\mathcal{F}(u^\nu))\mathcal{F}'(u^\nu)s = -\nabla J(\mathcal{F}(u^\nu)) \tag{3.1.4a}$$
$$u^{\nu+1} = \mathcal{F}(u^\nu + s) \tag{3.1.4b}$$

for a given $u^\nu \in \mathbb{R}^n$. Though, a critical point is ensuring a decrease for $u^{\nu+1}$, i.e., $J(u^{\nu+1}) < J(u^\nu)$. Therefore, we will now consider two different update strategies in order to ensure that the resulting method is a globalization strategy.

Linearized Update Strategy

In this approach we replace the right hand side in the update step (3.1.4b) as follows

$$\mathcal{F}(u^\nu + s) \approx \mathcal{F}(u^\nu) + \mathcal{F}'(u^\nu)s$$

In turn, this equation gives rise to the following nonlinear preconditioning scheme

$$\nabla^2 J(\mathcal{F}(u^\nu))\mathcal{F}'(u^\nu)s = -\nabla J(\mathcal{F}(u^\nu)) \tag{3.1.5a}$$

$$u^{\nu+1} = \mathcal{F}(u^\nu) + \mathcal{F}'(u^\nu)s \tag{3.1.5b}$$

This update scheme has two important advantages. On the one hand, it is not necessary to compute $\mathcal{F}(u^\nu + s)$ since the update in (3.1.5b) is based on the already known entities $\mathcal{F}(u^\nu)$ and $\mathcal{F}'(u^\nu)s$. On the other hand (3.1.5b) splits into two steps which can be analyzed separately.
The first step is the computation of $\mathcal{F}(u^\nu)$. As we will see in Chapter 4 and Chapter 5, one important objective of the present thesis is to define nonlinear update operators $\mathcal{F}(u^\nu)$ which satisfy certain sufficient decrease conditions. In turn, as we have seen in Chapter 2, this would give rise to

$$J(\mathcal{F}(u^\nu)) < J(u^\nu)$$

The Newton step in (3.1.4a) will be solved as a Linesearch or Trust-Region step. As we have seen, $s^\nu = \mathcal{F}'(u^\nu)s$ can then be chosen such that a sufficient decrease condition holds. In turn, we obtain

$$J(\mathcal{F}(u^\nu) + s^\nu) < J(\mathcal{F}(u^\nu)) < J(u^\nu)$$

We will see that this suffices to show global convergence of the iterative scheme (3.1.5).

Exact Update Strategy

If one employs $u^{\nu+1} = \mathcal{F}(u^\nu + s)$, convergence of the method can also be shown, if s is sufficiently damped. In particular, as we have just mentioned, we will show that the operator $\mathcal{F}$ can be chosen such that $J(\mathcal{F}(u^\nu)) < J(u^\nu)$ holds. Moreover, if $\mathcal{F} \in C^1(\mathbb{R}^n)$ we employ a damping parameter $\alpha > 0$ and obtain for $\alpha \to 0$ that $\mathcal{F}(u^\nu + \alpha s) \to \mathcal{F}(u^\nu)$. In turn, since J is continuous we obtain

$$J(\mathcal{F}(u^\nu + \alpha s)) < J(u^\nu)$$

for α sufficiently small. Though, the computation of α employing a backtracking algorithm is quite expensive, since in each iteration $\mathcal{F}(u^\nu + \alpha s)$ must be computed. Moreover, in order to derive a convergent scheme, αs or $\mathcal{F}(u + \alpha s)$ must induce a sufficient decrease. Though, as we will see, only for overlapping domain decompositions as in Section 3.1.6, a sufficient decrease for $\mathcal{F}$ can be shown.
Therefore, in the present thesis we will analyze the linearized update strategy. Moreover, in order to show convergence, we will introduce novel additive and multiplicative update operators which allow for proving a sufficient decrease of $\mathcal{F}(u^\nu)$.

The Derivative $\mathcal{F}'$

In some contexts, for instance in the context of the ADDITIVE PRECONDITIONED INEXACT NEWTON (ASPIN) method, $\mathcal{F}'$ can be derived analytically [CK02]. One important assumption in the analysis of the ASPIN operator is that the local problems are solved exactly. Though, as we will

see, in our context this will generally not hold. In this case, it becomes complicated and, perhaps, expensive to compute $\mathcal{F}'$, if it exists.
Whereas, as we have seen in the previous chapter, Trust-Region and Linesearch strategies are able to compute local minimizers, even if the exact Hessians are replaced by approximations. Therefore, in order to avoid the computation of $\mathcal{F}'$, one might employ approximations such as

$$\mathcal{F}'_A(u^\nu) \approx \sum_k I_k(\nabla^2 H_k^\nu(P_k u^\nu))^{-1} R_k \nabla^2 J(u^\nu)$$

for additive preconditioning strategies (cf., [CK02]). In a multiplicative setting one might choose

$$\mathcal{F}'_M(u^\nu) \approx \prod_k I_k(\nabla^2 H_k^\nu(P_k u^\nu))^{-1} R_k \nabla^2 J(u^\nu)$$

3.1.2 Nonlinear Additive and Multiplicative Update Operators

Therefore, we are interested in the construction of nonlinear additive and multiplicative update operators $\mathcal{F}_A, \mathcal{F}_M : \mathbb{R}^n \to \mathbb{R}^n$ which reduce the value of the objective function for a given iterate u^ν as follows

$$J(\mathcal{F}_A(u^\nu) + s^\nu) \leq J(u^\nu) \tag{3.1.6a}$$
$$J(\mathcal{F}_M(u^\nu) + s^\nu) \leq J(u^\nu) \tag{3.1.6b}$$

where J is the objective function in problem (M). The vector $s^\nu = \mathcal{F}'(u^\nu)s$ results from a possible global Trust-Region or Linesearch correction as a solution of, for instance the following problem. Find an $s \in \mathbb{R}^n$ such that

$$\frac{1}{2}\langle s, \nabla^2 J(\mathcal{F}(u^\nu))\mathcal{F}'(u^\nu)s\rangle + \langle s, \nabla J(\mathcal{F}(u^\nu))\rangle = \min! \qquad \text{w.r.t. } u^\nu + \mathcal{F}'(u^\nu)s \in \mathcal{B}$$

As pointed out in the introduction of this chapter, both operators $\mathcal{F}_A$ and $\mathcal{F}_M$ are based on the minimization, or at least on the reduction of certain nonlinear subdomain objective functions such that

$$J(\mathcal{F}_A(u^\nu)) \leq J(u^\nu) \tag{3.1.7a}$$
$$J(\mathcal{F}_M(u^\nu)) \leq J(u^\nu) \tag{3.1.7b}$$

Since this assumption is too weak to ensure global convergence by $\mathcal{F}_A$ and $\mathcal{F}_M$, s^ν must just be computed by means of a globalization strategy from Chapter 2 in order to derive a preconditioned globalization strategy.
But, from a theoretical point of view this does not answer the questions:

- Can one estimate the reduction $J(u) - J(\mathcal{F}_A(u))$ and $J(u) - J(\mathcal{F}_M(u))$?
- Depending on the decomposition of $\mathbb{R}^n$, is it possible to just employ $\mathcal{F}_A$ and $\mathcal{F}_M$ from (3.1.6) to compute a critical point?

An answer to these questions is given in this and the following chapters. In particular, in (4.1.2) and in (5.1.4) we will present actual Trust-Region implementations of the abstract operators in (3.1.6). Actual Linesearch implementations will be introduced in (4.2.2) and in (5.3.7).

3.1.3 Decomposition of the $\mathbb{R}^n$ and Construction of the Transfer Operators

As we have seen in Section 1.3, minimization problems of the kind (M) usually arise from the discretization of different problems stated in some finite dimensional spaces $\mathcal{X}$, for instance the L^p. For example, in Section 1.3, we employ Finite Elements to discretize systems of PDEs giving rise to a problem of type (M). Therefore, the solution of the resulting discrete minimization problem actually yields a set of coefficients of Finite Element functions.
Therefore, for finite dimensional problems we can generally assume that a coordinate isomorphism $X : \mathbb{R}^n \to \mathcal{X}$ exists which maps coefficients to elements in $\mathcal{X}$. Moreover, if $\mathcal{X}$ can be decomposed into N subsets with $\mathcal{X}_k \subset \mathcal{X}$, the original coefficient space $\mathbb{R}^n$ can also be decomposed into subspaces

$$\mathcal{D}_k = \mathbb{R}^{n_k} \subset \mathbb{R}^n \tag{3.1.8}$$

with $n_k \leq n$. In this case, we may also assume that local coordinate isomorphisms $X_k : \mathcal{D}_k \to \mathcal{X}_k$ exist.
For instance, additive preconditioning strategies usually employ horizontal decompositions such as

$$\bigcup_{k=1}^{N} \mathcal{X}_k = \mathcal{X} \qquad \text{and} \qquad \mathcal{D}_k \subset \mathbb{R}^n \qquad \forall k = 1, \dots, N \tag{3.1.9}$$

On the other hand, multiplicative preconditioning strategies can employ vertical decompositions of the kind

$$\mathcal{X} = \mathcal{X}_0 \supsetneq \dots \supsetneq \mathcal{X}_N \qquad \text{and} \qquad \mathbb{R}^n = \mathcal{D}_0 \supsetneq \dots \supsetneq \mathcal{D}_N \tag{3.1.10}$$

3.1.4 The Transfer Operators

Similar to linear preconditioning strategies, the basic principle of the preconditioning approaches in this chapter is to compute search directions for problem (M) by solving related, but less complex subproblems. In linear Schwarz methods, the starting point for the subproblem solution can be chosen arbitrary since one is free to let the current iterate vanish within the linear residual. In this case, the initial iterate on the respective subset is often trivially chosen and, thus, the computed correction vector is the final subset iterate itself.
In the nonlinear case, this does not hold anymore. Here, beginning from a given *initial subset iterate*, a nonlinear subproblem is solved yielding a *subset correction*. This subset correction is the difference between first and last subset iterate. Therefore, the choice of the initial subset iterate crucially influences the nonlinear behavior of the employed local objective function and makes a proper choice of a *projection operator* for primal variables, as will be introduced in (3.1.14), important. However, similarly to the linear case, the resulting subset correction is interpolated using the standard interpolation operator as will be introduced in (3.1.11). Figure 3.1 shows these aspects and highlights the influence of different transfer operators, as will be introduced in the following sections, to the resulting subspace corrections.

Construction of the Interpolation and Restriction Operators

As pointed out before, the decomposition of $\mathbb{R}^n$ is closely related to the decomposition of $\mathcal{X}_k$. Therefore, we define the *interpolation operator* $I_k : \mathcal{D}_k \to \mathbb{R}^n$ as the discretization of the embedding operator mapping from $\mathcal{X}_k$ to $\mathcal{X}$ given by

$$X I_k u = X_k u \qquad \text{for all } u \in \mathcal{D}_k \tag{3.1.11}$$

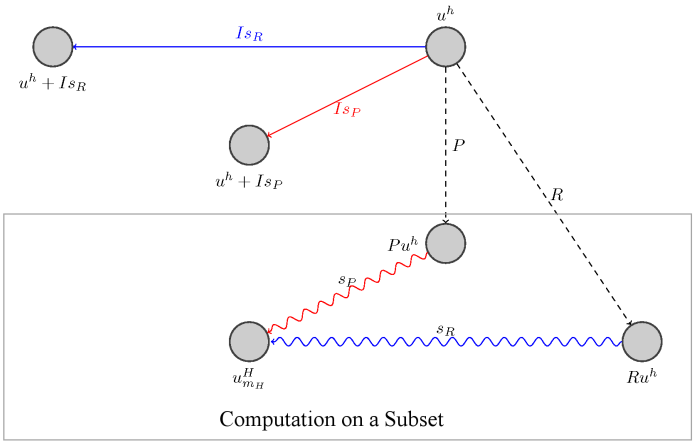

Figure 3.1: This figure shows, how the choice of the initial subset iterate influences the resulting subset correction. In particular, two different subset iterates are chosen, i.e., Ru^h and Pu^h. Both subset computations result in a final subset iterate $u^H_{m_H}$. By definition of the subset iterate, the difference between first and last subset iterate is interpolated and used as correction, i.e., Is_R and Is_P. Applying these subset corrections leads either to $u^h + Is_R$ as next fine level iterate or to $u^H + Is_P$ as next iterate. In general, one of the two corrections reduces the value of the objective function more than the other one.

Similarly, if $\mathcal{X}_{k+1} \subset \mathcal{X}_k$ we may define the interpolation operator $I^k_{k+1} : \mathcal{D}_{k+1} \to \mathcal{D}_k$ as

$$X_k I^k_{k+1} u = X_{k+1} u \tag{3.1.12}$$

Since (3.1.11) and (3.1.12) hold and X_k is an isomorphism, we have that I_k and I^k_{k+1} are uniquely given injections:

$$\begin{aligned} I_k &= X^{-1} X_k \\ I^k_{k+1} &= (X_k)^{-1} X_{k+1} \end{aligned}$$

On the other hand, the *restriction operator* is given by $R_k = (I_k)^T$ and $R^{k+1}_k = (I^k_{k+1})^T$. Moreover, there exist constants $C_R > 0$ and $c_I > 0$ such that for all k the inequalities

$$\|R_k\|_2 = \max_{k=1,\dots,N} \|R_k\|_2 \leq C_R \tag{3.1.13a}$$

$$\lambda_{\min}(R_k I_k) = \min_{k=1,\dots,N} \lambda_{\min}(R_k I_k) \geq c_I \tag{3.1.13b}$$

hold. Here, $\lambda_{\min}(R_k I_k)$ denotes the smallest eigenvalue of the full-ranked matrix $R_k I_k$.

Construction of the Projection Operator

As pointed out before, for nonlinear Schwarz methods initial iterates on the respective subsets should be good approximations to the most current global iterate or the iterate on the preceding subset, respectively. Therefore, we assume that the *projection operator* $P_k : \mathbb{R}^n \to \mathcal{D}_k$ satisfies

$$\|X\left(I_k P_k u - u\right)\|_{\mathcal{X}} \leq \|X\left(I_k v - u\right)\|_{\mathcal{X}} \tag{3.1.14}$$

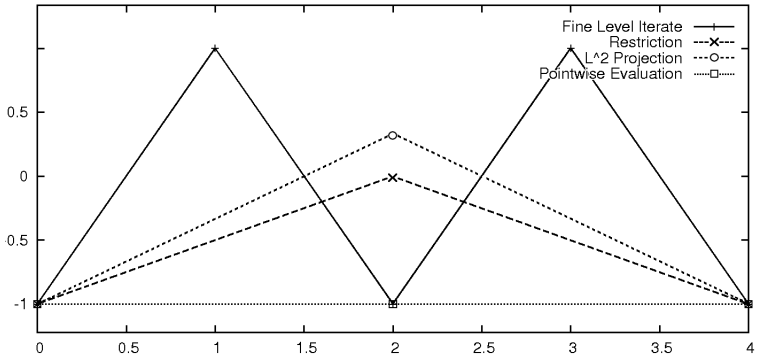

Figure 3.2: Comparison of the approximation strength of subset iterates on a highly frequent global iterate (solid line) in a multiscale setting. In this example, we focus on the computation of the value of differently transferred solutions at Position 2. The values at 0 and 4 are chosen fixed, i.e., these values are Dirichlet values. The shown approximations are: A pointwise evaluation of the original iterate (dotted line with squares), the restricted original iterate (dashed line with cross) and the L^2-projection of the original iterate (dashed line with circle). Note that in this example, the L^2-norm of the distance between the respective projected iterates and the original one are: $\|X_{k-1}(I_k^{k-1}\mathrm{PE}_{k-1}^k u - u)\|_{L^2} = 2.30$ if one evaluates pointwise with PE_{k-1}^k, $\|X_{k-1}(I_k^{k-1}R_{k-1}^k u - u)\|_{L^2} = 1.63$ if the iterate is restricted and only $\|X_{k-1}(I_k^{k-1}P_{k-1}^k u - u)\|_{L^2} = 1.54$ if the iterate is projected.

for all $u \in \mathbb{R}^n$ and all $v \in \mathcal{D}_k$. Similarly we assume that $P_k^{k+1} : \mathcal{D}_k \to \mathcal{D}_{k+1}$ satisfies

$$\|X_k \left(I_{k+1}^k P_k^{k+1} u - u \right) \|_{\mathcal{X}} \leq \|X_k \left(I_{k+1}^k v - u \right) \|_{\mathcal{X}} \tag{3.1.15}$$

for all $u \in \mathcal{D}_k$ and all $v \in \mathcal{D}_{k+1}$.

Remark 3.1.1. *Note that the restriction operator does generally not satisfy (3.1.14) or (3.1.15), respectively, and its approximation strength may be poor (cf., Figure 3.2). This is due to the fact that the restriction operator in standard linear multigrid methods (see for example [Bra07]) is an operator acting on dual spaces. Thus, by design, the restriction operator should only be applied to dual quantities as is the linear defect. See Figure 3.3 for an illustration of different transfer methods for primal variables.*

As a consequence, employing the restriction operator to repeatedly transfer primal variables from one subset to another, is numerically instable, since each transfer adds *artificial values to the transferred vector, i.e.,*

$$X_k R_k I_k R_k u \neq X_k R_k u$$

in contrast to

$$X_k P_k I_k P_k u = X_k P_k u$$

The next theorem shows that if $\mathcal{X}$ is a Hilbert space, the formulation of the projection operator as the solution of a least squares problem guarantees that the operator is well-defined.

Theorem 3.1.2. *Assume that $\mathcal{X}$ and $\mathcal{X}_k$ are Hilbert spaces. Then the projected iterate can be com-*

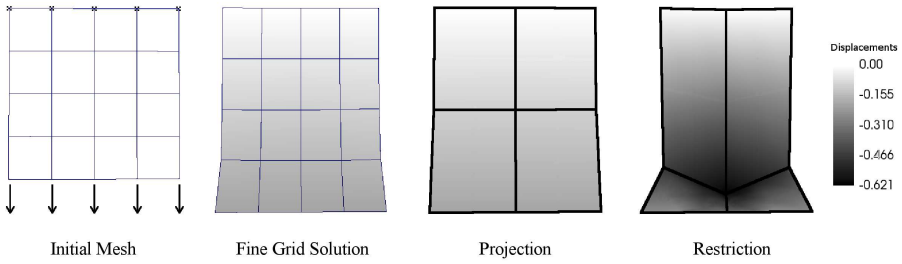

Figure 3.3: Example from continuum mechanics illustrating the difference between the L^2-projection and the standard restriction operator in a multiscale setting. Here, the sought solution is an "energy" optimal displacement field. From left to right: initial uniformly meshed cube (front view), current fine–level iterate (displaced mesh), L^2-projection (see also Figure 3.2 and Chapter 5.5) and restriction of the solution to the next subset. The restriction operator causes a strong displacement of the center node. Using the standard restriction, the value of the displacements at the center node is obtained by adding the fine level displacements of all neighbor nodes with a weighting factor of $1/2$ to the given displacement of the mesh's center node. Thus, the distortion of the restricted solution is dimension and connectivity dependent. On the other hand, the L^2-projection operator passes some mean value of the solution at the center node and its neighbors to the respective node on the subset.

puted as the solution of the following normal equations

$$(XI_k)^T(XI_k)P_k u = (XI_k)^T X u \tag{3.1.16a}$$

$$(X_{k+1}I_{k+1}^k)^T(X_{k+1}I_{k+1}^k)P_k^{k+1} u = (X_{k+1}I_{k+1}^k)^T X_{k+1} u \tag{3.1.16b}$$

Therefore, the projection operators are uniquely given by

$$P_k = \left((XI_k)^T(XI_k)\right)^{-1}(XI_k)^T X \tag{3.1.17a}$$

$$P_k^{k+1} = \left((X_{k+1}I_{k+1}^k)^T(X_{k+1}I_{k+1}^k)\right)^{-1}(X_{k+1}I_{k+1}^k)^T X_{k+1} \tag{3.1.17b}$$

This theorem is a result of Theorem 3.7 in [DH08]. Note that due to the matrix inversions in (3.1.17), the projection operator is in general expensive to compute. In particular, the projection operator may be a dense matrix, even if the interpolation operator is sparse. Thus, often the application of the projection is carried out as the solution of a system of linear equations, equation (3.1.16).

3.1.5 Example: a Multilevel Decomposition of Finite Element Spaces

We consider a multilevel decomposition as given in (3.1.10). For the ease of notation, we will drop the index k and denote the coarse level by H and the fine one by h. In the context of Finite Elements, which are employed to discretize a partial differential equation stated on a certain domain $\Omega \subset \mathbb{R}^d$, $d \in \mathbb{N}$, the coordinate isomorphism is given by

$$X^h = (\lambda_1^h, \ldots, \lambda_{n_h}^h)$$

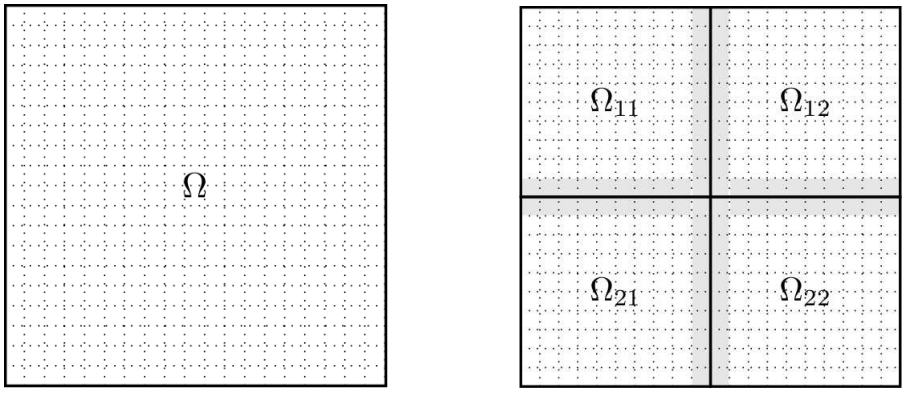

Figure 3.4: A two dimensional example domain (left image) is decomposed into four subdomains (right image). In the setting of a non-overlapping domain decomposition method, usually the nodal basis functions at subdomain edges are just represented by one domain (called master domain). Quadrature, etc. can then be carried out, if also the neighboring elements are known by the master processor. Therefore, a strip of one element width is usually also attached to the respective subdomain, as a row of ghost elements. In contrast, sometimes the parallelization in Finite Element packages, like for instance UG [BBJ+97], is designed such that basis functions at processor edges belong to more than one processor, yielding that some unknowns are represented on multiple processors. These unknowns must then be linearly combined to yield a *consistent* solution.

where $\lambda_i^h : \Omega \to \mathbb{R}^d$ are the basis functions. Here, $(X^h)^T X^h$ is the well-known mass matrix M^h with entries

$$(M^h)_{ij} = (\lambda_i^h, \lambda_j^h)_{L^2(\Omega)}$$

Thus, substituting X^h into (3.1.16a) yields

$$P^H = \left(RM^h I\right)^{-1} RM^h = (M^H)^{-1} RM^h$$

As pointed out in the previous section, one wants to avoid inverting a matrix, even if it is sparse. Moreover, since the mass-matrix is well conditioned and symmetric positive definite, one may employ the cg-method, to compute the projected iterate by simply solving

$$M^H u^H = RM^h u$$

on the subset $\mathcal{D}_H$. Often, for instance for Finite Elements with linear basis functions, it seems also to be convenient to substitute the actual mass matrix by the lumped one. Since the lumped mass matrix is given by

$$(\tilde{M}^H)_{ij} = \begin{cases} \sum_k (M^H)_{ik} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

its inversion is cheap which enables us to approximate the projection by

$$\tilde{P}^H = (\tilde{M}^H)^{-1} RM^h \tag{3.1.18}$$

3.1.6 Example: (Non-) Overlapping Domain Decomposition Methods

The easiest possible setting for constructing the respective operators is a *non-overlapping domain decomposition* of a Finite Element domain. In this case, the basis functions are usually distributed element wise to different subdomains, like shown in Figure 3.4. In our context it suffices to distribute the coefficients. Therefore the set of indices $C = \{1, \dots, n\}$ is distributed such that

$$C = \bigcup_k C_k \text{ with } C_i \cap C_j = \emptyset \text{ iff } i \neq j$$

where $C_i \subset \{1, \dots, n\}$. Then, we may define the operator $\tilde{R}_k$ as

$$\tilde{R}_k = (e_{l_1}, \dots, e_{l_{n_i}})^T \tag{3.1.19}$$

where we assume that $C_k = (l_1, \dots, l_{n_k})$ and e_i is the i-th Euclidean unit vector in $\mathbb{R}^n$. Therefore, we can define $R_k = \tilde{R}_k$, $I_k = \tilde{R}_k^T$ and $P_k = \tilde{R}_k$.

In the case of *overlapping domain decomposition* methods (cf. for instance in [Bas96]), the interpolation operator is employed to linearly combine different vectors from different subsets as follows

$$(s)_i = \sum_k \mu_{i_k} (s_k)_{i_k} \tag{3.1.20}$$

where $s_k \in \mathcal{D}_k$. Here, the index i_k corresponds to the index i on $\mathcal{D}_k$ and $\mu_{i_k} \in [0, 1]$. If $i \notin C_k$ we simply define $\mu_{i_k} = 0$. Often, it is reasonable to assume that the sum of the respective weights equals one, i.e.,

$$\sum_k \mu_{i_k} = 1$$

If this is not the case, the interpolation operator will over-relax or under-relax the computed subset corrections. However, the analysis of the next sections will hold in this case, too. In either case, the interpolation operator for an overlapping domain decomposition is given by

$$(I_k)_{ij} = \begin{cases} \mu_{i_k} & \text{if } i_k = j \\ 0 & \text{otherwise} \end{cases} \tag{3.1.21}$$

Note that by construction, each global unknown is linked to at most one unknown on each subset. This means, that

$$\text{for all } i, k \text{ there exists at most one } l : \sum_j (I_k)_{ij} = (I_k)_{il}$$

For this class of overlapping domain decomposition methods, the projection operator is given by

$$(P_k)_{ij} = \begin{cases} 1 & \text{if } (I_k)_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

These assumptions are, for instance, satisfied by the interpolation operator in the following example. Here, we decompose the $\mathbb{R}^5$ into two subsets with one node being represented on both subsets. For an illustration of this example we refer to Figure 3.5. Hence, the corresponding interpolation and

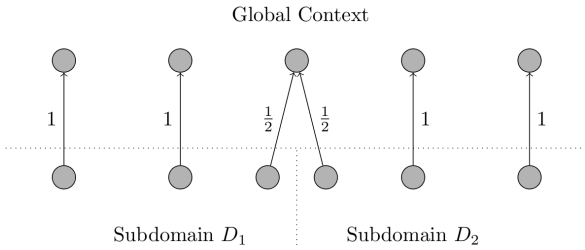

Figure 3.5: A one dimensional domain with 5 unknowns is decomposed into two subdomains. Each of the subsets has 3 unknowns, such that an overlap of one unknown exists. As indicated, the resulting interpolation at this unknown will be the mean value of the corresponding nodes at the subdomains.

projection operators are given by

$$I_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad I_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0.5 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$P_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

3.2 Abstract Formulation of the Nonlinear Additive Preconditioning Operator

Currently two algorithm classes coexist which may be employed to solve (M) in parallel and locally nonlinear: the PVD/PGD framework and the ASPIN framework. Some documented tests for the PVD show, that this approach seems to work well at least for elliptic problems [DS98]. Moreover, the ASPIN method, which does not have a convergence control and may not be employed generally for the solution of the problem (M), has been tested extensively. For instance in [CK02, CKY02, CKM02, HXC05a, HXC05b] it was shown that ASPIN is efficient and reliable for a certain class of PDEs.

In this section, we will present a (novel) generic framework for nonlinear additive preconditioning. Here, we will define nonlinear subset update operators $\mathcal{F}_k$ which reduce the value of particular subset objective functions as will be defined in (3.2.1). The interpolated local corrections are then combined by a nonlinear recombination operator. The result of this recombination process is then the nonlinear update operator $\mathcal{F}_A$. As it will turn out, this concept covers the novel APTS and APLS strategies from Section 4.1 and Section 4.2, but also the PVD approach, as shown in Section 3.2.4.

3.2.1 Derivation of the Additive Subset Objective Function

In this section, we aim at the construction of a nonlinear, additive update operator $\mathcal{F}_A$. We will see that it is crucial to connect the problems, which are solved within $\mathcal{F}_A$, to the global minimization

problem (M). To this end, we follow the approach which S. G. Nash proposed for nonlinear multigrid methods [Nas00], and couple the gradients of the respective objective functions with each other.
Let us start with mentioning that if in the ν-th iteration $u^\nu \in \mathbb{R}^n$ denotes the current global iterate, the initial iterate on $\mathcal{D}_k$ from (3.1.8) is given by $u^\nu_{k,0} = P_k(u^\nu)$. Moreover, we assume that on each subset exist sufficiently smooth (arbitrarily chosen) functions $J^\nu_k : \mathcal{D}_k \to \mathbb{R}$ approximating J on $\mathcal{D}_k$. Note that in many cases also the values of the global iterate u^ν on the neighboring subsets are necessary to compute a proper approximation of J on $\mathcal{D}_k$ which is taken into account by the superscript ν in J^ν_k. For instance, in the context of the examples of Chapter 5.5, we employed a function of the following kind

$$J^\nu_k(u_k) = J(u_k, u^\nu_{\overline{k}})$$

where $u^\nu_{\overline{k}}$ are the components of u^ν which are not represented on $\mathcal{D}_k$.
In the additive case of the present section, the *subset objective function* $H^\nu_k : \mathcal{D}_k \to \mathbb{R}$ for all $\nu \geq 0$ and all $1 \leq k \leq N$ is given by

$$H^\nu_k(u_k) = J^\nu_k(u_k) + \langle \delta g^\nu_k, u_k - u^\nu_{k,0} \rangle \qquad \forall u_k \in \mathcal{D}_k \tag{3.2.1}$$

where, the residual $\delta g^\nu_k \in \mathcal{D}_k$ is given by

$$\delta g^\nu_k = R_k \nabla J(u^\nu) - \nabla J^\nu_k(u^\nu_{k,0})$$

Further assumptions on H^ν_k and the gradients ∇H^ν_k are formulated in Chapter 4. The subset objective function H^ν_k from (3.2.1) has the important property that its gradient is dominated by the restricted global gradient if u_k is sufficiently close to $u^\nu_{k,0}$. This means that $\nabla H^\nu_k(u^\nu_{k,0}) = R_k \nabla J(u^\nu)$ which directly yields that the first Newton step on $\mathcal{D}_k$ is in direction of the restricted gradient. Though, it turns out that this formulation is broad enough to cover, for instance, overlapping domain decomposition methods, or the forget-me-not approach of M.C. Ferris and O.L. Mangasarian as shown in the next section.

3.2.2 Example: The Forget-Me-Not Approach

To speed up the rates of convergence of the PARALLEL VARIABLE DISTRIBUTION approach, in particular if only the best subset correction is chosen, M.C. Ferris and O.L. Mangasarian propose to solve the following problem

$$(u_k, \lambda_k) \in \mathcal{D}_k \times \mathbb{R}^{p_k} : J(u_k,\ u^\nu_{\overline{k}} + S_{\overline{k}}\lambda_{\overline{k}}) \leq J(\tilde{u},\ u^\nu_{\overline{k}} + S_{\overline{k}}\tilde{\lambda}) \qquad \forall (\tilde{u}, \tilde{\lambda}) \in \mathcal{D}_k \times \mathbb{R}^{p_k}$$

where $u^\nu_{\overline{k}}$ are the components of u^ν which are not represented on $\mathcal{D}_k$, $p_k \in \mathbb{N}$ and $S_{\overline{k}} \in \mathbb{R}^{n \times p_k}$ realizes

$$S_{\overline{k}}\lambda = \lambda_1 s_{\overline{k},1} + \ldots + \lambda_p s_{\overline{k},p_k}$$

such that each search direction $s_{\overline{k},i} \in \mathbb{R}^n$ is consistent with the distribution of the variables. Following [FM94], this means, that the interpolation operator

$$I_k = \left(\tilde{I}_k, S_{\overline{k}}\right) \tag{3.2.2}$$

has rank $n_k + p_k$. Here, we used $\tilde{I}_k = (\tilde{R}_k)^T$, $\tilde{P}_k = \tilde{R}_k$ and $\tilde{R}_k$ is as defined in equation (3.1.19). Now, if $u^\nu_{k,0} = (\tilde{P}_k u^\nu, 0)$ we obtain

$$H^\nu_k(\tilde{u}_k) = J(u_k, u^\nu_{\overline{k}} + S_{\overline{k}}\lambda_{\overline{k}})$$

where $\tilde{u}_k = (u_k, \lambda_k)$ since

$$R_k \nabla J(u^\nu) = \nabla_{(u_k,\lambda_k)} J(u_k, u^\nu_{\overline{k}} + S_{\overline{k}}\lambda_{\overline{k}}) \mid_{(u^\nu_{k,0},0)}$$

where $R_k = (I_k)^T$ from (3.2.2)

A Note on Second Order Coupling Terms

In [GST08] it was proposed that one might employ second–order coupling terms, as far as the objective functions are twice continuously differentiable. Also in the additive context, second–order coupling terms could be employed, such as

$$\tilde{H}^\nu_k(u_k) = J^\nu_k(u_k) + \langle \delta g^\nu_k, u_k - u^\nu_{k,0}\rangle + \frac{1}{2}\langle u_k - u^\nu_{k,0}, \delta B^\nu_k \cdot (u_k - u^\nu_{k,0})\rangle$$

where

$$\begin{aligned} \delta g^\nu_k &= R_k \nabla J(u^\nu) - \nabla J^\nu_k(u^\nu_{k,0}) \\ \delta B^\nu_k &= R_k \nabla^2 J(u^\nu) I_k - \nabla^2 J^\nu_k(u^\nu_{k,0}) \end{aligned}$$

In fact, if one employs this subset model, one ties the subset problems closer to the global ones, in particular, if the objective function J^ν_k is not closely related to J. But, even if our analysis, in particular, the arguments in Lemma 4.1.4 and Lemma 4.2.4, still hold, we will, due to necessary smoothness assumptions, focus on H^ν_k as objective function. This enables us to prove global convergence of the APLS and APTS strategies, if J and J^ν_k are just continuously differentiable.

3.2.3 The Nonlinear Additive Update and Preconditioning Operators

Now, the definition of the subset objective function (3.2.1) enables us to introduce a subset update operator $\mathcal{F}_k : \mathcal{D}_k \to \mathbb{R}$ as

$$H^\nu_k(\mathcal{F}_k(P_k u^\nu)) \leq H^\nu_k(P_k u^\nu) \tag{3.2.3}$$

where $u^\nu \in \mathbb{R}^n$. In the context of the present work the actual implementation of $\mathcal{F}_k$ is either a Trust-Region or Linesearch strategy on $\mathcal{D}_k$. Therefore, we can define the additive and nonlinear update operator by

$$\mathcal{F}_A(u^\nu) = \mathcal{A}^\nu\big(I_1(\mathcal{F}_1(u^\nu) - P_1 u^\nu), \ldots, I_N(\mathcal{F}_N(u^\nu) - P_N u^\nu), u^\nu\big) \tag{3.2.4}$$

where $\mathcal{A}^\nu : (\mathbb{R}^n)^N \times \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinear recombination operator. The particular definition of the recombination operator depends on the framework it is used within. For the case of Linesearch methods, the subset correction $s^\nu_k = \mathcal{F}_k(u^\nu) - P_k u^\nu$ is computed such that it satisfies a decrease condition and is, thus, a sufficiently good search direction. In turn, in the APLS strategy in Section 4.2 the recombination operator is given by

$$\mathcal{A}^\nu_{\text{APLS}}\left(I_1 s^\nu_1, \ldots, I_N s^\nu_N, u^\nu\right) = u^\nu + \alpha_A \sum_k I_k s^\nu_k$$

with $\alpha_A \in (0, 1]$ is chosen such that the Armijo condition holds (cf., Section 4.2.1). Note that, as we have seen in Section 2.2, a Linesearch parameter $\alpha_A > 1$ can cause that the rescaled correction is not admissible in $\mathcal{B}$.

On the other hand, in the context of the APTS strategy in Section 4.1, the recombination operator is given by

$$\mathcal{A}^{\nu}_{\text{APTS}}\left(I_1 s_1^{\nu}, \ldots, I_N s_N^{\nu}, u^{\nu}\right) = \begin{cases} u^{\nu} + \sum_k I_k s_k^{\nu} & \text{if } \sum_k I_k s_k^{\nu} \text{ is “sufficiently good”} \\ u^{\nu} & \text{otherwise} \end{cases}$$

In our context, the nonlinear subset update operator $\mathcal{F}_k$ is formulated based on the objective function. Moreover, in contrast to the concepts in [FM94, CK02], $\mathcal{F}_k(u)$ is not necessarily a local minimizer of H_k^{ν}, but in most examples an approximate solution of the problem

$$\mathcal{F}_k(P_k u^{\nu}) \in \mathcal{B}_k : H_k^{\nu}(\mathcal{F}_k(P_k u^{\nu})) = \min!$$

3.2.4 Example: Parallel Variable Distribution

The PVD principle looks similar to the ASPIN method, where in a first step asynchronously local problems are solved, i.e.,

$$s_k^{\nu} \in \mathcal{D}_k : \nabla H_k^{\nu}(P_k u^{\nu} + s_k^{\nu}) = 0$$

Though, the PVD approach employs the local objective function $H_k(\tilde{u}_k) = J(u_k, u_{\overline{k}}^{\nu} + S_{\overline{k}}\lambda_{\overline{k}})$ and the transfer operators as presented in Section 3.2.2. In the PVD context, we are interested in finding a local solution for

$$(u_k, \lambda_k) \in \mathcal{D}_k \times \mathbb{R}^{p_k} : H_k(\tilde{u}_k) = J(u_k, u_{\overline{k}}^{\nu} + S_{\overline{k}}\lambda_{\overline{k}}) = \min!$$

For given $\tilde{u}_k = (u_k, \lambda_{\overline{k}})$ and $s_k^{\nu} = u_k - P_k u^{\nu}$ the recombination operator is the solution of the following nonconvex, constrained minimization problem. Find $(\mu_1, \ldots, \mu_N) \in \mathbb{R}^N$ such that

$$J\left(u^{\nu} + \sum_k \mu_k \left(I_k s_k^{\nu} + S_{\overline{k}}\lambda_{\overline{k}}\right)\right) = \min! \tag{3.2.5a}$$

$$\sum_k \mu_k = 1 \tag{3.2.5b}$$

$$u^{\nu} + \sum_k \mu_k (I_k s_k^{\nu} + S_{\overline{k}}\lambda_{\overline{k}}) \in \mathcal{B} \tag{3.2.5c}$$

Therefore, the recombination operator is given by

$$\mathcal{A}^{\nu}_{\text{PVD}}(I_1 s_1 + S_{\overline{1}}\lambda_{\overline{1}}, \ldots, I_N s_N + S_{\overline{N}}\lambda_{\overline{N}}, u^{\nu}) = u^{\nu} + \sum_k \mu_k (I_k s_k^{\nu} + S_{\overline{k}}\lambda_{\overline{k}})$$

In turn, the nonlinear update operator is given by

$$\mathcal{F}(u^{\nu}) = \mathcal{A}^{\nu}_{\text{PVD}}\left(I_1 s_1 + S_{\overline{1}}\lambda_{\overline{1}}, \ldots, I_N s_N + S_{\overline{N}}\lambda_{\overline{N}}, u^{\nu}\right)$$

As M.C. Ferris and O.L. Mangasarian show, the approach is a globally convergent solution strategy. Though, in order to compute the damping parameters μ in (3.2.5) one must solve another minimization problem which generally cannot be carried out asynchronously.

3.2.5 The Construction of the Subset Obstacles in the Additive Setting

In order to define the subset obstacles, we follow [Man84] and assume that the linear interpolation operators have the following property

$$\begin{array}{rcll} (I_k)_{ij} & \geq & 0 & \forall i,j \\ (I_{k+1}^k)_{ij} & \geq & 0 & \forall i,j \end{array} \tag{3.2.6}$$

Such an assumption is reasonable, for instance for Finite Elements with linear nodal basis functions. Though, in [GMTWM08], S. Gratton et al. have shown that multigrid obstacles can be derived even for non–positive interpolation operators, such that the resulting corrections are admissible in the sense of the obstacles of $\mathcal{B}$. Similar arguments also allow for constructing obstacles for additive decomposition frameworks with non–positive interpolation operators.

However, for the ease of presentation, we restrict ourselves to the case that assumption (3.2.6) holds. Then, the respective subset obstacles are given by

$$(\underline{\phi}_k(u^\nu))_j = (P_k u^\nu)_j + \max_i\{\vartheta_i(\underline{\phi} - u^\nu)_i : (I_k)_{ij} > 0\} \tag{3.2.7a}$$

$$(\overline{\phi}_k(u^\nu))_j = (P_k u^\nu)_j + \min_i\{\vartheta_i(\overline{\phi} - u^\nu)_i : (I_k)_{ij} > 0\} \tag{3.2.7b}$$

with

$$\vartheta_i = \frac{1}{\sum_{k=1}^N \sum_{j=1}^{n_k} (I_k)_{ij}}$$

We also define the set of admissible solutions $\mathcal{B}_k(u^\nu) \subset \mathcal{D}_k$ as

$$\mathcal{B}_k(u^\nu) = \{u_k \in \mathcal{D}_k \mid \underline{\phi}_k(u^\nu) \leq u_k \leq \overline{\phi}_k(u^\nu)\} \tag{3.2.8}$$

where $u^\nu \in \mathbb{R}^n$ is the current global iterate. We will see that this definition of the subset obstacles has two major advantages. By construction, the projection of each admissible iterate is admissible on the subset, i.e., $u \in \mathcal{B} \Rightarrow P_k u \in \mathcal{B}_k(u)$. Moreover, we will see that if the current global iterate and the subset iterate are admissible, then the updated global iterate is also admissible, i.e., $u \in \mathcal{B} \Rightarrow \mathcal{F}_A(u) \in \mathcal{B}$.

Example. In Section 3.1.6, we have seen that the interpolation operator is a permutation matrix, if $\mathbb{R}^n$ is decomposed into N non-overlapping subsets $(\mathcal{D}_k)_k$. In this case, the subset obstacles are trivially given by

$$(\underline{\phi}_k)_i = (\underline{\phi})_j \text{ and } (\overline{\phi}_k)_i = (\overline{\phi})_j$$

where we assumed that the i-th index on a subset $\mathcal{D}_k$ represents the global index j. In the context of the non–overlapping domain decomposition in Section 3.1.6 we have

$$(\underline{\phi}_k)_{i_k} = (\underline{\phi})_j \text{ and } (\overline{\phi}_k)_{i_k} = (\overline{\phi})_j$$

Here, i_k is the index of the unknown which represents the global unknown i on $\mathcal{D}_k$.

In a similar fashion like in [Man84], we will prove that the additively updated global iterate $\mathcal{F}_A(u)$ still is admissible for $\mathcal{B}$. As it turns out, the proof is tailored to the recombination operator of the

PVD approach and the operators in Chapter 4.

Lemma 3.2.1. *Assume that $u \in \mathcal{B}$, that $\mathcal{F}_k(u) \in \mathcal{B}_k(u)$ and that the recombination operator is defined as*

$$\mathcal{A}\left(I_1 s_1, \ldots, I_N s_N, u^\nu\right) = \begin{cases} u^\nu + \sum_k \alpha_k I_k s_k^\nu & \text{if } \sum_k \alpha_k I_k s_k^\nu \text{ is "sufficiently good"} \\ u^\nu & \text{otherwise} \end{cases}$$

where $\alpha_k \in (0,1]$ and $s_k^\nu = \mathcal{F}_k(P_k u^\nu) - P_k u^\nu$. Suppose that the new iterate is given by $\mathcal{F}_A(u) = \mathcal{A}(I_1 s_1, \ldots, I_N s_N, u^\nu)$. Then, we obtain

$$\mathcal{F}_A(u) \in \mathcal{B}$$

Proof. First we will show that $\mathcal{F}_A(u) \geq \underline{\phi}$. If no correction is applied in $\mathcal{F}_A$, the result is trivial. On the other hand, due to the definition of the subspace obstacles, equation (3.2.7), and

$$\mathcal{F}_k(u)_j \geq (\underline{\phi}_k)_j = (P_k(u))_j + \max_i\{\vartheta_i(\underline{\phi} - u)_i : (I_k)_{ij} > 0\}$$

we have

$$(\mathcal{F}_k(u) - P_k(u))_j \geq \max_i\{\vartheta_i(\underline{\phi} - u)_i : (I_k)_{ij} > 0\}$$

Now we use the definition of ϑ_i, $\underline{\phi}_i - u_i < 0$ and $\alpha_k \leq 1$ and obtain the following inequality

$$\begin{aligned} \sum_{k=1}^N \alpha_k \left(\sum_{j=1}^{n_k} (I_k)_{ij} \max_i\{\vartheta_i(\underline{\phi} - u)_i : (I_k)_{ij} > 0\} \right) &\geq \sum_{k=1}^N \alpha_k \left(\sum_{j=1}^{n_k} (I_k)_{ij} \vartheta_i(\underline{\phi} - u)_i \right) \\ &\geq \sum_{k=1}^N \left(\sum_{j=1}^{n_k} (I_k)_{ij} \vartheta_i(\underline{\phi} - u)_i \right) \\ &= (\underline{\phi} - u)_i \end{aligned}$$

Combining the previous inequalities with the definition of the additive update operator yields

$$\begin{aligned} (\mathcal{F}_A(u))_i &= u_i + \sum_{k=1}^N \alpha_k \left(I_k \left(\mathcal{F}_k(u^\nu) - P_k u\right)\right)_i \\ &= u_i + \sum_{k=1}^N \alpha_k \left(\sum_{j=1}^{n_k} (I_k)_{ij} \left(\mathcal{F}_k(u^\nu) - P_k u\right)_j \right) \\ &\geq u_i + \sum_{k=1}^N \alpha_k \left(\sum_{j=1}^{n_k} (I_k)_{ij} \max_i\{\vartheta_i(\underline{\phi} - u)_i : (I_k)_{ij} > 0\} \right) \\ &\geq u_i + \left((\underline{\phi})_i - u_i\right) = (\underline{\phi})_i \end{aligned}$$

Similar arguments yield $(\mathcal{F}_A(u))_i \leq (\overline{\phi})_i$ which proves the lemma. □

3.3 Abstract Formulation of the Nonlinear Multiplicative Preconditioning Operator

Multiplicative, nonlinear preconditioning strategies are subject to research since the introduction of the FULL APPROXIMATION SCHEME (FAS) by A. Brandt [Bra81]. Similar to the ASPIN method, the FAS method provably converges for elliptic problems (see, for instance [Reu88a, Reu88b]).

Almost 20 years later, S. Nash introduced the MG/OPT scheme, a nonlinear multigrid method with globalization properties [Nas00]. Moreover, MG/OPT has proven to be efficient and reliable as presented in various scientific works [LN05a, LN05b, LN06]. Also extensions to Trust-Region and further Linesearch frameworks have been proven to be highly efficient and reliable globalization strategies [GMTWM08, GMS$^+$09, GK08b, GK08c, WG08].

In this section, we will introduce a framework for the multiplicative update operator $\mathcal{F}_M$ which extends the MG/Opt framework to a more general, constrained framework and employs the novel projection operator introduced in Section 3.1.4. But, in contrast to the additive framework, the nonlinear update operator is recursively formulated since, as for multiplicative Schwarz methods in general, the computation of each new correction depends on the previous ones. In turn, the subset objective functions and the subset obstacles depend on the current iterate of the previous subset and no longer on the most current global iterate.

3.3.1 Derivation of the Multiplicative Subset Objective Function

It will turn out that the multiplicative update operator $\mathcal{F}_M$ is based on inclusions of the respective subsets. Similar to the additive framework, on each subset a local smoother $\mathcal{F}_k$ is applied to compute a new iterate. Then, either a recursion is called, if $C_k \supset C_{k+1}$, or the computed correction is interpolated to the previous subset. Here, C_k is a set of indices represented on $\mathcal{D}_k$. As we have seen, in the context of domain decomposition methods, these stand for basis functions which are represented on the k-th subdomain. In contrast, in multiscale methods, these indices stand for nodes which are part of the coarse and fine grid.

Therefore, similarly to the objective function used for the nonlinear additive preconditioning, the objective function depends on a restricted gradient and a subset objective function which may be chosen arbitrarily. Thus, the initial iterate on $\mathcal{D}_k \neq \mathbb{R}^n$ in the ν-th iteration is given by $u^\nu_{k,0} = P^k_{k-1}(u^\nu_{k-1})$, where u^ν_{k-1} is the current iterate on $\mathcal{D}_{k-1}$.

Therefore, for a given subset function $J^\nu_k : \mathcal{D}_k \neq \mathbb{R}^n \to \mathbb{R}$ the (multiplicative) *subset objective function* $H^\nu_k : \mathcal{D}_k \to \mathbb{R}$ is given by

$$H^\nu_k(u_k) = \begin{cases} J^\nu_k(u_k) + \langle \delta g^\nu_k, u_k - u^\nu_{k,0} \rangle & \text{if } \mathcal{D}_k \neq \mathbb{R}^n \\ J(u_k) & \text{otherwise} \end{cases} \tag{3.3.1}$$

for all $\nu \geq 0$ and all $0 \leq k \leq N$. If $\mathcal{D}_k \neq \mathbb{R}^n$, the modified residual $\delta g^\nu_k \in \mathcal{D}_k$ is given by

$$\delta g^\nu_k = R^{k-1}_k \nabla H^\nu_{k-1}(u^\nu_{k-1}) - \nabla J^\nu_k(u^\nu_{k,0})$$

Note that, in the multiplicative setting, also alternating domain decomposition methods may be applied yielding that some sets $\mathcal{D}_k$ are the global solution space $\mathbb{R}^n$, for instance as in the Gauß-Seidel scheme in Section 3.3.3. In turn, we obtain a case differentiation in (3.3.1).

A Note on Second Order Coupling Terms

Also in the multiplicative context, one may employ second–order coupling terms, as employed in the following alternative subset objective function

$$\tilde{H}_k^\nu(u_k) = J_k^\nu(u_k) + \langle \delta g_k^\nu, u_k - u_{k,0}^\nu \rangle + \frac{1}{2}\langle u_k - u_{k,0}^\nu, \delta B_k^\nu \cdot (u_k - u_{k,0}^\nu) \rangle$$

where

$$\begin{aligned} \delta g_k^\nu &= R_{k-1}^k \nabla H_{k-1}^\nu(u_{k-1}^\nu) - \nabla J_k^\nu(u_{k,0}^\nu) \\ \delta B_k^\nu &= R_{k-1}^k \nabla^2 H_{k-1}^\nu(u_{k-1}^\nu) I_k^{k-1} - \nabla^2 J_k^\nu(u_{k,0}^\nu) \end{aligned}$$

if $\mathcal{D}_k \neq \mathbb{R}^n$. This second–order coupling term also yields a closer relationship between the subset objective functions. In this case, the nonlinear multiplicative scheme becomes somehow more similar to the linear scheme, for u_k sufficiently close to $u_{k,0}^\nu = P_{k-1}^k u_{k-1}^\nu$. However, our analysis of Chapter 5, in particular the results Lemma 5.1.4 and Lemma 5.3.5, still hold for this objective function. But, to keep our assumptions in Chapter 5 as simple as possible, we will just employ the first-order model (3.3.1).

3.3.2 The Nonlinear Multiplicative Update and Preconditioning Operator

In contrast to the additive preconditioning operator, the multiplicative version must be formulated recursively. As in the additive context the local update operator $\mathcal{F}_k : \mathcal{D}_k \to \mathcal{D}_k$ has the property

$$H_k^\nu(\mathcal{F}_k(u)) \leq H_k^\nu(u)$$

However, due to the multiplicative context, the update operator is more complex than the additive one and given by

$$\mathcal{F}_M(u^\nu) = \mathcal{A}_0\left(S_0(u^\nu) - u^\nu, u^\nu\right)$$

where $\mathcal{A}_k : \mathcal{D}_k \times \mathcal{D}_k \to \mathcal{D}_k$ is the nonlinear recombination operator. Here we used the nonlinear operator S_k which – by construction – controls the recursions by means of the relationships between two succeeding subsets $\mathcal{D}_k$ and $\mathcal{D}_{k+1}$. This operator is defined as follows

$$S_k(u_k) = \begin{cases} S_{r_k}\left(\mathcal{A}_k\left(I_{k+1}^k\left(S_{k+1}(P_k^{k+1}\mathcal{F}_k(u_k)) - P_k^{k+1}\mathcal{F}_k(u_k)\right), \mathcal{F}_k(u_k)\right)\right) & \text{if } C_k \supsetneq C_{k+1} \\ S_{k+1}(\mathcal{F}_k(u_k)) & \text{if } C_k = C_{k+1} \\ \mathcal{F}_k(u_k) & \text{if } C_k \subsetneq C_{k+1} \end{cases}$$

where r_k is the index when the recursion returns to the subset $\mathcal{D}_k$. Therefore, it is mandatory that if from C_k a recursion is called, there exists an index $r_k > k$ such that $C_{r_k} = C_k$. Though, in order to allow for employing pure pre or post-smoothing strategies, the definition of $\mathcal{F}_k$ also covers $\mathcal{F}_k(u_k) = u_k$.
We will consider the respective cases in more detail.

- The first case realizes the recursive part: after calling the nonlinear update operator $\mathcal{F}_k$, a recursion is called. The resulting *recursively* (or *multiplicatively*) *computed correction* is the interpolated difference between final and initial iterate on $\mathcal{D}_{k+1}$, i.e.,

$$I_{k+1}^k s_{k+1}^\nu = I_{k+1}^k\left(S_{k+1}(P_k^{k+1}\mathcal{F}_k(u_k)) - P_k^{k+1}\mathcal{F}_k(u_k)\right)$$

Algorithm: Multiplicative Update Operator

Input: $u_{k,0} \in \mathcal{D}_k, k \in \mathbb{N}$
Output: $u_k \in \mathcal{D}_k$

repeat {
  $u_k = \mathcal{F}_k(u_{k,0})$
  if $(C_{k+1} \supset C_k)$
    return u_k

  else if $(C_{k+1} = C_k)$ {
    $u_{k+1} = u_k$
    $k = k + 1$

  } **else if** $(C_{k+1} \subset C_k)$ {
    call *Multiplicative Preconditioning Operator* with $u_{k+1} = P_k u_k$
    and receive $u_{k+1,m_{k+1,f}} \in \mathbb{R}^{n_{k+1}}$
    $u_k = \mathcal{A}_k(I^k_{k+1}(u_{k+1,m_{k+1,f}} - P_k u_k), u_k)$
    $u_{r_k} = u_k$
    $k = r_k$
  }
}

Algorithm 4: Multiplicative Preconditioning Operator

Then $\mathcal{A}_k$ combines the computed correction $I^k_{k+1} s^\nu_{k+1}$ with $\mathcal{F}_k(u_k)$. Finally, in order to continue with the computation, S_{r_k} is called, where $\mathcal{D}_{r_k} = \mathcal{D}_k$ but $k < r_k$.

- The second case realizes a further call of the smoothing operator on the subdomain $\mathcal{D}_k$.
- The third case realizes a final smoothing step on a subset without calling a recursion, for instance when reaching the coarsest grid within a multigrid setting.

An algorithmic formulation of this operator is given in Algorithm 4.
As a matter of fact, the actual definition of the multiplicative recombination operator $\mathcal{A}_k$ is context dependent. In the context of Trust-Region methods, for instance the MPTS strategy in Section 5.1, this operator is given as

$$\mathcal{A}_{\text{MPTS},k}(s_M, u_k) = \begin{cases} u_k + s_M & \text{if } s_M \text{ is “sufficiently good”} \\ u_k & \text{otherwise} \end{cases}$$

Though, Linesearch strategies, such as the MPLS strategy in Section 5.3.2, employ a rescaling of the corrections as follows

$$\mathcal{A}_{\text{MPLS},k}(s_M, u_k) = u_k + \alpha_M s_M$$

where $\alpha_M \in (0, 1]$.
As we will see in the following sections, this multiplicative scheme is well suited to model various commonly used recursive schemes like V-cycles, W-cycles, but also multiplicative algorithms of Gauß-Seidel type.

3.3.3 Example: A Multiplicative Algorithm of Gauß-Seidel type

Besides the traditional multilevel scheme, also alternating domain decomposition schemes or (block) Gauß-Seidel schemes fit into the just presented multiplicative framework. In the latter case, we de-

compose the $\mathbb{R}^n$ employing a non-overlapping domain decomposition as presented in Section 3.1.6. Our Gauß-Seidel scheme successively computes corrections on subsets. But, in between, we must interpolate the corrections to the global context, ensure a descent and update the global iterate. The updated iterate, in turn, is then projected to the next subset yielding the initial subset iterate. Alltogether this is the well-known (block) Gauß-Seidel scheme.

In this case, we suppose that we can number the respective degrees of freedom such that

$$C_k = \{l_k, \dots, u_k\} \text{ for } k = \{1, \dots, N_I\}$$

where $N_I \in \mathbb{N}$, $l_1 = 1$, $l_k < u_k$, $l_{k+1} = u_k + 1$ and $u_{N_I} = n$. Now the decomposition is given by

$$C_1 = \text{idx}_n, C_2 = I_1, C_3 = \text{idx}_n, C_4 = I_2, \dots, C_{2N_I} = I_{N_I}, C_{2N_I+1} = \text{idx}_n$$

where $\text{idx}_n = \{1, \dots, n\}$ and we define $N = 2N_I + 1$. For instance, if we have $n = 5$ unknowns and two sets, this would be

$$C_1 = \{1, \dots, 5\}, C_2 = \{1, 2, 3\}, C_3 = \{1, \dots, 5\}, C_4 = \{4, 5\}, C_5 = \{1, \dots, 5\}$$

Here, the indices where the recursion returns to $\mathcal{D}_k$ are given by $r_1 = 3$ and $r_3 = 5$. As we have seen before, within such a decomposition framework, the transfer operators are given as $\tilde{I}_k$, $\tilde{R}_k$ and $P_k = \tilde{R}_k$. Furthermore, we suppose that in each global context C_j with $C_j = \text{idx}_n$ the update operator is given by the identity, i.e.,

$$\mathcal{F}_j = \text{Id}$$

Since a global smoothing is missing, a correction must eventually be computed on each subset $\mathcal{D}_k \neq \mathbb{R}^n$. This means that if $\mathcal{F}_k$ is realized by a Trust-Region method, this means that for a sufficiently small Trust-Region radius, all corrections are successful and applied. Thus, "eventually" means an iteration ν when the Trust-Region radius becomes sufficiently small. Therefore, in general we assume that

$$\mathcal{F}_k \neq \text{Id}$$

Along with the framework of Algorithm 4 this constitutes the sought-after nonlinear block Gauß-Seidel framework.

3.3.4 Example: A Multilevel V-Cycle Algorithm

Here, we decompose $\text{idx}_n = \{1, \dots, n\}$ into a sequence of subsets with

$$\begin{aligned} \text{idx}_n &= C_0 \supsetneq C_1 \supsetneq \dots \supsetneq C_{N'} \\ \text{idx}_n &= C_{2N'} \supsetneq C_{2N'-1} \supsetneq \dots \supsetneq C_{N'+1} = C_{N'} \end{aligned}$$

Here we have $N = 2N'$ and $C_i = C_{2N'+1-i}$. Moreover, the indices where the recursions return to $\mathcal{D}_k$ are here given by

$$r_0 = 2N', r_1 = 2N' - 1, \dots, r_{N'-1} = N' + 2$$

or simply $r_i = r_{2N'+1-i}$. This multilevel decomposition may be the result of a successive refinement of a mesh for a Finite Element discretization. In this case, the indices in idx_n represent the Finite Element basis functions.

We consider a simple example with $n = 9$ which may be the result of a uniform refinement of a

simple, one dimensional mesh. In this example we have

$$C_0 = \{1, \ldots, 9\},\ C_1 = \{1, 3, 5, 7, 9\} \text{ and } C_2 = \{1, 5, 9\}$$
$$C_5 = \{1, \ldots, 9\},\ C_4 = \{1, 3, 5, 7, 9\} \text{ and } C_3 = \{1, 5, 9\}$$

Therefore, on each level we compute two smoothing steps, one before and one after the recursion. As a matter of fact, the coarser levels cannot resolve the fine level unknowns $2, 4, 6, 8$. Therefore, it is mandatory to also compute a smoothing step on C_0 or C_5.

3.3.5 The Construction of the Subset Obstacles in the Multiplicative Setting

As in the additive case, we assume that the interpolation operators satisfy property (3.2.6), i.e., that the matrix components are either positive or zero. Now, if $C_{k-1} \supset C_k$ and for a given, admissible iterate $u_{k-1} \in \mathcal{D}_{k-1}$, the set of admissible subset solutions is given by

$$\mathcal{B}_k(u_{k-1}) = \{u_k \in \mathcal{D}_k \mid \underline{\phi}_k(u_{k-1}) \leq u_k \leq \overline{\phi}_k(u_{k-1})\} \tag{3.3.2}$$

with

$$\begin{aligned}\left(\underline{\phi}_k(u_{k-1})\right)_j &= (P_{k-1}^k u_{k-1})_j + \max_i\{\vartheta_i(\underline{\phi} - u_{k-1})_i : (I_k^{k-1})_{ij} > 0\}\\ \left(\overline{\phi}_k(u_{k-1})\right)_j &= (P_{k-1}^k u_{k-1})_j + \min_i\{\vartheta_i(\overline{\phi} - u_{k-1})_i : (I_k^{k-1})_{ij} > 0\}\end{aligned}$$

The scaling is defined by

$$\vartheta_i = \frac{1}{\sum_{j=1}^{n_k}(I_k^{k-1})_{ij}}$$

where $n_k = \dim \mathcal{D}_k$ (cf., Lemma 3.2 [GM90]).
As in Section 3.2.3, we will prove that the multiplicatively computed corrections are admissible in the context of $\mathcal{B}$.

Lemma 3.3.1. *Assume that for all k and $u_k \in \mathcal{B}_k(u_{k-1})$ that $\mathcal{F}_k(u_k) \in \mathcal{B}_k(u_{k-1})$. Moreover assume that $\mathcal{D}_j = \mathbb{R}^n$, $u_j^\nu \in \mathcal{B}$ and that $\mathcal{F}_M^{(j)}(u_j^\nu) = \mathcal{A}_j(I_{j-1}^j s_{j-1}, u_j^\nu)$. Moreover, suppose that the recombination operator is given by*

$$\mathcal{A}_k(I_{k-1}^k s_{k-1}, u_k) = \begin{cases} u_k + \alpha_k I_{k-1}^k s_{k-1} & \text{if } I_{k-1}^k s_{k-1} \text{ is "sufficiently good"} \\ u_k & \text{otherwise} \end{cases}$$

where $\alpha_k \in (0, 1]$. Then, we obtain

$$\mathcal{F}_M^{(j)}(u_j^\nu) \in \mathcal{B}$$

Proof. We will prove the proposition by showing that if the iterate on the previous level is admissible, it yields an admissible recursively computed correction.
First we assume that $\mathcal{D}_k$ is the lowermost subset in the first recursion, such that no recursively computed correction was applied yet. In particular this means that each u_l for $l < k$ was computed by means of $\mathcal{F}_k(u)$, and, thus, is admissible, i.e., $u_l \in \mathcal{B}_l(u_{l-1})$, which is the induction statement.
Now we consider the case that a recursion was called from $\mathcal{D}_{k-1}$. By assumption of this lemma, we have that $\mathcal{F}_k(u_k) \in \mathcal{B}_k(u_{k-1})$. Thus, we have due to the definition of the subspace obstacles for

$u_k = \mathcal{F}_k(P_k^{k-1}u_{k-1})$ on set $\mathcal{D}_k$ that

$$(u_k)_j \geq (\underline{\phi}_k)_j = (P_{k-1}^k u_{k-1})_j + \max_i\{\vartheta_i(\underline{\phi}_k - u_{k-1})_i : (I_k^{k-1})_{ij} > 0\}$$

where $\underline{\phi}_k = \underline{\phi}_k(u^h)$ if $\mathcal{D}_{k-1} \neq \mathbb{R}^n$ or $\underline{\phi}$ if $\mathcal{D}_{k-1} = \mathbb{R}^n$. Then we obtain

$$(u_k - P_{k-1}^k u_{k-1})_j \geq \max_i\{\vartheta_i(\underline{\phi} - u_{k-1})_i : (I_k^{k-1})_{ij} > 0\}$$

Now, we employ the definition of ϑ_i, that $\alpha_k \in (0,1]$ and that $(\underline{\phi}_k - u_{k-1})_i < 0$ and obtain the following estimation

$$\begin{aligned}(u_{k-1})_i + \alpha_k I_k^{k-1}(u_k - P_{k-1}^k u_{k-1})_i &\geq (u_{k-1})_i + \alpha_k \sum_{j=1}^{n_k}(I_k^{k-1})_{ij} \max_i\{\vartheta_i(\underline{\phi}_k - u_{k-1})_i : (I_k^{k-1})_{ij} > 0\} \\ &\geq (u_{k-1})_i + \alpha_k(\underline{\phi}_k - u_{k-1})_i \geq (u_{k-1})_i + (\underline{\phi}_k - u_{k-1})_i\end{aligned}$$

Thus, we obtain

$$(u_{k-1})_i + \alpha_k I_k^{k-1}(u_k - P_{k-1}^k u_{k-1})_i \geq (\underline{\phi}_k)_i$$

Employing analogous arguments shows that

$$(u_{k-1})_i + \alpha_k I_k^{k-1}(u_k - P_{k-1}^k u_{k-1})_i \leq (\overline{\phi}_k)_i \tag{3.3.3}$$

This means, that after interpolating the correction to level $k-1$, the resulting new iterate still is admissible.
Therefore, we can inductively deduce that recursively computed corrections are admissible: by assumption of this lemma and induction statement we have that $\mathcal{F}_k(u_k) \in \mathcal{B}_k(u_{k-1})$ and that

$$\mathcal{A}_k\left(I_{k+1}^k\left(S_{k+1}(P_k^{k+1}\mathcal{F}_k(u_k)) - P_k^{k+1}\mathcal{F}_k(u_k)\right), \mathcal{F}_k(u_k)\right)$$

is admissible. Together this yields that each iterate on $\mathcal{D}_k$ is admissible in $\mathcal{B}_k(u_{k-1})$ and proves the proposition. □

Therefore, we have just shown that a certain class of recombination operators can handle the multiplicative constraints. On the other hand, such results will not hold, if the recombination operator is based, for instance, on a solution of linear systems of equations. In this case, one must solve this linear system subject to the global constraints.
However, in the next two chapters we will introduce particular Linesearch and Trust-Region implementations of the just presented abstract concepts which give rise to the actual update operators $\mathcal{F}_A$ and $\mathcal{F}_M$ which were employed for computing the numerical results in Chapter 5.5.

4 Nonlinear Additively Preconditioned Globalization Strategies

As we have pointed out before, the convergence of globalization strategies particularly depends on the nonlinearities of the objective function and, in turn, on the rescaling of the corrections and search directions. In our case, the Trust-Region corrections are, due to the employed $\|\cdot\|_\infty$-norm, rescaled by means of box-constraints. But, different norms might generally lead to faster convergence. Thus, in the late 1970s many researchers focused on reformulating the Trust-Region constraint by employing scaling matrices and different norms (cf., for instance [Mor78, DS83]). But the Trust-Region rescaling may make the solution of the constrained quadratic minimization problem expensive (cf., for instance [Vav91]) and, in turn, the Trust-Region algorithm itself impracticable. Similarly, it may be desirable to compute Linesearch step–length parameters which adaptively rescale the computed search-direction.
The purpose of this chapter is to introduce two concrete implementations of the additive preconditioning strategy presented in Section 3.2 which aim at the computation of new search directions by the independent solution of local minimization problems. In particular, since during the asynchronous solution Trust-Region radii and Linesearch parameters can be chosen independently on each subset, we derive a locally adaptive globalization strategy for (M).

4.1 Nonlinear Additively Preconditioned Trust–Region Methods

Obviously, the exact solution of local minimization problems as proposed in the PARALLEL VARIABLE DISTRIBUTION framework is expensive and may, for objective functions with arbitrary nonlinearities, result in poor search directions. Therefore, we change the point of view, and consider the adaptively computed corrections as corrections for the global problem (M). In particular, this enables us to control the local step–length by means of one global Trust-Region radius which, by construction, reflects the current nonlinearity of J from (M). In turn, we are not in the need to solve a global minimization problem to compute a set of damping parameters. Therefore, we will just extend the Trust-Region framework of Section 2.1 to the framework of nonlinear additive domain decomposition methods, as presented in Section 3.2.3 to the following assumptions.

($A_{apts}1$) For a given initial global iterate $u^0 \in \mathcal{B}$, and for all $\nu \geq 0$, all $k \in \{1, \ldots, N\}$ and all initial iterates $u^\nu_{k,0} = P_k u^\nu$ on $\mathcal{D}_k$, we assume that the level sets

$$\mathcal{L}^0_G = \{u \in \mathcal{B} \mid J(u) \leq J(u^0)\}$$

and

$$\mathcal{L}^\nu_k = \{u \in \mathcal{B}_k(u^\nu) \mid H^\nu_k(u) \leq H^\nu_k(u^\nu_{k,0})\}$$

are nonempty and compact. Here, the subset objective functions H^ν_k are given by (3.2.1), P_k is defined as in Section 3.1.4 and $\mathcal{B}_k(u^\nu)$ is given by (3.2.8).

($A_{apts}2$) We assume that J is continuously differentiable on $\mathcal{L}_G^0$, and that for all $\nu \geq 0$ and all $k \in \{1, \ldots, N\}$ that H_k^ν is continuously differentiable on $\mathcal{L}_k^\nu$. Moreover, we assume that there exists a constant $C_g > 0$ for all $u \in \mathcal{L}_G^0$ and $u_k \in \mathcal{L}_k^\nu$ such that $\|\nabla J(u)\|_2 \leq C_g$ and $\|\nabla H_k^\nu(u_k)\|_2 \leq C_g$, respectively.

($A_{apts}3$) We assume that for all $\nu \geq 0$ and all $k = \{1, \ldots, N\}$ there exists a constant $C_B > 0$ such that the norm of each symmetric matrix $B(u)$, and $B_k(u_k)$ in (2.1.1) is bounded, i.e., $\|B(u)\|_2 \leq C_B$ and $\|B_k(u_k)\|_2 \leq C_B$ for all $u \in \mathcal{L}_G^0$ and $u_k \in \mathcal{L}_k^\nu$.

Remark 4.1.1. *In contrast to linear additive Schwarz methods, assumption ($A_{apts}1$) – ($A_{apts}3$) cannot be derived from ($A_{tr}1$) – ($A_{tr}3$) since the subset objective function mainly consists of the nonlinear objective function J_k^ν, which may be chosen arbitrarily.*

4.1.1 The APTS Framework

The paradigm of the Nonlinear **A**dditive **P**reconditioned **T**rust-Region **S**trategy, Algorithm 5, is to combine a priori and a posteriori strategies to

1. compute sufficiently "good" corrections
2. ensure a sufficient decrease

In fact, we control the step–length of the locally computed corrections by means of a global Trust-Region radius Δ^ν. Moreover, to prevent that the computations on some subsets dominate the whole strategy, the computation on a subset will only be carried out, if a certain relationship between the initial local gradient and current global gradient is satisfied, i.e., equation (4.1.6). To control that the computed corrections really induce a sufficient decrease we introduce with equation (4.1.1) a new decrease ratio. In combination with the local and global application of the Trust-Region algorithm, Algorithm 1, we obtain a certain implementation of the abstract framework of Section 3.2, the APTS algorithm, Algorithm 5.

Notation

During the parallel solution process we will employ k instances of the Trust-Region Algorithm 1. Therefore, in the ν-th APTS cycle, on subset $\mathcal{D}_k$, in iteration i of Algorithm 1 we will denote the current iterate by $u_{k,i}^\nu$ and the Trust-Region radius by $\Delta_{k,i}^\nu$. Trust-Region corrections will be denoted by $s_{k,i}$. On the other hand, variables in the Trust-Region algorithm employed for a possible global post–smoothing are denoted by $u_{G,i}^\nu$, $\Delta_{G,i}^\nu$ and $s_{G,i}$. The entities before computing the additive corrections are $\Delta^\nu = \Delta_{G,0}^\nu$ and $u^\nu = u_{G,0}^\nu$.

The Nonlinear Update Operator

In the context of the APTS method, Algorithm 5, the nonlinear subset update operator $\mathcal{F}_k$ from equation (3.2.3) is realized by the application of m Trust-Region iterations. Thus, we define for a given global iterate $u^\nu \in \mathbb{R}^n$ the local update operator as

$$\mathcal{F}_k(P_k u^\nu) = u_{k,m}^\nu$$

where, $u_{k,m}^\nu$ is the final iterate on $\mathcal{D}_k$. Thus, the locally computed corrections are defined as

$$s_k^\nu = u_{k,m}^\nu - P_k u^\nu = u_{k,m}^\nu - u_{k,0}^\nu$$

Algorithm: APTS – Nonlinear Additively Preconditioned Trust-Region Algorithm

Input: $J : \mathbb{R}^n \to \mathbb{R}$, $\mathcal{B}$, $u^0 \in \mathbb{R}^n$, $\Delta^0 \in \mathbb{R}^+$, $n \in \mathbb{N}$
Constants: $\gamma_1, \gamma_2, \eta \in \mathbb{R}^+$, m, $m_G \in \mathbb{N}$

$\nu = 0$
do {

Additive Preconditioning
On each subset where (4.1.6) holds,
call Algorithm 1 with m, $\underbrace{\dim \mathcal{D}_k}_{=n}$, $\underbrace{\mathcal{B}_k(u^\nu)}_{=\mathcal{B} \text{ cf. (3.2.8)}}$, $\underbrace{H_k}_{=J}$, $\underbrace{P_k u^\nu}_{=u_0}$, $\underbrace{\Delta^\nu}_{=\Delta_0}$
and modified constraint (4.1.5) and Trust-Region update (4.1.4).

Update and Global Smoothing
compute ρ^ν by means of (4.1.1)
update Δ^ν by means of (2.1.4)

call Algorithm 1 with $\underbrace{m_G}_{=m}$, n, $\mathcal{B}$, J, $\underbrace{\mathcal{F}_A(u^\nu)}_{=u_0}$ from (4.1.2), $\underbrace{\Delta^\nu}_{=\Delta_0}$

Iterate with $u^{\nu+1} = u^\nu_{G,m_G}$ and $\Delta^{\nu+1} = \Delta^\nu_{G,m_G}$, $\nu = \nu + 1$
}

Algorithm 5: APTS – Nonlinear Additively Preconditioned Trust-Region Algorithm

As we have seen, the nonlinear update operator $\mathcal{F}_A$ directly depends on a definition of "sufficiently good". In the traditional Trust-Region framework, this is measured employing the quotient of the actual reduction in J and the (by the quadratic model ψ (2.1.1)) predicted reduction. Similarly, within the context of the RMTR method [GST08], the coarse level objective function also serves as a *model* which allows for employing the quotient between fine-level reduction and coarse level reduction as a *decrease ratio*.

Following this approach, we will consider each subset objective function as a model for J. But, in order to derive a decrease ratio in the additive context we have to take all subset models into account giving rise to the following additive decrease ratio

$$\rho^\nu = \begin{cases} \frac{J(u^\nu) - J(u^\nu + \sum_{k \in C^\nu} I_k s_k^\nu)}{\sum_{k \in C^\nu} \left(H_k^\nu(u_{k,0}^\nu) - H_k^\nu(u_{k,m}^\nu) \right)} & \text{if } \exists k : u_{k,0}^\nu \neq u_{k,m}^\nu \\ 0 & \text{otherwise} \end{cases} \tag{4.1.1}$$

where $C^\nu = \{k = 1, \ldots, N \mid u_{k,m}^\nu \neq u_{k,0}^\nu\}$. Thus, in the APTS framework the nonlinear recombination operator is given by

$$\mathcal{A}^\nu_{\text{APTS}}\left(I_1 s_1^\nu, \ldots, I_N s_N^\nu, u^\nu\right) = \begin{cases} u^\nu + \sum_{k=1}^N I_k s_k^\nu & \text{if } \rho^\nu \geq \eta \\ u^\nu & \text{otherwise} \end{cases}$$

Hence, we just defined the nonlinear additive update operator as

$$\mathcal{F}_A\left(u^\nu\right) = \mathcal{A}^\nu_{\text{APTS}}\left(I_1 s_1^\nu, \ldots, I_N s_N^\nu, u^\nu\right) \tag{4.1.2}$$

The nonlinear preconditioning concept as presented in Section 3.1 includes the computation of post-smoothing steps. In the APTS framework, this is the computation of $m_G \in \mathbb{N}$, global Trust-Region

smoothing steps.
As we will see, post-smoothing is necessary to ensure convergence in certain decomposition frameworks, for instance when employing a multiscale decomposition. In this case, s^ν is computed by means of a global Trust-Region algorithm starting from $\mathcal{F}_A(u^\nu)$, i.e.,

$$J(\mathcal{F}_A(u^\nu) + s^\nu) \leq J(\mathcal{F}_A(u^\nu))$$

In the context of the linearized right preconditioning scheme (3.1.5) this means that we compute $m_G = 1$ Trust-Region steps by means of the modified Hessian in (3.1.5a).

The Local Trust-Region Update in the Additive Context

Since in this parallel Trust-Region framework each solver asynchronously computes a solution for the respective local minimization problems (3.2.3), it becomes necessary to globally control the local step–lengths. To this end, we employ the global Trust-Region radius as maximal step–length for locally computed corrections. In order to ensure that the subset corrections stay within the current global Trust-Region we have to modify the local Trust-Region update. In a first step, the intermediate radius is given by

$$\tilde{\Delta}^\nu_{k,i} \in \begin{cases} (\Delta^\nu_{k,i}, \gamma_2 \Delta^\nu_{k,i}] & \text{if } \rho^\nu_{k,i}(s_{k,i}) \geq \eta \\ [\gamma_1 \Delta^\nu_{k,i}, \Delta^\nu_{k,i}) & \text{if } \rho^\nu_{k,i}(s_{k,i}) < \eta \end{cases} \tag{4.1.3}$$

Then, the new Trust-Region radius will be computed by employing

$$\Delta^\nu_{k,i+1} = \begin{cases} \min\{\tilde{\Delta}^\nu_{k,i}, \Delta^\nu - \|I_k(u^\nu_{k,i+1} - u^\nu_{k,0})\|_\infty\} & \text{if } k \in \{1, \dots, N\} \\ \tilde{\Delta}^\nu_{G,i} & \text{otherwise} \end{cases} \tag{4.1.4}$$

where Δ^ν is the current, global Trust-Region radius. On the other hand, to ensure that the interpolated subset corrections actually are smaller than the global Trust-Region radius, we employ the following local Trust-Region constraint

$$\|s_{k,i}\|_k = \|I_k s_{k,i}\|_\infty \leq \Delta^\nu_{k,i} \tag{4.1.5}$$

Ensuring "Uniform" Convergence of the Parallel Trust-Region Algorithms

The analysis of Trust-Region algorithms (cf., Section 2.1) shows, that if the Trust-Region radius becomes sufficiently small, the decrease ratio which is the comparison between actual and predicted reduction as defined in equation (2.1.3) becomes sufficiently large. In other words, sufficiently small corrections are actually applied and convergence can be achieved. In the additive framework, we now have to control N separate Trust-Region algorithms, which – depending on the current nonlinearity of the respective objective function – may behave completely different. Thus, it may be possible that on some subsets the Trust-Region algorithms straight-forwardly compute local minimizers for H^ν_k, but on different subsets corrections are not applied since the decrease ratios are not sufficiently large. Since, by construction of the APTS algorithm, the initial Trust-Region radius on each subset is given by the current global radius and the number of Trust-Region iterations on each level is limited (cf., Algorithm 5), we introduce an additional criterion to enforce "uniform" convergence on all subsets:

$$\|\hat{g}^\nu_{k,0}\|_2 \quad \geq \quad \kappa_g \|\hat{g}^\nu\|_2 \tag{4.1.6}$$

where $0 < \kappa_g$ is a constant, chosen problem dependent, $\hat{g}^\nu_{k,i} = D^\nu_{k,i} \nabla H^\nu_k(u^\nu_{k,i})$ and $\hat{g}^\nu = D(u^\nu)\nabla J(u^\nu)$. Here, $D^\nu_{k,i} = D^\nu_k(u^\nu_{k,i})$ is the local scaling matrix on $\mathcal{B}_k(u^\nu)$ as given by (2.1.6). In

general, κ_g must be chosen sufficiently small such that computations on all subsets frequently take place. In the case of overlapping domain decomposition methods, as presented in Section 3.1.6, we choose

$$\kappa_g \leq \frac{1}{\sqrt{n}} \min_{i,k}\{\mu_{i_k}\} \tag{4.1.7}$$

In the case of non–overlapping methods, we choose

$$\kappa_g \leq \frac{1}{\sqrt{n}} \tag{4.1.8}$$

4.1.2 Convergence to First-Order Critical Points

As we have seen in Lemma 3.2.1, local iterates which do not violate $\mathcal{B}_k(u^\nu)$ as defined in (3.2.8) yield admissible additive corrections, i.e.,

$$u^\nu + \sum_k I_k s_k^\nu = u^\nu + \sum_k \left(I_k(u_{k,m}^\nu - P_k u^\nu)\right) \in \mathcal{B}$$

By construction the Trust-Region algorithm, Algorithm 1, computes admissible subset iterates and yields, in turn, admissible additive corrections. Though, we have to show that the constraint (4.1.4) ensures that each locally computed correction

$$I_k s_k^\nu = I_k(u_{k,m}^\nu - P_k u^\nu)$$

does not violate the Trust-Region constraint $\|I_k s_k^\nu\|_\infty \leq \Delta^\nu$.

Lemma 4.1.2. *For all $\nu \geq 0$, all $k \in \{1, \dots, N\}$ and each s_k^ν computed and accepted in algorithm APTS, it holds*

$$\|I_k s_k^\nu\|_\infty \leq \Delta^\nu \tag{4.1.9}$$

Proof. Due to $\Delta_{k,0}^\nu = \Delta^\nu$, the Trust-Region update criterion (4.1.4) and the Trust-Region constraint (4.1.5) we have

$$\begin{aligned} \|I_k(u_{k,l}^\nu - u_{k,0}^\nu)\|_\infty &\leq \|I_k(u_{k,l-1}^\nu - u_{k,0}^\nu)\|_\infty + \|I_k s_{k,l-1}\|_\infty \\ &\leq \|I_k(u_{k,l-1}^\nu - u_{k,0}^\nu)\|_\infty + \Delta^\nu - \|I_k(u_{k,l-1}^\nu - u_{k,0}^\nu)\|_\infty = \Delta^\nu \end{aligned}$$

for all $l = 1, \dots, m$ which proves the proposition. □

In Section 2.1.4 we have seen that the sufficient decrease condition is the key to prove convergence to first–order critical points of Trust-Region methods. Since each Trust-Region correction in Algorithm 1 satisfies the sufficient decrease condition, we are able to prove that also the subset corrections induce a sufficient decrease of the objective function J.

Lemma 4.1.3. *Let assumptions ($A_{apts}1$), ($A_{apts}2$) and ($A_{apts}3$) hold. Then we obtain for all subspace corrections $\sum_{k\in C^\nu} I_k s_k^\nu$ which are accepted in Algorithm 5 the following estimation*

$$J(u^\nu) - J(\mathcal{F}_A(u^\nu)) \geq \beta\eta^2 \sum_{k\in\mathcal{C}^\nu} \kappa_g \|\hat{g}^\nu\|_2 \min\left\{\kappa_g \|\hat{g}^\nu\|_2, \gamma_1^m \Delta^\nu\right\} \tag{4.1.10}$$

Here we used $\mathcal{C}^\nu = \{k : u_{k,m}^\nu \neq u_{k,0}^\nu\}$, the subsets where corrections were successfully computed.

Proof. First, we use the definition of ρ^ν from (4.1.1) and obtain

$$J(u^\nu) - J(\mathcal{F}_A(u^\nu)) \geq \eta \sum_{k \in \mathcal{C}^\nu} (H_k^\nu(u_{k,0}^\nu) - H_k^\nu(u_{k,m}^\nu))$$

Let us denote by $*$ the index of the first successful (and therefore applied) correction on subset k. Now, we employ the sufficient decrease condition (2.1.7) which provides the following estimation

$$\begin{aligned} J(u^\nu) - J(\mathcal{F}_A(u^\nu)) &\geq \eta \sum_{k \in \mathcal{C}^\nu} \Big(H_k^\nu(u_{k,0}^\nu) - H_k^\nu(u_{k,m}^\nu) \Big) \\ &\geq \eta \sum_{k \in \mathcal{C}^\nu} \Big(H_k^\nu(u_{k,0}^\nu) - H_k^\nu(u_{k,0}^\nu + s_{k,*}) \Big) \\ &\geq \beta \eta^2 \sum_{k \in \mathcal{C}^\nu} \|\hat{g}_{k,0}^\nu\|_2 \min\{\|\hat{g}_{k,0}^\nu\|_2, \Delta_{k,*}^\nu\} \end{aligned}$$

Now, we employ that $\Delta_{k,*}^\nu \geq \gamma_1^m \Delta_{k,0}^\nu$, $\Delta_{k,0}^\nu = \Delta^\nu$ and $\|\hat{g}_{k,0}^\nu\|_2 \geq \kappa_g \|\hat{g}^\nu\|_2$ and obtain

$$J(u^\nu) - J(\mathcal{F}_A(u^\nu)) \geq \beta \eta^2 \sum_{k \in \mathcal{C}^\nu} \kappa_g \|\hat{g}^\nu\|_2 \min\{\kappa_g \|\hat{g}^\nu\|_2, \gamma_1^m \Delta^\nu\}$$

which proves the lemma. □

The following lemma shows that, similarly to Trust-Region corrections, also additive corrections are eventually applied, if the Trust-Region radius becomes sufficiently small. This result is mainly due to the fact that the modified residual in H_k^ν contains the restricted global gradient. Along with the mean value theorem we will be able to show that the denominator converges to the nominator in ρ^ν from (4.1.1).

Lemma 4.1.4. *Let assumptions ($A_{apts}1$), ($A_{apts}2$) and ($A_{apts}3$) hold and suppose that $\|\hat{g}_k(u_{k,i}^\nu)\|_2 \geq \varepsilon > 0$ and that (4.1.6) holds for at least one subset. Then, for sufficiently small Δ^ν, corrections are computed additively and are successful, i.e.,*

$$\rho^\nu \geq \eta$$

where ρ^ν is as defined in (4.1.1).

Proof. Due to the assumptions of this lemma, Lemma 2.1.2 is applicable and we obtain that if Δ^ν is sufficiently small, corrections are computed on $\mathcal{D}_k$.
Next, we analyze the acceptance criterion in $\mathcal{F}_A(u)$. We employ the mean value theorem to reformulate the numerator of ρ^ν

$$\begin{aligned} J(u^\nu) - J(u^\nu + \sum_{k \in \mathcal{C}^\nu} I_k s_k^\nu) &= -\langle \nabla J(\xi^\nu), \sum_{k \in \mathcal{C}^\nu} I_k s_k^\nu \rangle \\ &= -\sum_{k \in \mathcal{C}^\nu} \langle \nabla J(\xi^\nu), I_k s_k^\nu \rangle \\ &= -\sum_{k \in \mathcal{C}^\nu} \langle R_k \nabla J(\xi^\nu), s_k^\nu \rangle \end{aligned}$$

for sufficiently small Δ^ν. Here, we defined $\xi^\nu = u^\nu + \tau^\nu \sum_{k \in \mathcal{C}^\nu} s_k^\nu$, the subset correction $s_k^\nu =$

$u^\nu_{k,m} - u^\nu_{k,0}$ and $\tau^\nu \in (0,1)$. This yields

$$\rho^\nu = \frac{J(u^\nu) - J(u^\nu + \sum_{k\in\mathcal{C}^\nu} I_k s^\nu_k)}{\sum\limits_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))} = \frac{-\sum_{k\in\mathcal{C}^\nu}\langle R_k\nabla J(\xi^\nu), s^\nu_k\rangle}{\sum\limits_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))}$$

Next we add $\pm\big(\sum\limits_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))\big)$ to the numerator of ρ^ν which provides

$$\rho^\nu = \frac{-\Big(\sum_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))\Big) - \sum_{k\in\mathcal{C}^\nu}\langle R_k\nabla J(\xi^\nu), s^\nu_k\rangle}{\sum_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))} + 1 \tag{4.1.11}$$

The mean value theorem and the definition of the objective functions H^ν_k provide for sufficiently small Δ^ν and $s^\nu_k \in \mathcal{D}_k$

$$\begin{aligned} 0 &< H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}) \\ &= J^\nu_k(u^\nu_{k,0}) - J^\nu_k(u^\nu_{k,m}) - \langle \delta g^\nu_k, s^\nu_k\rangle \\ &= -\langle \nabla J_k(\xi^\nu_k), s^\nu_k\rangle - \langle R_k\nabla J(u^\nu) - \nabla J^\nu_k(u^\nu_{k,0}), s^\nu_k\rangle \end{aligned} \tag{4.1.12}$$

where $\xi^\nu_k = u^\nu_{k,0} + \tau^\nu_k s^\nu_k$ and $\tau^\nu_k \in (0,1)$. Now, we employ (4.1.12) and reformulate (4.1.11)

$$\rho^\nu = \frac{\kappa_1 + \kappa_2}{\sum\limits_{k\in\mathcal{C}^\nu}(H^\nu_k(u^\nu_{k,0}) - H^\nu_k(u^\nu_{k,m}))} + 1$$

where

$$\begin{aligned} \kappa_1 &= \sum_{k\in\mathcal{C}^\nu}\langle \nabla J^\nu_k(\xi^\nu_k) - \nabla J^\nu_k(u^\nu_{k,0}), s^\nu_k\rangle \\ \kappa_2 &= \sum_{k\in\mathcal{C}^\nu}\langle -R_k\nabla J(\xi^\nu) + R_k\nabla J(u^\nu), s^\nu_k\rangle = \sum_{k\in\mathcal{C}^\nu}\langle -\nabla J(\xi^\nu) + \nabla J(u^\nu), I_k s^\nu_k\rangle \end{aligned}$$

Both terms, κ_1 and κ_2, will now be estimated by Δ^ν and some variable $\varepsilon_C > 0$.
Since ∇J^ν and ∇J^ν_k are continuous on a compact set, we obtain uniform continuity of both functions, i.e., for all $\varepsilon_C > 0$ exists a $\Delta_C > 0$ such that for all $\|\xi^\nu_k - u^\nu_{k,0}\|_\infty \le \Delta^\nu \le \Delta_C$ and $\|\xi^\nu - u^\nu\|_\infty \le \Delta^\nu \le \Delta_C$ the following holds

$$\|\nabla J_k(\xi^\nu_k) - \nabla J^\nu_k(u^\nu_{k,0})\|_2 \le \varepsilon_C \text{ and } \|\nabla J(\xi^\nu) - \nabla J(u^\nu)\|_2 \le \varepsilon_C$$

We employ Cauchy-Schwarz's inequality, Lemma 4.1.2 and (3.1.13b) and obtain

$$\begin{aligned} -|\kappa_1| &\ge -\sum_{k\in\mathcal{C}^\nu}\|\nabla J_k(\xi^\nu_k) - \nabla J^\nu_k(u^\nu_{k,0})\|_2\|s^\nu_k\|_2 \ge -\sum_{k\in\mathcal{C}^\nu}\varepsilon_C\|s^\nu_k\|_2 \\ &\ge -\sum_{k\in\mathcal{C}^\nu}\varepsilon_C c_I^{-1}\|I_k s^\nu_k\|_2 \ge -\sum_{k\in\mathcal{C}^\nu}\sqrt{n}\varepsilon_C c_I^{-1}\|s^\nu_k\|_k \ge -\sum_{k\in\mathcal{C}^\nu}\sqrt{n}\varepsilon_C c_I^{-1}\Delta^\nu \\ -|\kappa_2| &\ge -\sum_{k\in\mathcal{C}^\nu}\varepsilon_C\|s^\nu_k\|_k \ge -\sum_{k\in\mathcal{C}^\nu}\sqrt{n}\varepsilon_C\Delta^\nu \end{aligned}$$

Thus, we employ the previous inequalities and (2.1.7), i.e., the positivity of the denominator, and

obtain

$$\begin{aligned}\rho^\nu &\geq -\frac{|\kappa_1|+|\kappa_2|}{\sum\limits_{k\in\mathcal{C}^\nu}(H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,m}^\nu))}+1\\ &\geq -\frac{(1+c_I^{-1})\sum_{k\in\mathcal{C}^\nu}\varepsilon_C\sqrt{n}\Delta^\nu}{\sum\limits_{k\in\mathcal{C}^\nu}(H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,m}^\nu))}+1\\ &\geq -\frac{(1+c_I^{-1})N\sqrt{n}\varepsilon_C\Delta^\nu}{\sum\limits_{k\in\mathcal{C}^\nu}(H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,m}^\nu))}+1\end{aligned}$$

Note that the sufficient decrease condition, equation (2.1.7), gives rise to

$$H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,m}^\nu)\geq H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,*}^\nu)$$

where $*$ denotes the first successful correction on subset $\mathcal{D}_k$. Moreover, we have due to the definition of the Trust–Region update that $\Delta_{k,*}^\nu\geq\gamma_1^m\Delta^\nu$. Therefore we employ $\|\hat{g}_{k,0}^\nu\|_2\geq\kappa_g\|\hat{g}^\nu\|_2\geq\kappa_g\varepsilon$ and (2.1.7) and obtain for Δ^ν sufficiently small

$$H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,*}^\nu)\quad\geq\quad\eta\beta\kappa_g\varepsilon\min\{\kappa_g\varepsilon,\gamma_1^m\Delta^\nu\}\geq\eta\beta\kappa_g\varepsilon\gamma_1^m\Delta^\nu$$

Now, we can conclude

$$\begin{aligned}\rho^\nu &\geq -\frac{(1+c_I^{-1})N\sqrt{n}\varepsilon_C\Delta^\nu}{\sum\limits_{k\in\mathcal{C}^\nu}(H_k^\nu(u_{k,0}^\nu)-H_k^\nu(u_{k,m}^\nu))}+1\\ &\geq -\frac{(1+c_I^{-1})N\sqrt{n}\varepsilon_C\Delta^\nu}{\sum\limits_{k\in\mathcal{C}^\nu}\eta\beta\kappa_g\varepsilon\gamma_1^m\Delta^\nu}+1\geq-\frac{(1+c_I^{-1})N\sqrt{n}\varepsilon_C}{\eta\beta\kappa_g\varepsilon\gamma_1^m}+1\end{aligned}$$

Therefore, we have for sufficiently small ε_C and Δ^ν that $\rho^\nu\geq\eta$ and, thus, each correction $s^\nu=\sum_{k\in C^\nu}I_ks_k^\nu$ is successful, which proves the proposition. □

The next lemma considers the special case of overlapping and non-overlapping domain decomposition methods, as introduced in Section 3.1.6. In this case, one may prove that for sufficiently small Δ^ν

1. condition (4.1.6) is satisfied for at least one domain
2. on each domain where (4.1.6) holds, a Trust-Region correction will be applied

Lemma 4.1.5. *Let assumptions ($A_{apts}1$), ($A_{apts}2$) and ($A_{apts}3$) hold, and assume that $\|\hat{g}^\nu\|_2\geq\varepsilon>0$ for all $\nu>0$. Suppose that either $\mathbb{R}^n$ is overlappingly decomposed and (4.1.7) holds, or that $\mathbb{R}^n$ is non-overlappingly decomposed and (4.1.8) holds[1]. Then, if Δ^ν is sufficiently small, we obtain*

$$\mathcal{C}^\nu\neq\emptyset$$

where $\mathcal{C}^\nu=\{k:u_{k,m}^\nu\neq u_{k,0}^\nu\}$ is the set of computed subset corrections in Algorithm 5.

[1]The respective definitions of these decompositions are given in Section 3.1.6

Proof. To prove the proposition, we have to show that there exists at least one subset $\mathcal{D}_k$ where (4.1.6) holds and where $u^\nu_{k,m} \neq u^\nu_{k,0}$.
First we consider the case of a non–overlapping domain decomposition. Due to the definition of the subsets, the definition of R_k, $\|v\|_\infty \leq \|v\|_2 \leq \sqrt{n}\|v\|_\infty$ and, by (4.1.8), $\kappa_g \leq \frac{1}{\sqrt{n}}$, we obtain that there exists a $k \in \{1, \dots, N\}$ such that

$$\begin{aligned} \|\hat{g}^\nu_{k,0}\|_2 = \|D_k(u^\nu_{k,0})R_k\nabla J(u^\nu)\|_2 &\geq \|D_k(u^\nu_{k,0})R_k\nabla J(u^\nu)\|_\infty \\ &= \|D(u^\nu)\nabla J(u^\nu)\|_\infty \\ &\geq \tfrac{1}{\sqrt{n}}\|D(u^\nu)\nabla J(u^\nu)\|_2 \\ &\geq \kappa_g\|D(u^\nu)\nabla J(u^\nu)\|_2 \end{aligned} \tag{4.1.13}$$

Thus, on this subset $\mathcal{D}_k$ (4.1.6) is satisfied.
Now we consider the overlapping case. Similar to the non-overlapping case, we obtain by construction of the interpolation operator (3.1.21) and assumption (4.1.7) that there exists at least one subset k such that the following inequality holds

$$\begin{aligned} \|\hat{g}^\nu_{k,0}\|_2 = \|D_k(u^\nu_{k,0})R_k\nabla J(u^\nu)\|_2 &\geq \|D_k(u^\nu_{k,0})R_k\nabla J(u^\nu)\|_\infty \\ &= \min_{i,k}\{\mu_{i_k}\}\|D(u^\nu)\nabla J(u^\nu)\|_\infty \\ &\geq \tfrac{1}{\sqrt{n}}\min_{i,k}\{\mu_{i_k}\}\|D(u^\nu)\nabla J(u^\nu)\|_2 \\ &\geq \kappa_g\|D(u^\nu)\nabla J(u^\nu)\|_2 \end{aligned} \tag{4.1.14}$$

In combination with Δ^ν sufficiently small and Lemma 2.1.2 we obtain that in both cases $\mathcal{C}^\nu$ is non–empty. □

Now, we are able to prove the central result of this section namely that Algorithm 5 generates a sequence of iterates with at least one first-order critical accumulation point for problem (M).

Theorem 4.1.6. *Let assumptions ($A_{apts}1$), ($A_{apts}2$), ($A_{apts}3$) hold and suppose that we have either an overlapping or non-overlapping domain decomposition with constants from (4.1.7) and (4.1.8) or that $m_G > 0$ global post-smoothing Trust-Region steps are computed. Then for each sequence of global iterates $(u^\nu_{G,i})_{i,\nu}$ computed in Algorithm 5 it holds*

$$\liminf_{\nu\to\infty, i\in\{0,\dots,m_G\}} \|\hat{g}(u^\nu_{G,i})\|_2 = 0 \tag{4.1.15}$$

Proof. We prove this proposition by contradiction. Assume that there exists an $\nu_0 > 0$ and $\varepsilon > 0$ such that $\|\hat{g}(u^\nu_{G,i})\|_2 \geq \varepsilon$ for all $\nu \geq \nu_0$ and all $i \in \{0, \dots, m_G\}$. We will show, that this assumption implies that $\Delta^\nu_{G,i} \to 0$ for $\nu \to \infty$ and $i \in \{0, \dots, m_G\}$ and, in turn, $\rho^\nu_{G,i}, \rho^\nu \to 1$ which contradicts $\Delta^\nu_{G,i} \to 0$.
First, we will prove that $\Delta^\nu \to 0$ for $\nu \to \infty$. If there is only a finite number of successful corrections, we have due to the definition of $\Delta^\nu_{G,i}$ that $\Delta^\nu_{G,i} \to 0$ for $\nu \to \infty$.
On the other hand, if the sequence of successful corrections owns infinitely many terms, (2.1.7) and Lemma 4.1.3 imply for such corrections

$$J(u^\nu_{G,i+1}) < J(u^\nu_{G,i}) \text{ and } J(\mathcal{F}_A(u^\nu)) < J(u^\nu)$$

Therefore, we have due to ($A_{apts}1$), i.e., the compactness of $\mathcal{L}^0_G$, that

$$J(u^\nu_{G,i}) - J(u^\nu_{G,i+1}) \to 0 \text{ and } J(u^\nu) - J(\mathcal{F}_A(u^\nu)) \to 0$$

for $\nu \to \infty$. The fact that for all unsuccessful corrections $\Delta^\nu_{G,i+1} < \Delta^\nu_{G,i}$, the respective sufficient decrease conditions (2.1.7), (4.1.10) and $\|\hat{g}^\nu_{G,i}\|_2 \geq \varepsilon$ now provide that

$$\Delta^\nu_{G,i} \to 0$$

In fact, now for sufficiently small $\Delta^\nu_{G,i}$ Lemma 2.1.2 and Lemma 4.1.4 eventually yield that $\rho^\nu_{G,i} \geq \eta$ for all $i \in \{0, \ldots, m_G\}$. But:

- If $m_G > 0$, this would yield that $\Delta^{\nu+1} > \Delta^\nu$ and $\Delta^\nu_{G,i+1} > \Delta^\nu_{G,i}$ and, therefore, that the sequence $(\Delta^\nu_{G,i})_{i,\nu}$ is bounded from below.
- If $m_G = 0$ and the decomposition is overlapping or non-overlapping, then Lemma 4.1.5 provides for sufficiently small $\Delta^\nu_{G,i}$ that $C^\nu \neq \emptyset$. Therefore, this would yield that $\Delta^{\nu+1} > \Delta^\nu$ and that the sequence $(\Delta^\nu_{G,i})_{i,\nu}$ is bounded from below.

Together, this proves the proposition. □

The next theorem is closely related to Theorem 5.1.6 and shows that all limit points are first–order critical points.

Theorem 4.1.7. *Let assumptions ($A_{apts}1$), ($A_{apts}2$) and ($A_{apts}3$) hold. Then Algorithm 5 generates a sequence of iterates converging to a first–order critical point, i.e.,*

$$\lim_{\nu \to \infty} \|\hat{g}^\nu\|_2 = 0 \tag{4.1.16}$$

Proof. The proof of this theorem is the same like for Theorem 2.1.4 with the only difference, that depending on the correction (additively computed, or by means of the global Trust-Region algorithm), the sufficient decrease condition looks differently. Therefore, it suffices to substitute (2.1.15) by the following, weaker condition

$$J(u^\nu_{G,i}) - J(u^\nu_{G,i+1}) \geq \eta^2 \beta \sum_{k \in \mathcal{C}^\nu} \varepsilon_2 \min\{\varepsilon_2, \gamma_1^m \Delta^\nu_{G,i}\}$$

for all $i \in \{0, \ldots, m\}$, where $i = 0$ denotes the additively computed correction. □

4.2 Nonlinear Additively Preconditioned Linesearch Methods

In 1995, O.L.. Mangasarian introduced the parallel gradient distribution (PGD), an asynchronous Linesearch algorithm [Man95]. The paradigm of the PGD method is to asynchronously compute local corrections s_k which serve as a starting point for the computation of a global update

$$s_k \in \mathcal{D}_k : J(u + I_k s_k) - J(u) \geq \rho_{PGD} \|\nabla J(u)\|_2^2 \tag{4.2.1a}$$

$$s \in \mathbb{R}^n : J(u + s) \leq \min_k J(u + I_k s_k) \tag{4.2.1b}$$

where $\rho_{PGD} > 0$. As a matter of fact, this algorithm can be regarded as a globalization strategy. Though, it is not clear, how to cheaply compute the sought-after correction s within (4.2.1b). Indeed, one might solve another nonconvex minimization problem to compute a set of damping parameters or one just employs the "best" correction s_k.
To avoid disposing $N - 1$ corrections and to avoid the solution of another complex minimization problem, respectively, we will consider the asynchronously computed search directions as a search

direction for the global problem. This allows for employing the traditional backtracking scheme to compute a Linesearch parameter as a solution of a scalar problem. Along with a priori assumptions and the subset objective function H_k^ν from (3.2.1) this allows for proving convergence of a clearly stated asynchronous Linesearch algorithm.

4.2.1 The APLS Framework

The algorithm of this section, the **A**dditively **P**reconditioned **L**inesearch **S**trategy, Algorithm 7, is the second implementation of the abstract additive preconditioning framework from Section 3.2. The APLS consists of three phases: an asynchronous solution phase, a recombination phase and a possible global Linesearch smoothing phase. Similar to the APTS algorithm, we are interested in the framework's efficiency (cf., Chapter 5.5) and robustness. Surprisingly, besides the actual algorithmic framework, it suffices to slightly extend the assumptions of Section 2.2 for proving convergence:

($A_{\text{apls}}1$) For the given initial global iterate $u^0 \in \mathcal{B}$, for all $\nu \geq 0$ and all initial iterates on $\mathcal{D}_k$, i.e., $u_{k,0}^\nu \in \mathcal{B}_k(u^\nu)$, it is assumed that the level sets

$$\mathcal{L}_G^0 = \{u \in \mathcal{B} \mid J(u) \leq J(u^0)\}$$

and

$$\mathcal{L}_k^\nu = \{u \in \mathcal{B}_k(u^\nu) \mid H_k^\nu(u) \leq H_k^\nu(u_{k,0}^\nu)\}$$

are nonempty and compact. Here H_k^ν is defined in (3.2.1) and $\mathcal{B}_k(u^\nu)$ is given by (3.2.8).

($A_{\text{apls}}2$) We assume that J is continuously differentiable on $\mathcal{L}_G^0$, as well as, for all $\nu \geq 0$ and all $k = 1, \ldots, N$ that H_k^ν is continuously differentiable on $\mathcal{L}_k^\nu$. Moreover, we assume that for all $u \in \mathcal{L}_G^0$ and $u_k \in \mathcal{L}_k^\nu$ the respective gradients are Lipschitz continuous with a constant $L_g > 0$, i.e.,

$$\|\nabla J(u) - \nabla J(u+s)\|_2 \leq L_g \|s\|_2$$

and

$$\|\nabla H_k^\nu(u_k) - \nabla H_k^\nu(u_k + s_k)\|_2 \leq L_g \|s_k\|_2$$

for $s \in \mathbb{R}^n$ such that $u + s \in \mathcal{B}$, and $s_k \in \mathcal{D}_k$ such that $u_k + s_k \in \mathcal{B}_k(u^\nu)$, respectively.

The Nonlinear Update Operator

Similar to the mechanisms of the previous section, the nonlinear operator $\mathcal{F}_k$ is the result of m iterations of the Linesearch algorithm, Algorithm 6, employed on $\mathcal{D}_k$. In this case, the Linesearch algorithm stops after computing $u_{k,m}^\nu$, the final iterate on this subset, which gives rise to the following definition

$$\mathcal{F}_k(P_k u^\nu) = u_{k,m}^\nu$$

where u^ν is the current global iterate. Therefore, we define the locally computed correction as

$$s_k^\nu = \mathcal{F}_k(P_k u^\nu) - P_k u^\nu = u_{k,m}^\nu - P_k u^\nu$$

Since we consider a Linesearch framework, a damping parameter within the recombination operator $\mathcal{F}_A$ in Lemma 3.2.1, now becomes active and corrections will always be damped and applied. Thus,

the APLS recombination operator is given by

$$\mathcal{A}^{\nu}_{\text{APLS}}(I_1 s^{\nu}_1, \dots, I_N s^{\nu}_N, u^{\nu}) = u^{\nu} + \alpha^{\nu} \sum_{k=1}^{N} I_k s^{\nu}_k$$

In particular, $\alpha^{\nu} \in (0, \alpha_0]$ with $\alpha_0 \leq 1$ will now be chosen such that the Armijo condition (2.2.5) is satisfied. Hence, the nonlinear additive update operator is given by

$$\mathcal{F}_A(u^{\nu}) = \mathcal{A}^{\nu}_{\text{APLS}}(I_1 s^{\nu}_1, \dots, I_N s^{\nu}_N, u^{\nu}) \tag{4.2.2}$$

Notation

We will employ the same notation as in the Trust-Region framework: the additive solution process will employ N instances of the Linesearch Algorithm 6. Afterwards we compute $m_G \geq 0$ global Linesearch steps to smooth globally. Therefore, ν counts the number of APLS cycles, k denotes the current context, i.e., $k \in \{1, \dots, N\}$ denotes a subset, $k = G$ the global post-smoothing context and the index i will count the Linesearch iterations.

4.2.2 A Modified Armijo Condition for the Additive Context

Similar to the APTS context, we need several assumptions on the algorithm. To this end, we extend the Armijo-Condition (2.2.5) to the additive preconditioning context yielding the following Armijo condition:

$$H^{\nu}_k(u^{\nu}_{k,i} + \alpha^{\nu}_{k,i} s_{k,i}) \leq H^{\nu}_k(u^{\nu}_{k,i-1}) + \rho_A \alpha^{\nu}_{k,i} \langle s_{k,i}, g^{\nu}_{k,i} \rangle \tag{4.2.3}$$

where $\rho_A \in (0, 1)$ (from (2.2.5)) and $g^{\nu}_{k,i} = \nabla H^{\nu}_k(u^{\nu}_{k,i})$. Moreover, on each subset, we demand that beginning from the second subset iteration the following inequality holds.

$$\langle u^{\nu}_{k,i} - u^{\nu}_{k,0} + \alpha^{\nu}_{k,i} s_{k,i}, g^{\nu}_{k,0} \rangle \leq \rho_R \langle u^{\nu}_{k,1} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \tag{4.2.4}$$

where $0 < \rho_R \leq \rho_A < 1$. Both conditions, (4.2.3) and (4.2.4), will now ensure that each subset search direction is a descent direction for J. On the other hand, condition (4.2.4) is weak enough, to leave space for an iterative minimization process of H^{ν}_k on $\mathcal{D}_k$.
The following lemma addresses the fact that, if the subset objective function's Hessians are positive semi-definite, assumption (4.2.4) is trivially satisfied if the Armijo condition holds.

Lemma 4.2.1. *Assume that ($A_{apls}1$) and ($A_{apls}2$) hold and that H^{ν}_k is twice continuously differentiable. Moreover assume that the Hessians of H^{ν}_k are positive semi-definite, i.e.,*

$$0 \leq \langle s, \nabla^2 H^{\nu}_k(u) s \rangle$$

for all $u \in \mathcal{D}_k$ and all $s \in \mathcal{D}_k : u + s \in \mathcal{B}_k(u^{\nu})$. Suppose furthermore that all search directions satisfy (2.2.2b) and (4.2.3). Then assumption (4.2.4) is satisfied if $\alpha^{\nu}_{k,i} \in (0, \alpha_0]$ satisfies (4.2.3).

Proof. We employ Taylor's theorem along with the positive semi-definiteness of the Hessians and obtain

$$H^{\nu}_k(u^{\nu}_{k,0} + s) - H^{\nu}_k(u^{\nu}_{k,0}) = \langle s, g^{\nu}_{k,0} \rangle + \frac{1}{2} \langle s, \nabla^2 H^{\nu}_k(\xi) s \rangle \geq \langle s, g^{\nu}_{k,0} \rangle$$

for all $s : u^{\nu}_{k,0} + s \in \mathcal{B}_k(u^{\nu})$. Here we employed $\xi = u^{\nu}_{k,0} + \tau s$ and $\tau \in (0, 1)$. Therefore the

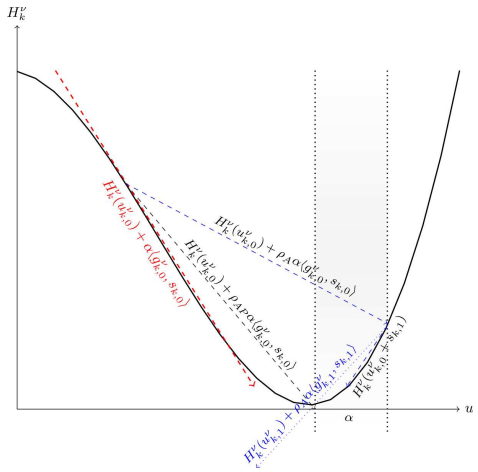

Figure 4.1: This figure illustrates, how the subset criterion (4.2.8) works. The initial Linesearch parameter $\alpha_{k,0}^\nu$ on $\mathcal{D}_k$ is chosen such that just the Armijo condition holds. Note that within the presented example, the gradient is a negative scalar and the initial search direction is a positive scalar. Therefore, the step–length constraint prevents moving back to $u_{k,0}^\nu$. On the other hand, the blue dotted line represents the Armijo condition. Both together yield the set of admissible step–lengths α, as indicated by the interval in between the dotted lines.

following inequality holds

$$H_k^\nu(u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i}) - H_k^\nu(u_{k,0}^\nu) \geq \langle u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i} - u_{k,0}^\nu, g_{k,0}^\nu\rangle \tag{4.2.5}$$

On the other hand, we employ the decrease condition (2.2.2b) and the Armijo condition (4.2.3) and obtain

$$0 \geq \rho_A\langle\alpha_{k,0}^\nu s_{k,0}, g_{k,0}^\nu\rangle \geq H_k^\nu(u_{k,0}^\nu + \alpha_{k,0}^\nu s_{k,0}^\nu) - H_k^\nu(u_{k,0}^\nu) \tag{4.2.6}$$

Thus, since in every iteration (4.2.3) holds, we obtain

$$\rho_A\langle u_{k,1}^\nu - u_{k,0}^\nu, g_{k,0}^\nu\rangle \geq H_k^\nu(u_{k,1}^\nu) - H_k^\nu(u_{k,0}^\nu) \geq \ldots \geq H_k^\nu(u_{k,i}^\nu) - H_k^\nu(u_{k,0}^\nu) \tag{4.2.7}$$

Thus, we may combine the inequalities (4.2.5), (4.2.7), the fact that in iteration i the Armijo condition holds and $\rho_R \leq \rho_A$ to

$$\begin{aligned}
\rho_R\langle u_{k,1}^\nu - u_{k,0}^\nu, g_{k,0}^\nu\rangle &\geq \rho_A\langle u_{k,1}^\nu - u_{k,0}^\nu, g_{k,0}^\nu\rangle \\
&\geq H_k^\nu(u_{k,1}^\nu) - H_k^\nu(u_{k,0}^\nu) \\
&\geq H_k^\nu(u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i}) - H_k^\nu(u_{k,0}^\nu) \\
&\geq \langle u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i} - u_{k,0}^\nu, g_{k,0}^\nu\rangle
\end{aligned}$$

which proves the proposition. □

A Practicable Descent Condition

Even if for convex functions assumption (4.2.4) may easily be satisfied, our aim is to introduce a practicable backtracking algorithm which is able to compute an appropriate step–length satisfying (4.2.4). To this end, we introduce the following, altered descent condition, which will substitute

Modified Linesearch Algorithm

Input: $u^{\nu}_{k,0} \in \mathcal{D}_k, \mathcal{B}_k, k \in \mathbb{N} \cup \{G\}, m, n_k \in \mathbb{N}$
Output: $u^{\nu}_{k,m} \in \mathcal{D}_k$

```
i = 0
do until (i = m) {
    if (D_k ≠ R^n and i > 0) {
        compute a search direction s_{k,i} satisfying (2.2.2a) and (4.2.10a) and u_{k,i} + s_{k,i} ∈ B_k
        call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,i} satisfying (4.2.3) and (4.2.8)
    } else if (D_k ≠ R^n) {
        compute a search direction s_{k,i} satisfying (2.2.2a) and (2.2.2b) and u_{k,i} + s_{k,i} ∈ B_k
        call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,i} satisfying (4.2.3)
    } else {
        compute a search direction s_{G,i} satisfying (2.2.2a) and (2.2.2b) and u_{G,i} + s_{G,i} ∈ B
        call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,i} satisfying (2.2.5)
    }
    set u^ν_{k,i+1} = u^ν_{k,i} + α^ν_{k,i} s^ν_{k,i}
    i = i + 1
}

return u^ν_{k,m}
```

Algorithm 6: Modified Linesearch Algorithm

(4.2.4) in our analysis and in the actual implementation of the APLS algorithm:

$$\langle u^{\nu}_{k,i} - u^{\nu}_{k,0} + \alpha^{\nu}_{k,i} s_{k,i}, g^{\nu}_{k,0} \rangle \leq \rho_{AP} \langle u^{\nu}_{k,i} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \tag{4.2.8}$$

where $1 > \rho_{AP} > 0$ is chosen such that $\rho_R \leq \rho^m_{AP}$. An illustration of this criterion is given in Figure 4.1. Moreover, it implies (4.2.4), since

$$\begin{aligned} \langle u^{\nu}_{k,i} - u^{\nu}_{k,0} + \alpha^{\nu}_{k,i} s_{k,i}, g^{\nu}_{k,0} \rangle &\leq \rho_{AP} \langle u^{\nu}_{k,i} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \leq \ldots \leq \\ &\leq \rho^{i-1}_{AP} \langle u^{\nu}_{k,1} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \\ &\leq \rho_R \langle u^{\nu}_{k,1} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle < 0 \end{aligned}$$

In particular, we obtain

$$\langle u^{\nu}_{k,m} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \leq \rho^m_{AP} \langle u^{\nu}_{k,1} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \leq \rho_R \langle u^{\nu}_{k,1} - u^{\nu}_{k,0}, g^{\nu}_{k,0} \rangle \tag{4.2.9}$$

The following lemma addresses the question, if the backtracking algorithm in Algorithm 6 is able to compute a Linesearch parameter α such that both conditions, (4.2.3) and (4.2.8) hold.

Lemma 4.2.2. *Assume that ($A_{apls}1$) and ($A_{apls}2$) hold. Suppose that $i > 0$ and all computed search directions on $\mathcal{D}_k$ were descent directions satisfying (2.2.2b) and that in each iteration condition (4.2.3) and (4.2.8) hold. Then, for a given descent direction $s_{k,i} \in \mathcal{D}_k$, there exists an $\alpha^{\nu}_{k,i} \leq \alpha_0 \leq 1$ such that $u^{\nu}_{k,i} + \alpha^{\nu}_{k,i} s_{k,i}$ satisfies (4.2.3) and (4.2.8).*

Proof. Since the ray $\rho_A \alpha^{\nu}_{k,0} \langle s_{k,0}, g^{\nu}_{k,0} \rangle$ lies for sufficiently small $\alpha^{\nu}_{k,0}$ above $H^{\nu}_k(u^{\nu}_{k,0} + \alpha^{\nu}_{k,0} s_{k,0}) - H^{\nu}_k(u^{\nu}_{k,0})$ and $s_{k,0}$ is a descent direction, we obtain that (4.2.3) holds.
Now, assume that $i > 0$. Moreover, since (2.2.2b) holds, we obtain that there exists an $\alpha' > 0$ such that for all $\alpha^{\nu}_{k,i} \in (0, \alpha']$ the Armijo condition, equation (4.2.3), holds. On the other hand, since

$\rho_{AP} < 1$, and since (4.2.8) holds for each computed iterate, we obtain

$$\langle u^\nu_{k-1,i} - u^\nu_{k,0}, g^\nu_{k,0}\rangle \leq \rho^{k-1}_{AP}\langle u^\nu_{1,i} - u^\nu_{k,0}, g^\nu_{k,0}\rangle$$

Therefore, for sufficiently small $\alpha^\nu_{k,i}$ also inequality (4.2.8) is satisfied, even if $\langle s_{k,i}, g^\nu_{k,0}\rangle > 0$. Together this proves the proposition. □

Similar to the argumentation for the APTS algorithm, additively computed corrections employing the Linesearch algorithm, Algorithm 3 are by Lemma 3.2.1 admissible corrections for $\mathcal{B}$. Moreover, like in the original Linesearch algorithm, we have to ensure that the lengths of the subset search directions are limited by the norm of the initial subset gradient. To this end, we introduce the following criterion

$$\|u^\nu_{k,i} - u^\nu_{k,0} + s_{k,i}\|^2_\infty \leq \beta_{ls}\|\hat{g}^\nu_{k,0}\|^2_\infty \tag{4.2.10a}$$

$$\|s_{k,i}\|^2_\infty \leq \beta_{ls}\|\hat{g}^\nu_{k,i}\|^2_\infty \tag{4.2.10b}$$

for all $i = 0, \ldots, m-1$ and $k = 1, \ldots, N$. Here $\beta_{ls} > 0$ is the constant from the initial step–length criterion (2.2.2a) and $\hat{g}^\nu_{k,i} = D^\nu_{k,i}\nabla H^\nu_k(u^\nu_{k,i})$. As a matter of fact, (4.2.10a) which gives rise to

$$\|u^\nu_{k,m} - u^\nu_{k,0}\|^2_\infty \leq \beta_{ls}\|\hat{g}^\nu_{k,0}\|^2_\infty$$

since the step–length parameter satisfies $0 < \alpha^\nu_{k,i} \leq 1$. Moreover, in order to handle both inequalities in (4.2.10) one might substitute (4.2.10) by

$$\|s^\nu_{k,i}\|^2_\infty \leq \min\{\beta_{ls}\|\hat{g}^\nu_{k,0}\|^2_\infty - \|u^\nu_{k,i} - u^\nu_{k,0}\|^2_\infty, \beta_{ls}\|\hat{g}^\nu_{k,i}\|^2_\infty\}$$

Now, we are able to introduce the APLS algorithm, Algorithm 7, which is an actual Linesearch implementation of the abstract additive framework of Chapter 3. By construction, this nonlinear solution strategy, combines a priori assumptions on the additively computed search directions, i.e., (4.2.8) and (4.2.10) to ensure that these search directions are descent directions for the objective function J from (M). As it turns out, together with the Armijo condition as a posteriori step–length control strategy, we are able to show that APLS is a globalization strategy.

4.2.3 Convergence to First–Order Critical Points

Like in Section 2.2, we will show that the APLS algorithm satisfies a sufficient decrease condition. This will be carried out by showing that the step–length parameters are bounded from below and by showing that the additive search directions satisfy a descent condition similar to (2.2.2b).

Lemma 4.2.3. *Suppose that ($A_{apls}1$) and ($A_{apls}2$) hold. Then for each global smoothing step and each initial subset step in Algorithm 7 the Armijo condition (4.2.3) is satisfied for*

$$2\alpha_{\min} = 2\frac{\eta_{ls}(1-\rho_A)}{nL_g\beta_{ls}}$$

Moreover, we obtain for the step–length $\alpha^\nu_{k,i}$ of each global smoothing step and each initial subset correction

$$\min\{\alpha_0, 2\tau\alpha_{\min}\} \leq \alpha^\nu_{k,i}$$

where, by definition, $\alpha_0 \leq 1$.

Algorithm: APLS – Nonlinear Additively Preconditioned Linesearch Algorithm

Input: $J : \mathbb{R}^n \to \mathbb{R}$, $\mathcal{B}$, $u^0 \in \mathbb{R}^n$, $n \in \mathbb{N}$
Constants: $m, m_G \in \mathbb{N}$

do {

Additive Preconditioning
On each subset k where (4.1.6) holds,
call the Linesearch algorithm, Algorithm 6, with $\underbrace{P_k u^\nu}_{=u^\nu_{k,0}}$, $\underbrace{\mathcal{B}_k(u^\nu)}_{=\mathcal{B}_k \text{ cf. (3.2.8)}}$, k, m, $\underbrace{\dim \mathcal{D}_k}_{=n_k}$

Global Smoothing
call Algorithm 6 with $\underbrace{\mathcal{F}_A(u^\nu)}_{=u^\nu_{G,0}}$ from (4.2.2), $\underbrace{\mathcal{B}}_{=\mathcal{B}_k}$, $\underbrace{G}_{=k}$, $\underbrace{m_G}_{=m}$, $\underbrace{\dim \mathcal{D}_k}_{=n_k}$

Iterate with $u^{\nu+1} = u^\nu_{G,m_G}$, $\nu = \nu + 1$
}

Algorithm 7: APLS – Nonlinear Additively Preconditioned Linesearch Algorithm

Proof. Due to the assumptions, the proof follows exactly the proof of Lemma 2.2.2 and Lemma 2.2.3. □

Lemma 4.2.4. *Assume that ($A_{apls}1$) and ($A_{apls}2$) hold. Then each additively computed correction s^ν in Algorithm 7 satisfies the descent condition*

$$-\langle g^\nu, s^\nu \rangle \geq \kappa_g^2 \rho_{AP}^m \eta_{ls} \min\{\alpha_0, 2\tau\alpha_{\min}\} \|\hat{g}^\nu\|_2^2$$

where $\alpha_{\min} = \frac{2\eta_{ls}(1-\rho_A)}{nL_g\beta_{ls}}$. Moreover we employed $s^\nu = \sum_{k \in C^\nu} I_k s_k^\nu$ and $C^\nu = \{k : s_k^\nu = u_{k,m}^\nu - u_{k,0}^\nu \neq 0\}$, i.e., the indices of the subsets, where corrections where computed.

Proof. Due to the definition of the subset objective function we have $g_{k,0}^\nu = \nabla H_k^\nu(u_{k,0}^\nu) = R_k g^\nu$. Now we use the definition of R_k and s^ν and obtain

$$\begin{aligned} -\langle g^\nu, s^\nu \rangle &= -\sum_{k \in C^\nu} \langle g^\nu, I_k(u_{k,m}^\nu - u_{k,0}^\nu) \rangle \\ &= -\sum_{k \in C^\nu} \langle R_k g^\nu, u_{k,m}^\nu - u_{k,0}^\nu \rangle \\ &= -\sum_{k \in C^\nu} \langle g_{k,0}^\nu, u_{k,m}^\nu - u_{k,0}^\nu \rangle \end{aligned}$$

Since each correction is a descent direction which satisfies (4.2.8), we employ (4.2.9) which gives rise to

$$-\langle g^\nu, s^\nu \rangle \geq -\rho_{AP}^m \sum_{k \in C^\nu} \langle g_{k,0}^\nu, \alpha_{k,0}^\nu s_{k,0} \rangle$$

Now we may apply Lemma 4.2.3 and obtain

$$-\langle g^\nu, s^\nu \rangle \geq -\rho_{AP}^m \min\{\alpha_0, 2\tau\alpha_{\min}\} \sum_{k \in C^\nu} \langle g_{k,0}^\nu, s_{k,0} \rangle$$

Since each (initial) search direction on $\mathcal{D}_k$ satisfies (2.2.2b) and since $\|\hat{g}^\nu_{k,0}\|_2^2$ satisfies (4.1.6), i.e., the relationship to $\|\hat{g}^\nu\|_2^2$, we have

$$\begin{aligned} -\langle g^\nu, s^\nu\rangle &\geq \rho_{AP}^m \eta_{ls} \min\{\alpha_0, 2\tau\alpha_{\min}\} \sum_{k\in C^\nu} \|\hat{g}^\nu_{k,0}\|_2^2 \\ &\geq \kappa_g^2 \rho_{AP}^m \eta_{ls} \min\{\alpha_0, 2\tau\alpha_{\min}\} \|\hat{g}^\nu\|_2^2 \end{aligned}$$

□

Lemma 4.2.5. *Assume that (A$_{apls}$1) and (A$_{apls}$2) hold. Then for each Linesearch parameter $\alpha^\nu_{G,0}$ computed in Algorithm 6, to rescale the additively computed search direction in Algorithm 7, we obtain*

$$\alpha^\nu_{G,0} \geq \min\{\alpha_0, 2\tau c_{APLS}\alpha_{\min}\}$$

where

$$c_{APLS} = \alpha_{\min}\frac{\rho_{AP}^m}{nC_I}\min\{\alpha_0, 2\tau\alpha_{\min}\}$$

where $C_I = \sum_{k\in C^\nu}\|I_k\|_2^2$ and $\alpha_{\min}$ from Lemma 4.2.4.

Proof. Under the assumptions of this lemma, we may apply Lemma 2.2.2 and obtain

$$\alpha^\nu_{G,0} \geq \min\left\{\alpha_0, 2\tau\frac{2(\rho_A - 1)\langle s^\nu, g^\nu\rangle}{nL_g\|s^\nu\|_2^2}\right\} \tag{4.2.11}$$

By Lemma 4.2.4, we have

$$-\langle s^\nu, g^\nu\rangle \geq \kappa_g^2\rho_{AP}^m\eta_{ls}\min\{\alpha_0, 2\tau\alpha_{\min}\}\|\hat{g}^\nu\|_2^2$$

On the other hand, (4.2.10a) and the relationship between the gradients, equation (4.1.6), give rise to

$$\|s^\nu_k\|_\infty^2 = \|u^\nu_{k,m} - u^\nu_{k,0}\|_\infty^2 \leq \beta_{ls}\|\hat{g}^\nu_{k,0}\|_\infty^2 \leq \beta_{ls}\kappa_g^2\|g^\nu\|_2^2$$

Now, we employ this inequality and obtain

$$\begin{aligned} \|s^\nu\|_2^2 &= \|\sum_{k\in C^\nu} I_k s^\nu_k\|_2^2 \leq \sum_{k\in C^\nu}\|I_k\|_2^2\|s^\nu_k\|_2^2 \\ &\leq n\sum_{k\in C^\nu}\|I_k\|_2^2\|s^\nu_k\|_\infty^2 \\ &\leq n\beta_{ls}C_I\kappa_g^2\|\hat{g}^\nu\|_2^2 \end{aligned}$$

where $C_I = \sum_{k\in C^\nu}\|I_k\|_2^2$. Now, we combine the estimates for $\|s^\nu\|_2^2$ and $\langle s^\nu, g^\nu\rangle$ and obtain

$$\alpha^\nu_{G,i} \geq \min\left\{\alpha_0, 2\tau\frac{2(1-\rho_A)\eta_{ls}}{nL_g\beta_{ls}}\frac{\rho_{AP}^m}{nC_I}\min\{\alpha_0, 2\tau\alpha_{\min}\}\right\}$$

which proves the proposition. □

Lemma 4.2.6. *Let assumptions (A$_{apls}$1) and (A$_{apls}$2) hold, and suppose that $\|\hat{g}^\nu\|_2 > 0$. Moreover assume that (4.1.7) holds in the case of an overlapping domain decomposition or (4.1.8) for a non-*

overlapping domain decomposition. Then we obtain

$$\mathcal{C}^\nu \neq \emptyset$$

where $\mathcal{C}^\nu = \{k : u^\nu_{k,m} \neq u^\nu_{k,0}\}$ *is the set of computed subset corrections in Algorithm 7.*

Proof. Similar to the proof of Lemma 4.1.5, we have to prove that in each iteration at least on one subset a correction is computed. This means, in the context of Linesearch methods, that we have to prove that one $\mathcal{D}_k$ satisfies (4.1.6). Due to the assumptions of this lemma, the argumentation employed in (4.1.13) and (4.1.14) holds and therefore (4.1.6) is satisfied at least on one domain. □

In a similar fashion like Theorem 2.2.4, we will prove the convergence of the nonlinear additively preconditioned Linesearch algorithm, Algorithm 7.

Theorem 4.2.7. *Suppose that (A$_{apls}$1) and (A$_{apls}$2) hold. Assume furthermore that either the domain is overlappingly or non-overlappingly decomposed with constants from (4.1.7) and (4.1.8) or that* $m_G > 0$ *global post-smoothing Linesearch steps are computed. Then the APLS algorithm, Algorithm 7, computes a sequence of iterates converging to a first–order critical point for problem (M), i.e.,*

$$\lim_{\nu\to\infty, i\in\{0,\dots,m+1\}} \|\hat{g}^\nu_{G,i}\|_2 = 0$$

Proof. As in the proof of Theorem 2.2.4, we use that each global step–length parameter α_i satisfies the Armijo condition (4.2.3), i.e.,

$$J(u^\nu_{G,i}) - J(u^\nu_{G,i+1}) \geq -\alpha^\nu_{G,i}\rho_A\langle s_{G,i}, g^\nu_{G,i}\rangle$$

Using, Lemma 4.2.4 and (2.2.2b), respectively, gives,

$$J(u^\nu_{G,i}) - J(u^\nu_{G,i+1}) \geq \begin{cases} \alpha^\nu_{G,i}\left(\kappa_g^2\rho_{AP}^m\eta_{ls}\min\{\alpha_0, 2\tau\alpha_{\min}\}\|\hat{g}^\nu_{G,i}\|_2^2\right) & \text{if } s_{G,i} \text{ was comp. additively} \\ \alpha^\nu_{G,i}\left(\eta_{ls}\|\hat{g}^\nu_{G,i}\|_2^2\right) & \text{otherwise} \end{cases}$$

Now we employ Lemma 2.2.3 and Lemma 4.2.5 which gives

$$\min\{\alpha_0, 2\tau\alpha_{\min}, 2\tau c_{\text{APLS}}\alpha_{\min}\} \leq \alpha^\nu_{G,i}$$

Note that if $m_G = 0$ we have due to Lemma 4.2.6 that $C^\nu \neq \emptyset$ for all ν, as long as $\|\hat{g}^\nu\|_2 \neq 0$. Together with the compactness of $\mathcal{L}^0_G$ and, thus, $J(u^\nu_{G,i}) - J(u^\nu_{G,i+1}) \to 0$ we can conclude that $\|\hat{g}^\nu_{G,i}\|_2 \to 0$ which proves the proposition.

□

4.3 A Remark on Parallel Communication

As we have seen in this chapter, the APTS and APLS methods basically split into three phases:

1. asynchronous local solution phase
2. recombination phase
3. global post-smoothing phase

As a matter of fact, in the first step communication does not take place. In particular, the only communication is needed within the recombination operator $\mathcal{A}$ which combines the interpolated search-directions and computes the decrease ratio or Linesearch parameter, respectively. If one discretizes the PDEs employing Finite Elements with linear basis functions, the computation of the decrease ratio or Linesearch parameter is extremely cheap, since quadrature can perfectly be parallelized. Therefore, only within the global post–smoothing parallel communication must periodically take place. In turn, the overall parallel communication of additively preconditioned globalization strategies and the traditional ones is more or less the same.

4.4 A Remark on Second-Order Convergence

In the context of Trust-Region methods, T. Coleman and Y. Li have shown in [CL96], that second–order convergence can generally only be ensured by computing the "right" corrections as was outlined in Section 2.1.5. In fact, if the Hessian is indefinite but the gradient is zero, one succeeded in computing a saddle point. To compute a local minimizer, though, a correction must satisfy a stronger decrease condition as, for instance,

$$\psi^{\nu}_{G,0}(s^{\nu}) \leq c\psi^{\nu}_{G,0}(s_{\min}) \text{ such that } \|s^{\nu}\|_{\infty} \leq \Delta^{\nu} \text{ and } u^{\nu} + s^{\nu} \in \mathcal{B} \tag{4.4.1}$$

where $c > 0$ and

$$\psi^{\nu}_{G,0}(s) = \langle g^{\nu}, s\rangle + \frac{1}{2}\langle s, \nabla^2 J(u^{\nu})s\rangle$$

The solution of this quadratic minimization problem is given by

$$\psi^{\nu}_{G,0}(s_{\min}) = \min\{\psi^{\nu}_{G,0}(s) \mid u^{\nu} + s \in \mathcal{B},\ \|s\|_{\infty} \leq \Delta^{\nu}\}$$

As it turns out, the additively computed correction

$$s^{\nu} = \sum_{k\in C^{\nu}} I_k s^{\nu}_k = \sum_{k\in C^{\nu}} I_k(u^{\nu}_{k,m} - u^{\nu}_{k,0})$$

can hardly solve (4.4.1) without leaving the asynchronous setting. Suppose that the local objective function are quadratic functions given by

$$H^{\nu}_k(u^{\nu}_{k,0} + s_k) = \psi^{\nu}_k(s_k) = \langle R_k g^{\nu}, s_k\rangle + \frac{1}{2}\langle s_k, R_k\nabla^2 J(u^{\nu})I_k s_k\rangle$$

Furthermore, suppose that each local minimization problem is solved exactly, i.e.,

$$\begin{aligned}\psi^{\nu}_k(s_k) &= \langle R_k g^{\nu}, s_k\rangle + \frac{1}{2}\langle s_k, R_k\nabla^2 J(u^{\nu})I_k s_k\rangle \\ &= \min\{\psi^{\nu}_k(s) \mid P_k u^{\nu} + s \in \mathcal{B}_k(u^{\nu}),\ \|s\|_k \leq \Delta^{\nu}\}\end{aligned}$$

In this case, we obtain after interpolating and summing up

$$\psi^{\nu}_{G,0}(\sum_k I_k s_k) = \sum_k \psi^{\nu}_k(s_k) + \underbrace{\sum_{k\neq i}\langle s_k, R_k\nabla^2 J(u^{\nu})I_i s_i\rangle}_{\text{Coupling terms}}$$

As it turns out, the coupling terms may now yield that the exact solution of the quadratic minimization problem differs from the additively computed. Suppose that we have an non-overlapping domain decomposition and that each $g_k = 0$ and that $R_k \nabla^2 J(u^\nu) I_k$ is positive definite. Moreover suppose that each $R_k \nabla^2 J(u^\nu) I_i$ is negative definite. In this case, the complete Hessian $\nabla^2 J(u^\nu)$ might be negative definite, as in the following example

$$\nabla^2 J(u^\nu) = \begin{pmatrix} 2 & -4 \\ -4 & 2 \end{pmatrix}$$

Then, either way the solution of the local minimization problems is $s_k = 0$ yielding $s^\nu = 0$. But, as a matter of fact, the minimizer of (4.4.1) for such a Hessian may be[2]

$$s_{\min} = \alpha x_{\lambda_{\min}}$$

where $x_{\lambda_{\min}}$ with $\|x_{\lambda_{\min}}\|_2 = 1$ is the eigenvector related to the smallest (negative) eigenvalue of $\nabla^2 J(u^\nu)$. The scaling parameter $\alpha > 0$ is the maximal possible step–length such that $u^\nu + s_{\min} \in \mathcal{B}$ and $\|s_{\min}\|_\infty \leq \Delta^\nu$ holds. Therefore we obtain

$$\langle s^\nu, \nabla^2 J(u^\nu) s^\nu \rangle = 0$$

but

$$\langle \alpha x_{\lambda_{\min}}, \nabla^2 J(u^\nu) \alpha x_{\lambda_{\min}} \rangle = \lambda_{\min} \alpha < 0$$

In this case, the additive corrections will without further assumptions not be the solution of (4.4.1). Note that this argumentation even holds for more complex subset objective functions since subspace correction methods, like the presented, are generally not able to resolve all eigenvalues of the Hessian.

As outlined in Section 2.2.5, in [CL94] it was shown that also Linesearch strategies are able to resolve the second–order conditions. Then, similarly to Trust-Region strategies, the search direction must then satisfy (4.4.1). But, as we have just proven, without further assumptions, the (4.4.1) can generally not be satisfied by additively computed corrections.

Therefore, second–order convergence and, perhaps, quadratic convergence rates may just be provided by the global Trust-Region or Linesearch algorithm. In turn, the additive preconditioning strategy aims at adaptively determining step–lengths and is designed for massive parallel computing.

[2] Due to the Trust-Region constraint one may also linearly combine different eigenvectors.

5 Nonlinear Multiplicatively Preconditioned Globalization Strategies

The solution of discretized elliptic PDEs is due to complexity considerations often carried out employing iterative solvers like, for instance, the cg method. In fact, the rates of convergence of the cg method depends on the condition number of the stiffness matrix, which is, in the case of a Finite Element discretization, closely related to the number of unknowns. As a matter of fact, the better the basis functions resolve high frequency contributions the worse become the rates of convergence of the iterative solver. In other words, slower convergence is often connected to the resolution of low-frequency contributions by single basis functions.
Thus, in order to improve the resolution of low-frequency contributions several (preconditioning-) techniques were developed, such as Wavelets (for an introduction see [Dah97]) or multigrid methods (for an introduction see [Bra07]). In Finite Element methods, multigrid strategies prevailed as a preconditioner for the cg method. Here, on each grid, a smoother computes a solution for a local linear system of equations, which yields a coarse level correction. This correction is then interpolated and employed within the next cg iteration. As it turns out, this method has optimal complexity and is an optimal preconditioner.
For the parallel solution of positive definite linear systems of equations often the use of a *coarse grid* is suggested (cf. the monograph [TW05]) to improve the rates of convergence. Here, on one processor a coarse problem is solved. This particularly enables mathematicians to prove logarithmic dependence between the condition number of the preconditioned stiffness matrix and the mesh size.
Also in the nonlinear case, the application of multilevel strategies seems to be reasonable. Intuitively, in a first step one may employ a linear multigrid strategy to solve the occurring quadratic minimization problems. But, depending on the Hessian or its approximation, respectively, the convergence of the linear preconditioner can not be guaranteed. Moreover, limited step–sizes remain as a problem, even if the search directions are computed faster.
In contrast to just applying a better solution strategy for the quadratic minimization problems, S.Nash [Nas00] introduced MG/OPT, a nonlinear solution strategy, which

- attacks nonlinear low-frequency contributions of the problem
- is guaranteed to converge to first–order critical points

Since we consider a nonlinear multigrid strategy, the initial iterate is given based on the current fine level iterate. Therefore, for instance, in [Nas00] it was proposed to employ the restriction operator to transfer the fine level iterate to the coarse level. Though, as we have pointed out in Section 3.1.3, this may cause numerical problems. Therefore, to speed up convergence of the RMTR strategy, Gratton et al. propose to damp the computed corrections in the RMTR strategy.
An interesting feature of the RMTR method is that the algorithm is allowed to stay on a coarse level until the coarse level problem is solved approximately. But, this has a particular draw-back: the algorithm must leave the coarse level, if the first–order conditions become small and the algorithm must not go into a recursion, if the gradient is too small. Both is needed to make the algorithm *computable*. But, as the theory of linear multigrid methods shows, the (desired) asymptotic convergence behavior

crucially depends on the recursions. On the other hand, the convergence of Gauß-Seidel-like multiplicative schemes (cf. Section 3.3.3) can not be guaranteed, if the subset problems are just solved approximately.
In order to show convergence without constraints on the first–order conditions, in [GK08b] it was shown that it suffices to ensure that a limited number of iterations takes place on the subsets. Therefore, based on this RMTR variant, we will introduce a generalized nonlinear multiplicatively preconditioned Trust-Region strategy for problems of the kind (M). This method implements the abstract framework of Section 3.3 and can therefore be employed, for instance, as a multigrid or Gauß-Seidel-like scheme. Moreover, the abstract multiplicative preconditioning framework allows for proving convergence without assuming that eventually the recursions stop. Also a nonlinear multiplicatively preconditioned Linesearch method will be presented in this chapter. Here, we will follow Z. Wen's and D. Goldfarb's point of view and regard the multiplicatively computed correction as a search–direction for the fine–level problem. However, due to significant weaker a priori assumptions, our approach is applicable to the non–smooth context of problem (M).
As we will see, the analysis for multiplicative Trust-Region and Linesearch methods is similar to the one presented in the additive context. This has the advantage, that one can easily deduce global convergence properties for a combined, nonlinear additively and multiplicatively preconditioned strategy, attacking both, nonlinear low–frequency contributions and local nonlinearities.

5.1 Nonlinear Multiplicatively Preconditioned Trust-Region Methods

As for the Trust-Region context in general, we are interested in controlling the Trust-Region radius by means of the local nonlinearity of the objective function. Since we consider the multiplicative context where the local objective function may or must be chosen different from the original objective function J, the Trust-Region radius on the current subset must depend on the nonlinearity of the local objective function and on the nonlinearity of the preceding objective functions. In turn, the coupling of the objective functions and a "global" control of the Trust-Region radius allows for proving convergence to critical points.

5.1.1 The MPTS Framework

Following the setting of the abstract formulation of Section 3.3, we decompose $\mathbb{R}^n$ into a sequence of spaces $(\mathcal{D}_0, \ldots, \mathcal{D}_N)$ such that there exist projection, interpolation and restriction operators between each of the respective spaces.
Similar to the assumptions on the previous Trust-Region algorithms, we will state once more assumptions on the respective objective functions.

($A_{\text{mpts}}1$) For the given initial iterate $u^0 \in \mathbb{R}^n$, for all $\nu \geq 0$ and all subsets $k \in \{1, \ldots, N\}$ with $\mathcal{D}_k \neq \mathbb{R}^n$ and all initial subset iterates $u^\nu_{k,0} = P^k_{k-1} u_{k-1} \in \mathbb{R}^{n_k}$, where $u_{k-1} \in \mathcal{D}_{k-1}$ is admissible, it is assumed that the level sets

$$\mathcal{L}^0 = \{u \in \mathcal{B} \mid J(u) \leq J(u^0)\}$$

and

$$\mathcal{L}^\nu_k = \{u \in \mathcal{B}_k(u_{k-1}) \mid H^\nu_k(u) \leq H^\nu_k(u^\nu_{k,0})\}$$

are nonempty and compact. Here H^ν_k is from (3.3.1) and $\mathcal{B}_k(u_{k-1})$ is from (3.3.2).

($A_{mpts}2$) We assume that J is continuously differentiable on $\mathcal{L}^0$ and that for all $\nu \geq 0$ and $k = 1, \dots, N$, $\mathcal{D}_k \neq \mathbb{R}^n$ the function H_k^ν is continuously differentiable on $\mathcal{L}_k^\nu$. Moreover, there exists a constant $C_g > 0$ such that

$$\begin{aligned} \|\nabla J(u)\|_2 &\leq C_g \text{ for all } u \in \mathcal{L}^0 \\ \|\nabla H_k^\nu(u_k)\|_2 &\leq C_g \text{ for all } u_k \in \mathcal{L}_k^\nu \end{aligned}$$

for all $\nu \geq 0$, $k = 1, \dots, N$, $\mathcal{D}_k \neq \mathbb{R}^n$.

($A_{mpts}3$) We assume that there exists a constant $C_B > 0$ such that for all $B(u)$ approximating $\nabla^2 J(u)$ and $B_k(u_k^\nu)$ approximating $\nabla^2 H_k^\nu(u_k^\nu)$ in (2.1.1) the following holds

$$\begin{aligned} \|B(u)\|_2 &\leq C_B \text{ for all } u \in \mathcal{L}^0 \\ \|B_k(u_k)\|_2 &\leq C_B \text{ for all } u_k \in \mathcal{L}_k^\nu \end{aligned}$$

for all $\nu \geq 0$, $k = 1, \dots, N$, $\mathcal{D}_k \neq \mathbb{R}^n$.

The multiplicatively preconditioned Trust-Region strategy, Algorithm 8, now implements the multiplicative framework of Section 3.3. Similar to Algorithm 4, this framework may remain on the current subset, call a recursion or return the current iterate. In order to ensure convergence, we assume that in every computation on each subset, $m_k \leq m$ Trust-Region smoothing steps are computed, where $m = \max_{k=0,\dots,N} m_k$.

Remark 5.1.1. *As it turns out, the presented analysis, in particular the convergence results, hold if*

- *on one $\mathcal{D}_k = \mathbb{R}^n$ at least either $m_k = m_G > 0$ Trust-Region steps are computed or we have a domain decomposition as in Section 3.1.6 with $m_k > 0$ for $\mathcal{D}_k \neq \mathbb{R}^n$*
- *on each subset at most m Trust-Region steps are computed*

We will see that similarly to the additive context, the Trust-Region radius also depends on the decrease ratio induced by the subset correction. Thus, also in the present framework it might happen, that the computations on one subset dominate the whole computation. This means in particular that the global Trust-Region radius is too large to ensure that global corrections are applied. But, as we will see, the global Trust-Region radius also depends on the local corrections. If now, computations on $\mathcal{D}_k$ are always successful and applied, the global radius stays too large and convergence will not be achieved. Thus, we introduce the following, slightly modified advance criterion

$$\|\hat{g}_{k+1,0}^\nu\|_2 \geq \kappa_g \|\hat{g}_{k,m_k}^\nu\|_2 \tag{5.1.1}$$

Here, m_k is the index when calling the recursion, $\kappa_g \in (0,1)$ and $\hat{g}_{k,i}^\nu = D_{k,i}^\nu g_{k,i}^\nu = D_k^\nu(u_{k,i}^\nu)\nabla H_k^\nu(u_{k,i}^\nu)$ with D_k^ν as defined in Section 2.1.3.

The Multiplicative Update Operator

Similar to the additive context, we define the subset correction as the difference between initial and final iterate on the succeeding subset $\mathcal{D}_k \supset \mathcal{D}_{k+1}$, i.e.,

$$s_{k,m_k} = I_{k+1}^k s_k^\nu = I_{k+1}^k (u_{k+1,m_{k+1,f}}^\nu - u_{k+1,0}^\nu) \tag{5.1.2}$$

Algorithm: MPTS – Multiplicatively Preconditioned Trust-Region Strategy

Input: $k \in \{0, \dots, N\}, \mathcal{B}_k, u^\nu_{k,0}, R^k_{k-1} g^\nu_{k-1,m_{k-1}} \in \mathcal{D}_{k-1}, \Delta_{k-1} \in \mathbb{R}^+ \cup \{\infty\}$
Output: new iterate $u^\nu_{k,m_{k,f}} \in \mathcal{D}_k$

Smoothing
call Algorithm 1 with $\underbrace{u^\nu_{k,0}}_{=u_0}, \underbrace{H^\nu_k}_{=J}, \underbrace{\Delta^\nu_{k,0}}_{=\Delta_0}, \underbrace{\dim \mathcal{D}_k}_{=n}$
with modified constant m_k, modified constraint (5.1.7) and Trust-Region update (5.1.6).
receive a new iterate u^ν_{k,m_k} and a new trust–region radius Δ^ν_{k,m_k}

if $(C_{k+1} \supsetneq C_k)$
return $u^\nu_{k,m_{k,f}} = u^\nu_{k,m_k}$

else if $(C_{k+1} = C_k)$ {
$u^\nu_{k+1,0} = u^\nu_{k,m_k}$, $\Delta^\nu_{k+1,0} = \Delta^\nu_{k,m_k}$
$k = k + 1$
goto *Smoothing*

} **else if** $(C_{k+1} \subsetneq C_k)$ {
if ((5.1.1) holds on D_{k+1}) {
call *MPTS* with $k+1$, $\underbrace{\mathcal{B}_{k+1}(u^\nu_{k,m_k})}_{=\mathcal{B}_{k+1} \text{ cf. (3.3.2)}}$, $\underbrace{P^{k+1}_k u^\nu_{k,m_k}}_{=u^\nu_{k+1,0}}$, $R^{k+1}_k g^\nu_{k,m_k}$, $\underbrace{\Delta^\nu_{k,m_k}}_{=\Delta_k}$
and receive $u^\nu_{k+1,m_{k+1,f}}$
set $s_{k,m_k} = I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - P^{k+1}_k u^\nu_{k,m_k})$
$u^\nu_{k,m_k+1} = \mathcal{A}^\nu_{\text{MPTS},k}(s_{k,m_k}, u^\nu_{k,m_k})$
update Δ^ν_{k,m_k} according to (5.1.6)
}
if $(k \neq 0)$
$u^\nu_{r_k,0} = u^\nu_{k,m_k+1}$, $\Delta^\nu_{r_k,0} = \Delta^\nu_{k,m_k+1}$, $k = r_k$
else
$u^{\nu+1}_{0,0} = u^\nu_{k,m_k+1}$, $\Delta^{\nu+1}_{0,0} = \Delta^\nu_{k,m_k+1}$, $\nu = \nu + 1$

goto *Smoothing*
}

Algorithm 8: MPTS – Nonlinear Multiplicatively Preconditioned Trust-Region Strategy

where $m_{k+1,f}$ denotes the index of the final iterate on $\mathcal{D}_{k+1}$. In particular, we define $u_{k+1,m_{k+1,f}} = u^\nu_{l,m_l}$ with $r_k = l + 1$.

Now, we follow [GST08] and define the decrease ratio for the subset corrections as

$$\rho^\nu_{k,m_k} = \begin{cases} \dfrac{H^\nu_k(u^\nu_{k,m_k}) - H^\nu_k(u^\nu_{k,m_k} + s_{k,m_k})}{H^\nu_{k+1}(u^\nu_{k+1,0}) - H^\nu_{k+1}(u^\nu_{k+1,m_{k+1,f}})} & \text{if } u^\nu_{k+1,0} \neq u^\nu_{k+1,m_{k+1,f}} \\ 0 & \text{otherwise} \end{cases} \tag{5.1.3}$$

Hence, ρ^ν_{k,m_k} compares the reduction of the preceding subset objective function value to the one on the current subset. Moreover, similarly to the additive context, ρ^ν_{k,m_k} allows for proving a sufficient decrease of H^ν_k, if a sufficient decrease can be shown for H^ν_{k+1}. Therefore, this quantity is used to accept or reject a subset correction and to adjust the Trust-Region radius. Thus, we may define the

subset-dependent nonlinear multiplicative recombination operators $\mathcal{A}^{\nu}_{\text{MPTS},k}$ as

$$\mathcal{A}^{\nu}_{\text{MPTS},k}(s_{k,m_k}, u^{\nu}_{k,m_k}) = \begin{cases} u^{\nu}_{k,m_k} + s_{k,m_k} & \text{if } \rho^{\nu}_{k,m_k} \geq \eta \\ u^{\nu}_{k,m_k} & \text{otherwise} \end{cases}$$

Now, we can define the nonlinear multiplicative update operator as

$$\mathcal{F}^{(j)}_{M}(u^{\nu}_{j,m_j}) = \mathcal{A}^{\nu}_{\text{MPTS},j}(s_{j,m_j}, u^{\nu}_{j,m_j}) \tag{5.1.4}$$

for all j with $\mathcal{D}_j = \mathbb{R}^n$. Here, we have to distinguish between different update operators, since from every $D_j = \mathbb{R}^n$ a (differently looking) recursion might be called, as for instance in the context of the Gauß-Seidel method in Section 3.3.3.
We define the subset obstacles as in (3.3.2). Therefore, following Lemma 3.3.1, it suffices to ensure that each subset iterate is admissible to obtain admissible multiplicatively computed corrections. Since this is the case by construction of Algorithm 8, Lemma 3.3.1 holds which gives rise to $u^{\nu}_{k,i} + I^{k}_{k+1} s^{\nu}_{k+1} \in \mathcal{B}^{\nu}_{k}$.

The Local Trust-Region Update in the Multiplicative Context

To ensure that the multiplicatively computed corrections are scaled according to the nonlinearity of the respective preceding objective functions, we modify the Trust-Region update (2.1.4) to fit into the multiplicative context. Similar to the additive context, we compute an intermediate Trust-Region radius as follows

$$\tilde{\Delta}^{\nu}_{k,i} \in \begin{cases} (\Delta^{\nu}_{k,i}, \gamma_2 \Delta^{\nu}_{k,i}] & \text{if } \rho^{\nu}_{k,i} \geq \eta \\ [\gamma_1 \Delta^{\nu}_{k,i}, \Delta^{\nu}_{k,i}) & \text{if } \rho^{\nu}_{k,i} < \eta \end{cases} \tag{5.1.5}$$

In a second step, we either use the intermediate Trust-Region radius or the previous one to update $\Delta^{\nu}_{k,i}$ as given in the following equality

$$\Delta^{\nu}_{k,i+1} = \begin{cases} \min\{\tilde{\Delta}^{\nu}_{k,i}, \Delta_{k-1} - \|u^{\nu}_{k,i+1} - u^{\nu}_{k,0}\|_k\} & \text{if } \mathcal{D}_k \neq \mathbb{R}^n \\ \tilde{\Delta}^{\nu}_{k,i} & \text{otherwise} \end{cases} \tag{5.1.6}$$

Here Δ_{k-1} is the current Trust-Region radius on the preceding subset, $\mathcal{D}_{k-1}$. Moreover, we employed the following – from the additive context known – multilevel norm

$$\|s_k\|_k = \|I_k s_k\|_\infty$$

where I_k is as defined in (3.1.11). In Lemma 5.1.2, we will see that together with the modified Trust-Region constraint

$$\|s_{k,i}\|_k \leq \Delta^{\nu}_{k,i} \tag{5.1.7}$$

substituted in (2.1.2), this formulation ensures that each correction which is computed on $\mathcal{D}_k$ stays within the previous subset's Trust–Region.

5.1.2 Convergence to First-Order Critical Points

We emphasize that the functions J^{ν}_{k} are not required to be twice continuously differentiable to obtain convergence to first–order critical points. However, the proof of Theorem 5.1.5 crucially depends on the fact that only finitely many Trust-Region iterations are carried out at each subset. However, to

prevent that the algorithm spends too much time for solving subset minimization problems, it seems to be reasonable to limit the number of subset Trust-Region steps a priori.
Similar to Lemma 4.1 in [GST08], we show that the subset corrections will not violate the global Trust-Region constraint.

Lemma 5.1.2. *For all ν, k and i and each $s_{k,m_k} = I_{k+1}^k(u_{k+1,m_{k+1,f}}^\nu - u_{k+1,0}^\nu) \in \mathcal{D}_k$ computed recursively in algorithm MPTS the following holds*

$$\|s_{k,m_k}\|_k \leq \Delta_{k,m_k}^\nu \tag{5.1.8}$$

Proof. We prove the proposition by induction.
Assume that $\mathcal{D}_k$ is a subset from where no recursion is called. Then we obtain for each iterate $u_{k,i}^\nu \neq u_{k,i-1}^\nu$

$$\begin{aligned} \|u_{k,i}^\nu - u_{k,0}^\nu\|_k &\leq \|u_{k,i-1}^\nu - u_{k,0}^\nu\|_k + \|s_{k,i}^\nu\|_k \\ &\leq \|u_{k,i-1}^\nu - u_{k,0}^\nu\|_k + \Delta_{k-1} - \|u_{k,i-1}^\nu - u_{k,0}^\nu\|_k = \Delta_{k-1} = \Delta_{k-1,m_{k-1}}^\nu \end{aligned}$$

Therefore, equation (5.1.8) is satisfied. Moreover, we have

$$\|s_{k-1,m_{k-1}}\|_{k-1} = \|I_k^{k-1}(u_{k,m_{k,f}}^\nu - u_{k,0}^\nu)\|_{k-1} = \|u_{k,m_{k,f}}^\nu - u_{k,0}^\nu\|_k \leq \Delta_{k-1} = \Delta_{k-1,m_{k-1}}^\nu$$

Now, assume that $\mathcal{D}_k$ is not the deepest subset. Due to the update formula of the Trust-Region radius (5.1.6) we may now employ the just used argumentation and obtain that on subset $\mathcal{D}_k$ each recursively computed correction satisfies equation (5.1.8). □

Lemma 5.1.3. *Let assumptions ($A_{mpts}1$), ($A_{mpts}2$) and ($A_{mpts}3$) hold and suppose that the correction $s_{k,i}$ in iteration ν is computed recursively. Moreover assume that $\|\hat{g}_{k,i}^\nu\|_2 \neq 0$. Then there exists a constant $c_{rsd} = c(\gamma_1, \eta, \kappa_g, N) > 0$ such that the following sufficient decrease condition holds*

$$H_k^\nu(u_{k,i}^\nu) - H_k^\nu(u_{k,i}^\nu + s_{k,i}) \geq c_{rsd}\|\hat{g}_{k,i}^\nu\|_2 \min\{\Delta_{k,i}^\nu, \|\hat{g}_{k,i}^\nu\|_2\} \tag{5.1.9}$$

Similar to the proof of Lemma 4.3 [GK08b], we follow the recursion to the deepest subset, where the *first* Trust-Region correction was computed and propagated. Down to this subset, we derive estimations for the first–order conditions and the Trust-Region radius. Both will relate the respective local entities to the entities on subset $\mathcal{D}_k$. Together with an estimation of the local decrease, this yields the sought–after sufficient decrease estimation.

Proof. First, we analyze a successful recursion beginning at subset $\mathcal{D}_k$. Successful means that

$$H_k^\nu(u_{k,m_k}^\nu) - H_k^\nu(u_{k,m_k}^\nu + s_{k,m_k}) \geq \eta(H_{k+1}^\nu(u_{k+1,0}^\nu) - H_{k+1}^\nu(u_{k+1,m_{k+1,f}}^\nu)) \tag{5.1.10}$$

holds. This implies that there must have been a subset $l > k$ and an iteration r such that there has a *first* successful Trust-Region correction $s_{l,r}$ been applied and *propagated* to subset $\mathcal{D}_k$. Hence, this implies that also the gradient did not change on subset $\mathcal{D}_r$ before applying $s_{l,r}$. This means

$$\|\hat{g}_{l,0}^\nu\|_2 = \|\hat{g}_{l,r}^\nu\|_2$$

Using equation (5.1.1), $1 > \kappa_g > 0$, and that there exist at most N subsets before returning to the global context yields

$$\|\hat{g}_{l,r}^\nu\|_2 \geq (\kappa_g)^{l-k}\|\hat{g}_{k,m_k}^\nu\|_2 \geq (\kappa_g)^N\|\hat{g}_{k,m_k}^\nu\|_2 \tag{5.1.11}$$

To derive a lower bound for the Trust-Region radius $\Delta^\nu_{l,r}$, we suppose that the Trust-Region radii were only reduced on their propagation to subset $\mathcal{D}_l$ and to iteration r. Note that at most $m = \max_{k=0,\dots,N} m_k$ Trust-Region steps are computed on each subset. Therefore, equation (5.1.6), $\gamma_1 < 1$, $r \leq m+1$ for all l provide

$$\begin{aligned}\Delta^\nu_{l,r} &\geq \gamma_1^r \Delta^\nu_{l,0} = \gamma_1^r \Delta^\nu_{l-1,m_{l-1}} \\ &\geq \gamma_1^{(2\cdot(m+1))} \Delta^\nu_{l-2,m_{l-2}} \\ &\geq \gamma_1^{((N+1)\cdot(m+1))} \Delta^\nu_{k,m_k} > 0\end{aligned}$$

Now, since $u^\nu_{l,r+1}$ was computed in the Trust-Region algorithm, Algorithm 1, we may employ $\rho^\nu_{l,r} \geq \eta$ and the sufficient decrease condition (2.1.7) and obtain

$$H^\nu_l(u^\nu_{l,0}) - H^\nu_l(u^\nu_{l,r+1}) \geq \eta\beta \|\hat{g}^\nu_{l,r}\|_2 \min\{\Delta^\nu_{l,r}, \|\hat{g}^\nu_{l,r}\|_2\} \tag{5.1.12a}$$

$$\geq \eta\beta(\kappa_g)^N \|\hat{g}^\nu_{k,m_k}\|_2 \min\{\gamma_1^{((N+1)\cdot(m+1))}\Delta^\nu_{k,m_k}, (\kappa_g)^N\|\hat{g}^\nu_{k,m_k}\|_2\} \tag{5.1.12b}$$

We still need to estimate the left hand side of this inequality by $H^\nu_k(u^\nu_{k,m_k}) - H^\nu_k(u^\nu_{k,m_k} + s_{k,m_k})$. We obtain that at all subsets p with $k > p \geq l$ and iterations i at subset p the following inequality holds

$$H^\nu_p(u^\nu_{p,m_p} + s_{p,m_p}) = H^\nu_p(u^\nu_{p,m_p+1}) \geq H^\nu_p(u^\nu_{p,m_{p,f}}) \tag{5.1.13}$$

where s_{p,m_p} is the recursively computed correction. The acceptance criterion for recursively computed corrections (5.1.10) and (5.1.13) imply

$$\begin{aligned}H^\nu_p(u^\nu_{p,0}) - H^\nu_p(u^\nu_{p,m_{p,f}}) &\geq H^\nu_p(u^\nu_{p,m_p}) - H^\nu_p(u^\nu_{p_p,m+1}) \\ &\geq \eta(H^\nu_{p+1}(u^\nu_{p+1,0}) - H^\nu_{p+1}(u^\nu_{p+1,m_{p+1,f}}))\end{aligned}$$

Using this inequality, (5.1.10), the choice of $\eta < 1$, and the fact that maximal N recursions take place yields

$$\begin{aligned}H^\nu_k(u^\nu_{k,m_k}) - H^\nu_k(u^\nu_{k,m_k} + s_{k,m_k}) &\geq \eta^N(H^\nu_l(u^\nu_{l,0}) - H^\nu_l(u^\nu_{l,m_{l,f}})) \\ &\geq \eta^N(H^\nu_l(u^\nu_{l,0}) - H^\nu_l(u^\nu_{l,r} + s_{l,r})) \\ &= \eta^N(H^\nu_l(u^\nu_{l,0}) - H^\nu_l(u^\nu_{l,r+1}))\end{aligned}$$

Combining this inequality with equation (5.1.12) yields

$$H^\nu_k(u^\nu_{k,m_k}) - H^\nu_k(u^\nu_{k,m_k} + s_{k,m_k}) \geq c_{\text{rsd}} \|\hat{g}^\nu_{k,m_k}\|_2 \min\{\Delta^\nu_{k,m_k}, \|\hat{g}^\nu_{k,m_k}\|_2\} \tag{5.1.14}$$

where $c_{\text{rsd}} = \beta\eta^{N+1}\kappa_g^{2N}\gamma_1^{((N+1)\cdot(m+1))} > 0$. This concludes the proof. □

Lemma 5.1.4. *Let assumptions ($A_{mpts}1$), ($A_{mpts}2$) and ($A_{mpts}3$) hold and suppose that $\|\hat{g}_k(u^\nu_{k,i})\|_2 \geq \varepsilon > 0$. Then we obtain for the decrease ratio of each recursively computed correction (5.1.3) in Algorithm 8*

$$\rho^\nu_{k,m_k} \geq \eta$$

for Δ^ν_{k,m_k} sufficiently small.

Proof. Due to the definition of the initial Trust-Region radius on $\mathcal{D}_{k+1}$ and the considerations in

Lemma 2.1.2 we have that, for Δ^{ν}_{k,m_k} sufficiently small, each Trust-Region correction on $\mathcal{D}_{k+1}$ is successful, i.e., $\rho^{\nu}_{k+1,i} \geq \eta$.

Now we consider the definition of the recursive decrease ratio (5.1.3), i.e.,

$$\rho^{\nu}_{k,m_k} = \frac{H^{\nu}_k(u^{\nu}_{k,m_k}) - H^{\nu}_k(u^{\nu}_{k,m_k} + s_{k,m_k})}{H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,m_{k+1,f}})} \tag{5.1.15}$$

Here, s_{k,m_k} is the multiplicatively computed correction given by

$$s_{k,m_k} = I^k_{k+1} s^{\nu}_{k+1} = I^k_{k+1}(u^{\nu}_{k+1,m_{k+1,f}} - u^{\nu}_{k+1,0})$$

Employing the meanvalue theorem allows for rewriting the numerator in (5.1.15) as follows

$$\begin{aligned} H^{\nu}_k(u^{\nu}_{k,m_k}) - H^{\nu}_k(u^{\nu}_{k,m_k} + I^k_{k+1} s^{\nu}_{k+1}) &= -\langle \nabla H^{\nu}_k(\xi^{\nu}_k), I^k_{k+1} s^{\nu}_{k+1} \rangle \\ &= -\langle R^{k+1}_k \nabla H^{\nu}_k(\xi^{\nu}_k), s^{\nu}_{k+1} \rangle \end{aligned} \tag{5.1.16}$$

with $\xi^{\nu}_k = u^{\nu}_{k,m_k} + \tau^{\nu}_k I^k_{k+1} s^{\nu}_{k+1}$ and $\tau^{\nu}_k \in (0,1)$. Now, we obtain

$$\rho^{\nu}_{k,m_k} = \frac{-\langle R^{k+1}_k \nabla H^{\nu}_k(\xi^{\nu}_k), s^{\nu}_{k+1} \rangle}{H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,m_{k+1,f}})} \tag{5.1.17}$$

We will have a closer look at the denominator. The mean value theorem and the definition of the subset objective functions H^{ν}_k from (3.3.1) provide for sufficiently small Δ^{ν}_{k,m_k} the following inequality

$$\begin{aligned} 0 &< H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1}) \\ &= J^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - J^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1}) - \langle \delta g^{\nu}_{k+1}, s^{\nu}_{k+1} \rangle \\ &= -\langle \nabla J^{\nu}_{k+1}(\xi^{\nu}_{k+1}), s^{\nu}_{k+1} \rangle - \langle R^{k+1}_k \nabla J^{\nu}_k(u^{\nu}_{k,m_k}) - \nabla J^{\nu}_{k+1}(u^{\nu}_{k+1,0}), s^{\nu}_{k+1} \rangle \end{aligned} \tag{5.1.18}$$

where $\xi^{\nu}_{k+1} = u^{\nu}_{k+1,0} + \tau^{\nu}_{k+1} s^{\nu}_{k+1}$ and $\tau^{\nu}_{k+1} \in (0,1)$. To reformulate (5.1.17), we add $\pm(H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1}))$ and obtain

$$\begin{aligned} \rho^{\nu}_{k,m_k} &= \frac{-(H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1})) - \langle R^{k+1}_k \nabla H^{\nu}_k(\xi^{\nu}_k), s^{\nu}_{k+1} \rangle}{H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1})} + 1 \\ &= -\frac{|\kappa_1| + |\kappa_2|}{H^{\nu}_{k+1}(u^{\nu}_{k+1,0}) - H^{\nu}_{k+1}(u^{\nu}_{k+1,0} + s^{\nu}_{k+1})} + 1 \end{aligned}$$

Here, we used (5.1.16) and (5.1.18) to obtain the following abbreviations

$$\begin{aligned} \kappa_1 &= \langle \nabla J^{\nu}_{k+1}(\xi^{\nu}_{k+1}) - \nabla J^{\nu}_{k+1}(u^{\nu}_{k+1,0}), s^{\nu}_{k+1} \rangle \\ \kappa_2 &= \langle R^{k+1}_k \nabla H^{\nu}_k(u^{\nu}_{k,m_k}) - R^{k+1}_k \nabla H^{\nu}_k(\xi^{\nu}_k), s^{\nu}_{k+1} \rangle \end{aligned}$$

Next, we derive estimations for $|\kappa_1|$ and $|\kappa_2|$. We employ Cauchy-Schwarz's inequality and obtain

$$\begin{aligned} |\kappa_1| &= |\langle \nabla J^\nu_{k+1}(\xi^\nu_{k+1}) - \nabla J^\nu_{k+1}(u^\nu_{k+1,0}), s^\nu_{k+1}\rangle| \\ &\leq \|\nabla J^\nu_{k+1}(\xi^\nu_{k+1}) - \nabla J^\nu_{k+1}(u^\nu_{k+1,0})\|_2 \|s^\nu_{k+1}\|_2 \\ |\kappa_2| &= |\langle R^{k+1}_k \nabla H^\nu_k(u^\nu_{k,m_k}) - R^{k+1}_k \nabla H^\nu_k(\xi^\nu_k), s^\nu_{k+1}\rangle| \\ &\leq \|R^{k+1}_k\|_2 \|\nabla H^\nu_k(u^\nu_{k,m_k}) - \nabla H^\nu_k(\xi^\nu_k)\|_2 \|s^\nu_{k+1}\|_2 \end{aligned}$$

Since ∇J^ν_k and ∇J^ν_{k+1} are continuous on a compact set, we obtain uniform continuity of both functions. In particular, for all $\varepsilon_C > 0$ there exists an $\Delta_C > 0$ such that for all $\Delta^\nu_{k,i} \leq \Delta^\nu_{k,m_k} \leq \Delta_C$, the following holds

$$\begin{aligned} -|\kappa_1| &\geq -\|\nabla J^\nu_{k+1}(\xi^\nu_{k+1}) - \nabla J^\nu_{k+1}(u^\nu_{k+1,0})\|_2 \|s^\nu_{k+1}\|_2 \geq -\varepsilon_C \|s^\nu_{k+1}\|_2 \\ -|\kappa_2| &\geq -\|R^{k+1}_k\|_2 \|\nabla H^\nu_k(u^\nu_{k,m_k}) - \nabla H^\nu_k(\xi^\nu_k)\|_2 \|s^\nu_{k+1}\|_2 \geq -\varepsilon_C C_R \|s^\nu_{k+1}\|_2 \end{aligned}$$

Here, we exploited (3.1.13a), i.e., $\|R^{k+1}_k\|_2 \leq C_R$. Assume now, that l denotes the first successful correction at subset $\mathcal{D}_{k+1}$. Hence, we employ $H^\nu_{k+1}(u^\nu_{k+1,0} + s_{k+1,l}) \geq H^\nu_{k+1}(u^\nu_{k+1,0} + s^\nu_{k+1})$ which gives rise to

$$\begin{aligned} \rho^\nu_{k,m_k} &\geq \frac{-\varepsilon_C \|s^\nu_{k+1}\|_2 - \varepsilon_C C_R \|s^\nu_{k+1}\|_2}{H^\nu_{k+1}(u^\nu_{k+1,0}) - H^\nu_{k+1}(u^\nu_{k+1,0} + s^\nu_{k+1})} + 1 \\ &\geq \frac{-\varepsilon_C \|s^\nu_{k+1}\|_2 - \varepsilon_C C_R \|s^\nu_{k+1}\|_2}{H^\nu_{k+1}(u^\nu_{k+1,0}) - H^\nu_{k+1}(u^\nu_{k+1,0} + s_{k+1,l})} + 1 \end{aligned} \tag{5.1.19}$$

Using the result of Lemma 5.1.3 (the MPTS sufficient decrease condition) yields for sufficiently small $\Delta^\nu_{k+1,l}$

$$\rho^\nu_{k,m_k} \geq \frac{-\varepsilon_C \|s^\nu_{k+1}\|_2 - \varepsilon_C C_R \|s^\nu_{k+1}\|_2}{c_{\text{rsd}} \|\hat{g}^\nu_{k+1,l}\|_2 \min\{\Delta^\nu_{k+1,l}, \|\hat{g}^\nu_{k+1,l}\|_2\}} + 1 \tag{5.1.20}$$

$$\geq \frac{-\varepsilon_C \|s^\nu_{k+1}\|_2 - \varepsilon_C C_R \|s^\nu_{k+1}\|_2}{c_{\text{rsd}} \varepsilon \Delta^\nu_{k+1,l}} + 1 \tag{5.1.21}$$

Where l is the first successful correction at $\mathcal{D}_{k+1}$. Now, we can apply Lemma 5.1.2 and (3.1.13b) which provides

$$\Delta^\nu_{k+1,l} \geq \gamma_1^l \Delta^\nu_{k+1,0} \geq \gamma_1^{N(m+1)} \Delta^\nu_{k,m_k} \geq \gamma_1^{N(m+1)} \|u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0}\|_{k+1} \tag{5.1.22a}$$

$$= \gamma_1^{N(m+1)} \|s^\nu_{k+1}\|_{k+1} \geq c_I \gamma_1^{N(m+1)} \|s^\nu_{k+1}\|_2 \tag{5.1.22b}$$

Moreover, we use (5.1.1), $\|\hat{g}^\nu_{k,m_k}\|_2 \geq \varepsilon$ and obtain

$$\|\hat{g}^\nu_{k+1,l}\|_2 = \|\hat{g}^\nu_{k+1,0}\|_2 \geq \kappa_g \|\hat{g}^\nu_{k,m_k}\|_2 \geq \kappa_g \varepsilon \tag{5.1.23}$$

Combining equation (5.1.23), (5.1.22a) and (5.1.21) provides for sufficiently small Δ^ν_{k,m_k}

$$\begin{aligned} \rho^\nu_{k,m_k} &\geq \frac{-\varepsilon_C \|s^\nu_{k+1}\|_2 - \varepsilon_C C_R \|s^\nu_{k+1}\|_2}{c_{\text{rsd}} \kappa_g \varepsilon \gamma_1^{N(m+1)} c_I \|s^\nu_{k+1}\|_2} + 1 \\ &= \frac{-(1 + C_R)\varepsilon_C}{c_{\text{rsd}} \kappa_g \varepsilon \gamma_1^{N(m+1)} c_I} + 1 \end{aligned}$$

Thus, choosing ε_C and Δ^ν_{k,m_k} small enough yields

$$\rho^\nu_{k,m_k} \geq \eta$$

which proves the proposition. □

Theorem 5.1.5. *Let assumptions ($A_{mpts}1$), ($A_{mpts}2$) and ($A_{mpts}3$) hold. Moreover assume that either $m_j = m_G > 0$ Trust-Region steps are computed on at least one subset $\mathcal{D}_j = \mathbb{R}^n$ or an overlapping or nonoverlapping domain decomposition is employed. If a domain decomposition is employed, we assume furthermore that $m_k > 0$ holds for $\mathcal{D}_k \neq \mathbb{R}^n$ and that the constants are given as in (4.1.7) and (4.1.8), respectively. Then we obtain that the MPTS algorithm, Algorithm 8, computes a sequence of iterates such that*

$$\liminf_{\nu\to\infty, \mathcal{D}_j=\mathbb{R}^n, i\in\{0,m+1\}} \|\hat{g}^\nu_{j,i}\|_2 = 0$$

Proof. We prove the assertion by contradiction.
Assume that there exists an index $\nu_0 > 0$ and $\varepsilon > 0$ such that $\|\hat{g}^\nu_{j,i}\|_2 \geq \varepsilon$ for all $\nu \geq \nu_0$, $\mathcal{D}_j = \mathbb{R}^n$ and all $i = 0, \ldots, m+1$. We will show, that this assumption implies that $\Delta^\nu_{j,i} \to 0$ for $\nu \to \infty$ and, in turn, $\rho^\nu_{j,i} \to 1$ which contradicts $\Delta^\nu_{j,i} \to 0$.
First, we will prove that $\Delta^\nu_{j,i} \to 0$ for $\nu \to \infty$. If there is only a finite number of successfully computed corrections we have due to the definition of $\Delta^\nu_{j,i}$ that $\Delta^\nu_{j,i} \to 0$.
On the other hand, if the sequence of successful corrections on $D_j = \mathbb{R}^n$ is infinitely long, equation (2.1.7) and (5.1.9) imply for successful corrections

$$J(u^\nu_{j,i}) > J(u^\nu_{j,i+1})$$

for all $l \geq 0$ and, therefore, we obtain due to ($A_{mpts}1$) that

$$J(u^\nu_{j,i}) - J(u^\nu_{j,i+1}) \to 0$$

The sufficient decrease condition in Lemma 5.1.3 and $\|\hat{g}^\nu_{j,i}\|_2 \geq \varepsilon$ give rise to

$$\Delta^\nu_{j,i} \to 0$$

Now, Lemma 5.1.4 and Lemma 2.1.2, respectively, provide that for sufficiently small $\Delta^\nu_{j,i}$ each correction is successful. But,

- if $m_G > 0$, this would yield that eventually all corrections are successful and therefore that $(\Delta^\nu_{j,i})_{\nu,i}$ is bounded from below.

- if $m_G = 0$, we have a domain decomposition and Lemma 4.1.5 applies. For sufficiently small $\Delta^\nu_{j,i}$ this yields that in each iteration ν the set $C^\nu = \{\mathcal{D}_j = \mathbb{R}^n \mid s^\nu_{j+1} \neq 0, C_{j+1} \subsetneq C_j\}$ is non-empty. In particular, we have that for sufficiently small $\Delta^\nu_{j,i}$ each recursively computed correction is successful. Therefore, due to the Trust-Region update, we obtain that $(\Delta^\nu_{j,i})_{\nu,i}$ is bounded from below.

In turn, the assumption must be wrong. □

Theorem 5.1.6. *Let the assumptions of the previous theorem hold. Then the MPTS algorithm, Algorithm 8, computes a sequence of iterates converging to a first–order critical point of (M), i.e.,*

$$\lim_{\nu\to\infty,\mathcal{D}_j=\mathbb{R}^n,i\in\{0,..,m+1\}} \|\hat{g}^{\nu}_{j,i}\|_2 = 0 \tag{5.1.24}$$

Proof. This proof is the same like the one of Theorem 2.1.4, except that we must substitute the sufficient decrease condition in inequality (2.1.15) by the weaker condition (5.1.9). □

5.2 Combined Nonlinearly Preconditioned Trust-Region Methods

At least for quadratic and convex minimization problems, often one of the drawbacks of pure additive preconditioning strategies, compared to multigrid strategies, are significantly slower rates of convergence. In particular, depending on the domain decomposition, additive preconditioners suffer from a delay of "information transfer" of the linear residual's low frequency contributions. To deal with this drawback, often a coarse space is employed to improve the rates of convergence (for an introduction see [TW05]).
Since problem (M) is arbitrarily nonlinear, we propose to employ the MPTS method to resolve low frequency contributions of the nonlinear residual. On each subset, the MPTS solver itself may employ, besides the Trust-Region smoother, the APTS strategy to solve the respective minimization problems in parallel.
Thus, in this section, we assume that a *vertical* domain decomposition of $\mathbb{R}^n$ exists, as introduced in Section 3.1.5. This means, that after decomposing $\mathbb{R}^n$ hierarchically, each of the resulting subsets $\mathcal{D}_k$ will also be decomposed *horizontally*, i.e.,

$$\bigcup_l \mathcal{D}_{k,l} = \mathcal{D}_k \tag{5.2.1}$$

Moreover, we assume that on each subset $\mathcal{D}_k$ assumptions ($A_{mpts}1$)–($A_{mpts}3$) hold, as well as assumptions ($A_{apts}1$)-($A_{apts}3$) on each subset $\mathcal{D}_{k,l}$. In this case, the MPTS algorithm, Algorithm 8, may now employ the APTS strategy, Algorithm 5, to solve the local minimization problems. Thus, the APTS method substitutes the Trust-Region strategy yielding the combined, nonlinear additively and multiplicatively preconditioned Trust-Region strategy, Algorithm 9.

Convergence to First–Order Critical Points

Similar to the additive and multiplicative contexts, we have to ensure that the APTS algorithm is able to compute the first-order conditions. To this end, either a domain decomposition as in Section 3.1.6 must be employed or at least one Trust-Region step on $\mathcal{D}_k$ must in each iteration be computed. However, since the convergence of all, within the AMPTS algorithm employed, globalization strategies has been proven, the proof of the following theorem is confined to proving that certain sufficient decrease conditions hold.

Theorem 5.2.1. *Assume that $\mathbb{R}^n$ is decomposed into a sequence of nested subspaces, as introduced in Section 3.1.4 and that the respective subsets $\mathcal{D}_k$ are decomposed as in (5.2.1). Furthermore, suppose that ($A_{mpts}1$)–($A_{mpts}3$) hold on each "multiplicative" subset $\mathcal{D}_k$ and that ($A_{apts}1$), ($A_{apts}2$) and ($A_{apts}3$) hold for the respective subspaces $\mathcal{D}_{k,l}$. Moreover suppose that either each subset $\mathcal{D}_j = \mathbb{R}^n$ is decomposed employing an overlapping or non-overlapping domain decomposition or that $m_G > 0$ global smoothing steps within the APTS algorithm are computed. If a domain decomposition is*

Algorithm: AMPTS – Nonlinear Additive and Multiplicative Preconditioned Trust-Region Strategy

Input: $k \in \{0, \dots, N\}, \mathcal{B}_k, u^\nu_{k,0}, g^\nu_{k-1} \in \mathcal{D}_{k-1}, \Delta_{k-1} \in \mathbb{R}^+ \cup \{\infty\}$
Output: new iterate $u^\nu_{k,m_{k,f}} \in \mathcal{D}_k$

Smoothing
call the *APTS* algorithm with $\underbrace{u^\nu_{k,0}}_{=u_0}, \underbrace{H^\nu_k}_{=J}, \underbrace{\Delta^\nu_{k,0}}_{=\Delta_0}, \underbrace{\dim \mathcal{D}_k}_{=n}$
and receive a new iterate u^ν_{k,m_k} and a new Trust-Region radius Δ^ν_{k,m_k}
if $(C_{k+1} \supsetneq C_k)$
return $u^\nu_{k,m_{k,f}} = u^\nu_{k,m_k}$

else if $(C_{k+1} = C_k)$ {
$u^\nu_{k+1,0} = u^\nu_{k,m_k}, \Delta^\nu_{k+1,0} = \Delta^\nu_{k,m_k}$
$k = k + 1$
goto *Smoothing*

} **else if** $(C_{k+1} \subsetneq C_k$ **and** (5.1.1) holds) {
call *AMPTS* with $k+1, \underbrace{\mathcal{B}_{k+1}(u^\nu_{k,m_k})}_{=\mathcal{B}_{k+1} \text{ cf. (3.3.2)}}, \underbrace{P^{k+1}_k u^\nu_{k,m_k}}_{=u^\nu_{k+1,0}}, \underbrace{R^{k+1}_k g^\nu_{k,m_k}}_{g^\nu_k}, \underbrace{\Delta^\nu_{k,m_k}}_{=\Delta_k}$
and receive $u^\nu_{k+1,m_{k+1,f}}$
set $s_{k,m_k} = I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - P^{k+1}_k u^\nu_{k,m_k})$
$u^\nu_{k,m_k+1} = \mathcal{A}^\nu_{\text{MPTS},k}(s_{k,m_k}, u^\nu_{k,m_k})$, update Δ^ν_{k,m_k} according to (5.1.6)

if $(k \neq 0)$
$u^\nu_{r_k,0} = u^\nu_{k,m_k+1}, \Delta^\nu_{r_k,0} = \Delta^\nu_{k,m_k+1}, k = r_k$
else
$u^{\nu+1}_{0,0} = u^\nu_{k,m_k+1}, \Delta^{\nu+1}_{0,0} = \Delta^\nu_{k,m_k+1}, \nu = \nu + 1$

goto *Smoothing*
}

Algorithm 9: AMPTS – Nonlinear Additive and Multiplicative Preconditioned Trust-Region Strategy

employed, we assume furthermore that $m_{j,l} > 0$ holds for the subsets $\mathcal{D}_{j,l}$ of $\mathcal{D}_j = \mathbb{R}^n$ and that the constants are given as in (4.1.7) and (4.1.8), respectively.
Then the sequence of iterates $(u^\nu_{j,i})_{i,\mathcal{D}_j=\mathbb{R}^n,\nu}$, computed in Algorithm 9, satisfies

$$\liminf_{\nu\to\infty,\mathcal{D}_j=\mathbb{R}^n,i\in\{0,\dots,m+1\}} \|\hat{g}(u^\nu_{j,i})\|_2 = 0 \tag{5.2.2}$$

Proof. We start with gathering some important results. Under the assumptions of this theorem and $\|\hat{g}^\nu_{k,i}\|_2 \neq 0$, Lemma 4.1.3 from Section 4.1.2 holds. Therefore, if the APTS strategy additively computes corrections on $\mathcal{D}_k$, i.e., $\mathcal{F}^{(k)}_A(u^\nu_{k,0}) \neq u^\nu_{k,0}$, we obtain the following decrease condition

$$\begin{aligned} H^\nu_k(u^\nu_{k,0}) - H^\nu_k(\mathcal{F}^k_A(u^\nu_{k,0})) &\geq \eta^2\beta \sum_{l\in\mathcal{C}^\nu_k} \left(\kappa_g\|\hat{g}^\nu_{k,0}\|_2 \min\left\{\kappa_g\|\hat{g}^\nu_{k,0}\|_2, \gamma_1^m\Delta^\nu_{k,0}\right\}\right) \\ &\geq \eta^2\beta\kappa_g\|\hat{g}^\nu_{k,0}\|_2 \min\left\{\kappa_g\|\hat{g}^\nu_{k,0}\|_2, \gamma_1^m\Delta^\nu_{k,0}\right\} \end{aligned}$$

Here, $\hat{g}^\nu_{k,0}$ denotes the gradient at $u^\nu_{k,0}$, the iterate before calling the APTS method on $\mathcal{D}_k$. Moreover, $\mathcal{C}^\nu_k$ is the set of each $\mathcal{D}_{k,l}$ where a correction was computed. Since this decrease condition is weaker[1]

[1] η and κ_g are smaller than one

than the original decrease condition (2.1.8), we now obtain that for all corrections $s_{k,i}$ on $\mathcal{D}_k$ the following decrease condition holds

$$H_k^\nu(u_{k,i}^\nu) - H_k^\nu(u_{k,i}^\nu + s_{k,i}) \geq \eta^2 \beta \kappa_g \|\hat{g}_{k,i}^\nu\|_2 \min\left\{\kappa_g \|\hat{g}_{k,i}^\nu\|_2, \gamma_1^m \Delta_{k,i}^\nu\right\} \tag{5.2.3}$$

Therefore, replacing (5.1.12b) in Lemma 5.1.3 by (5.2.3), yields for all recursively computed corrections

$$H_k^\nu(u_{k,i}^\nu) - H_k^\nu(u_{k,i}^\nu + s_{k,i}) \geq c_{\text{ampts}} \|\hat{g}_{k,i}^\nu\|_2 \min\{\Delta_{k,i}^\nu, \|\hat{g}_{k,i}^\nu\|_2\} \tag{5.2.4}$$

Where we introduce the following constant,

$$c_{\text{ampts}} = \beta \eta^{N+3} \kappa_g^{N+2} \gamma_1^{(N+1)\cdot m} \min\{\kappa_g^2, \kappa_g \gamma_1^m\} > 0$$

Since (5.2.4) is – once more – weaker than (5.2.3), this inequality is a valid sufficient decrease condition for all computed corrections.
Now suppose that the assumption of this theorem does not hold, i.e., for all $\nu > 0$, $i \in \{0, \ldots, m+1\}$ and each $\mathcal{D}_j = \mathbb{R}^n$ there exists an $\varepsilon > 0$ such that $\|\hat{g}(u_{j,i}^\nu)\|_2 > \varepsilon$.
If we now employ the same argumentation like in Theorem 2.1.3, we obtain that due to the sufficient decrease condition (5.2.4) the Trust-Region radius $\Delta_{j,i}^\nu$ tends to zero. Now, Lemma 2.1.2, Lemma 4.1.4 and Lemma 5.1.4 directly imply that eventually all corrections – Trust-Region corrections, additively and multiplicatively computed corrections – are successful on $\mathcal{D}_j = \mathbb{R}^n$, i.e., $\rho_{j,i}^\nu \geq \eta$ which contradicts $\Delta_{j,i}^\nu \to 0$ and proves the proposition. □

Theorem 5.2.2. *Suppose that the assumptions of Theorem 5.2.1 hold. Then, we obtain that*

$$\lim_{\nu \to \infty, \mathcal{D}_j = \mathbb{R}^n, i \in \{0,\ldots,m+1\}} \|\hat{g}(u_{j,i}^\nu)\|_2 = 0 \tag{5.2.5}$$

Proof. The proof is similar to the one of Theorem 2.1.4, except that the sufficient decrease conditions must be replaced by (5.2.4) on each subset $\mathcal{D}_j = \mathbb{R}^n$. □

5.3 Nonlinear Multiplicatively Preconditioned Linesearch Methods

In contrast to Trust-Region algorithms in general, the Linesearch framework has the advantage that each search–direction, as far as it satisfies a decrease condition, is scaled and applied. This means that due to well–balanced a priori and a posteriori descent control strategies no computation time is wasted for computing corrections which finally will not be applied. Thus, also in the context of multiplicatively preconditioned globalization strategies, it would be desirable to rescale all computed corrections, rather than disposing them which might lead to faster convergence.

5.3.1 The MPLS Framework

Similar to the APLS framework, we will state assumptions on the respective subset objective functions. But, even if the following assumptions look equivalent to the ones in Section 4.2.1, the employed objective functions are now the multiplicative objective functions H_k^ν from (3.3.1), Section 3.3.1. This gives rise to a recursive formulation of the traditional Linesearch assumptions.

($A_{\text{mpls}}1$) For the given initial global iterate $u^0 \in \mathcal{B}$, for all $\nu \geq 0$, all subsets $\mathcal{D}_k$ and all initial iterates on $\mathcal{D}_k \neq \mathbb{R}^n$, i.e., $u_{k,0}^\nu = P_{k-1}^k u_{k-1} \in \mathcal{B}_k(u_{k-1})$, and $u_{k-1} \in \mathcal{D}_{k-1}$ is admissible, it

is assumed that the level sets

$$\mathcal{L}^0 = \{u \in \mathcal{B} \mid J(u) \leq J(u^0)\}$$

and

$$\mathcal{L}_k^\nu = \{u \in \mathcal{B}_k(u_{k-1}) \mid H_k^\nu(u) \leq H_k^\nu(u_{k,0}^\nu)\}$$

are nonempty and compact. Here H_k^ν is from (3.3.1) and $\mathcal{B}_k(u_{k-1})$ is from (3.3.2).

($A_{mpls}2$) We assume that J is continuously differentiable on $\mathcal{L}^0$, as well as, for all $\nu \geq 0$ and all subsets $\mathcal{D}_k \neq \mathbb{R}^n$ that H_k^ν is continuously differentiable on $\mathcal{L}_k^\nu$. Moreover, we assume that for all $u \in \mathcal{L}^0$ and $u_k \in \mathcal{L}_k^\nu$ the respective gradients are Lipschitz continuous with a constant $L_g > 0$, i.e.,

$$\|\nabla J(u) - \nabla J(u+s)\|_2 \leq L_g \|s\|_2$$

and

$$\|\nabla H_k^\nu(u_k) - \nabla H_k^\nu(u_k + s_k)\|_2 \leq L_g \|s_k\|_2$$

for $u + s \in \mathcal{B}$, $u_k + s_k \in \mathcal{B}_k(u_{k-1})$ respectively.

To allow for employing a Gauß-Seidel like iterative scheme, we will see that $\mathcal{F}_k$ is the result of $m_k \leq m$ Linesearch iterations on $\mathcal{D}_k \neq \mathbb{R}^n$ and $m_G \geq 0$ Linesearch iterations on $\mathbb{R}^n$. Here, $m = \max_{k=0,\dots,N} m_k$, is the maximal number of Linesearch steps on each subset. Indeed, to derive a convergent scheme, we will see that we must either compute $m_G > 0$ global smoothing steps or have a domain decomposition with $m_k > 0$ local smoothing steps.

5.3.2 A Modified Armijo Condition

Similar to the additive Linesearch approach, we extend the Armijo-Condition (2.2.5) to the multiplicative preconditioning context. Therefore, on each subset $\mathcal{D}_k$, the step–length parameter must satisfy the following Armijo condition

$$H_k^\nu(u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i}) \leq H_k^\nu(u_{k,i}^\nu) + \rho_A \alpha_{k,i}^\nu \langle s_{k,i}, g_{k,i}^\nu \rangle \tag{5.3.1}$$

Moreover, on each subset $\mathcal{D}_k \neq \mathbb{R}^n$, we demand that beginning from the second subset iteration the following inequality holds

$$\langle u_{k,i}^\nu - u_{k,*}^\nu + \alpha_{k,i}^\nu s_{k,i}, g_{k,*}^\nu \rangle \leq \rho_R \langle u_{k,1}^\nu - u_{k,*}^\nu, g_{k,*}^\nu \rangle \tag{5.3.2}$$

where $0 < \rho_R \leq \rho_A < 1$. Here, $u_{k,*}^\nu$ denotes the first iterate on $\mathcal{D}_k$ in the sense that $u_{k,*}^\nu = u_{l,0}^\nu$ where the index l is the smallest integer with $C_l = C_k$ and $C_l \subsetneq C_{l-1}$ such that there does not exists an i with $l < i < k$, $C_i = C_k$ and $C_i \subsetneq C_{i-1}$.
In contrast to [WG08], this condition is stated directly and is not a result of a condition to the objective function, like, e.g.,

$$H_k^\nu(u_{k,i}^\nu + \alpha_{k,i}^\nu s_{k,i}) > H_k^\nu(u_{k,*}^\nu) + \rho_{MP_2} \langle g_{k,*}^\nu, u_{k,i}^\nu - u_{k,*}^\nu + \alpha_{k,i}^\nu s_{k,i} \rangle \tag{5.3.3}$$

where $1 - \rho_A \leq \rho_{MP_2} < 1$. This condition of Wolfe type can be illustrated as in Figure 5.1. In fact, the formulation of such a condition has the drawback, that local minimizers are not generally included in the region of feasible step–lengths (cf., [NW06] and Figure 5.1). Even worse, a step–length parameter satisfying (5.3.3) may be $\alpha > 1$ and, therefore, can in general not be computed

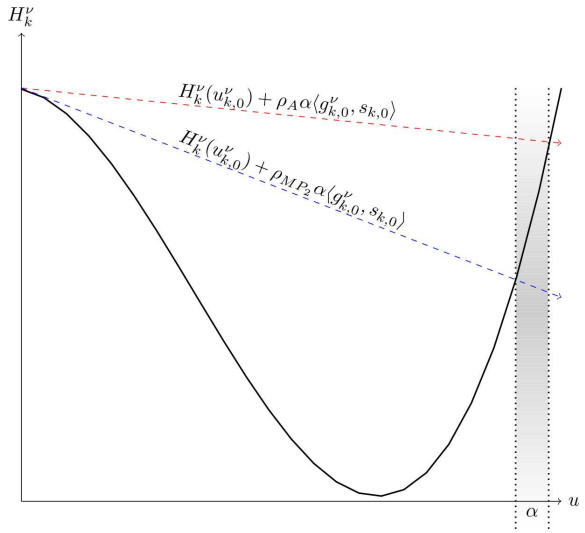

Figure 5.1: Illustration of Z. Wen's and D. Goldfarb's step–length condition (5.3.3) in the first computation step on domain $\mathcal{D}_k$. As one can see, the local minimizer is not included in the region of admissible step–lengths (between the dotted lines). Moreover, in general there must not necessarily exist step–lengths α which are smaller than one, in contrast to the step–lengths which satisfy the Armijo condition. Thus, cheap backtracking algorithms are generally not applicable. An illustration of the step–length criterion within the MPLS algorithm is given in Figure 4.1.

with the traditional backtracking scheme, when starting from $\alpha_0 < \alpha$. To this end, one must employ a (possibly expensive) bisection algorithm. As a sideeffect, this may yield that even for admissible search directions the resulting scaled search–direction may not be admissible in the sense of $\mathcal{B}_k(u_{k-1})$. Though, as it turns out, only if the subset objective function has a uniformly positive definite Hessian, (5.3.3) as a constraint vanishes, cf. Lemma 3.2 [WG08]. In contrast, the following lemma shows that (5.3.2) is always satisfied if the Hessians are just positive semi-definite.

Lemma 5.3.1. *Assume that ($A_{mpls}1$) and ($A_{mpls}2$) hold and suppose that H_k^ν is twice continuously differentiable. Moreover assume that the Hessians of H_k^ν are positive semidefinite, i.e.,*

$$0 \leq \langle s, \nabla^2 H_k^\nu(u) s \rangle \qquad \forall u \in \mathcal{D}_k, \qquad \forall s : u + s \in \mathcal{B}_k(u_{k-1})$$

Suppose furthermore that all search directions satisfy (2.2.2b) and (5.3.1). Then assumption (5.3.2) is satisfied for every step–length $\alpha_{k,i}^\nu \in (0, \alpha_0]$ satisfying (5.3.1).

Proof. The proof is the same as the one of Lemma 4.2.1 (Section 4.2.2). □

A Practicable Descent Condition

As we have seen, in the additive case, the descent criterion (5.3.2) can, for instance, be satisfied by employing the following descent criterion

$$\langle u_{k,i}^\nu - u_{k,*}^\nu + \alpha_{k,i}^\nu s_{k,i}, g_{k,*}^\nu \rangle \leq \rho_{MP} \langle u_{k,i}^\nu - u_{k,*}^\nu, g_{k,*}^\nu \rangle \tag{5.3.4}$$

where $\rho_R \leq \rho_{MP}^m$ and $m = \max_{k=0,\ldots,N} m_k$. Since, as we will see, such a criterion ensures that a backtracking algorithm is able to compute an admissible step–length, we will employ (5.3.1) and

Modified Linesearch Algorithm

Input: $u^{\nu}_{k,0} \in \mathcal{D}_k, \mathcal{B}_k, k \in \mathbb{N}, m, n_k \in \mathbb{N}$
Output: $u^{\nu}_{k,m_k} \in \mathcal{D}_k$

i = 0
do until $(i = m)$ {
  if $(\mathcal{D}_k \neq \mathbb{R}^n$ **and** $u^{\nu}_{k,i} \neq u^{\nu}_{k,*})$ {
    compute a search–direction $s_{k,i}$ satisfying (2.2.2a) and (5.3.5) and $u_{k,i} + s_{k,i} \in \mathcal{B}_k$
    call the Backtracking Algorithm 2 to compute a step–length $\alpha^{\nu}_{k,i}$ satisfying (5.3.1) and (5.3.4)
  } **else if** $(\mathcal{D}_k \neq \mathbb{R}^n)$ {
    compute a search–direction $s_{k,i}$ satisfying (2.2.2a) and (5.3.5) and $u_{k,i} + s_{k,i} \in \mathcal{B}_k$
    call the Backtracking Algorithm 2 to compute a step–length $\alpha^{\nu}_{k,i}$ satisfying (5.3.1)
  } **else** {
    compute a search–direction $s_{k,i}$ satisfying (2.2.2a) and (2.2.2b) and $u_{k,i} + s_{k,i} \in \mathcal{B}_k$
    call the Backtracking Algorithm 2 to compute a step–length $\alpha^{\nu}_{k,i}$ satisfying (2.2.5)
  }
  set $u^{\nu}_{k,i+1} = u^{\nu}_{k,i} + \alpha^{\nu}_{k,i} s^{\nu}_{k,i}$
  $i = i + 1$
}

return u^{ν}_{k,m_k}

Algorithm 10: Modified Linesearch Algorithm

(5.3.4) within our MPLS algorithm, Algorithm 11. However, even if we employ this criterion, the fact that each search–direction is a descent direction is still a direct consequence and not a result of a condition of Wolfe type.

Lemma 5.3.2. *Assume that ($A_{mpls}1$) and ($A_{mpls}2$) hold. Suppose that a given $u^{\nu}_{k,i} \in \mathcal{B}_k(u_{k-1})$ was computed using (5.3.1) and (5.3.4), as well as that each search–direction $s_{k,i} \in \mathcal{D}_k$ is a descent direction, according to (2.2.2a) and that $u^{\nu}_{k,i} + s_{k,i} \in \mathcal{B}_k(u_{k-1})$. Then there exists an $0 < \alpha^{\nu}_{k,i} \leq \alpha_0 \leq 1$ such that (5.3.1) holds and for $u^{\nu}_{k,i} \neq u^{\nu}_{k,*}$ such that, both, (5.3.1) and (5.3.4) hold.*

Proof. This proof is the same as the proof of Lemma 4.2.2. □

Finally, like in the APLS setting, we have to ensure that the lengths of the subset search directions are limited by the norm of the initial subset gradient. To this end, we formulate the following (recursive) criterion

$$\|u^{\nu}_{k,i} - u^{\nu}_{k,*} + s_{k,i}\|^2_{\infty} \leq \beta_{ls} \Delta_k \tag{5.3.5a}$$

$$\|s_{k,i}\|^2_{\infty} \leq \beta_{ls} \|\hat{g}^{\nu}_{k,i}\|^2_{\infty} \tag{5.3.5b}$$

for all $i = 0, \ldots, m_k - 1$ and $k = 1, \ldots, N$, if $\mathcal{D}_k \neq \mathbb{R}^n$. Here, Δ_k is given as

$$\Delta_k = \begin{cases} \|I_k^{k-1}\|_{\infty}^{-2} \min\{ \quad \|\hat{g}^{\nu}_{k-1,m_{k-1}}\|^2_{\infty}, & \\ \qquad\qquad \Delta_{k-1} - \beta_{ls}^{-1} \|u^{\nu}_{k-1,m_{k-1}} - u^{\nu}_{k-1,*}\|^2_{\infty} \} & \text{if } \mathcal{D}_k \neq \mathbb{R}^n \\ \infty & \text{if } \mathcal{D}_k = \mathbb{R}^n \end{cases} \tag{5.3.6}$$

Here, $\beta_{ls} > 0$ is the constant from (2.2.2b). As it will turn out, this condition substitutes (4.2.10a) within the modified Linesearch algorithm, Algorithm 10.

Algorithm: MPLS – Nonlinear Multiplicatively Preconditioned Linesearch Strategy

Input: $k \in \{0, \ldots, N\}$, Δ_k, $\mathcal{B}_k$, $u^\nu_{k,0} \in \mathcal{D}_k$, g^ν_k
Output: final iterate $u^\nu_{k,m_{k,f}} \in \mathcal{D}_k$

Smoothing
call the Linesearch algorithm, Algorithm 10, with $u^\nu_{k,0}$, $\mathcal{B}_k$, $\underbrace{m}_{=m_k}$, $\underbrace{\dim \mathcal{D}_k}_{=n}$
and receive u^ν_{k,m_k}.

if $(C_{k+1} \supsetneq C_k)$
return $u^\nu_{k,m_{k,f}} = u^\nu_{k,m_k}$

else if $(C_{k+1} = C_k)$ {
$u^\nu_{k+1,0} = u^\nu_{k,m_k}$
$k = k + 1$
goto *Smoothing*

} **else if** $(C_{k+1} \subsetneq C_k$ **and** (5.1.1) holds) {
compute Δ_{k+1} by means of (5.3.6)
call *MPLS* with $k+1$, Δ_{k+1}, $\underbrace{\mathcal{B}_{k+1}(u^\nu_{k,m_k})}_{=\mathcal{B}_{k+1} \text{ cf. (3.3.2)}}$, $\underbrace{P^{k+1}_k u^\nu_{k,m_k}}_{=u^\nu_{k+1,0}}$, $\underbrace{R^{k+1}_k g^\nu_{k,m_k}}_{g^\nu_k}$ **and** receive $u^\nu_{k+1,m_{k+1,f}}$
set $s_{k,m_k} = I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - P^{k+1}_k u^\nu_{k,m_k})$

if ($\mathcal{D}_k \neq \mathbb{R}^n$ and $u^\nu_{k,m_k} \neq u^\nu_{k,*}$) {
call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,m_k} satisfying (5.3.1) and (5.3.4)
} **else** {
call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,m_k} satisfying (5.3.1)

set $u^\nu_{r_k,0} = u^\nu_{k,m_k} + \alpha^\nu_{k,m_k} s^\nu_{k,m_k}$

if $(k \neq 0)$
$k = r_k$
else
$u^{\nu+1}_{0,0} = u^\nu_{r_k,m_k+1}$, $\nu = \nu + 1$
}
}

Algorithm 11: MPLS – Nonlinear Multiplicatively Preconditioned Linesearch Strategy

Numerically both inequalities can be fulfilled by substituting (5.3.5) by

$$\|s^\nu_{k,i}\|^2_\infty \leq \min\{\beta_{ls}\Delta_k - \|u^\nu_{k,i} - u^\nu_{k,*}\|^2_\infty, \beta_{ls}\|\hat{g}^\nu_{k,i}\|^2_\infty\}$$

On the other hand, the construction of the Linesearch algorithm, Algorithm 10, along with the definition of $\mathcal{B}_{k+1}(u^\nu_{k,m_k})$ in (3.3.2) satisfies the assumptions in Lemma 3.3.1 and shows that multiplicatively computed corrections are admissible in $\mathcal{B}_k(u_{k-1})$.

The Nonlinear Update Operator

The nonlinear operator $\mathcal{F}_M$ is the result of N possible recursions. In the particular framework of this chapter, we define a recursively computed correction by

$$s_{k,m_k} = I^k_{k+1} s^\nu_{k+1} = I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0})$$

where m_k is the iteration when a recursion is called and $u^\nu_{k+1,m_{k+1,f}}$ is the final iterate on $\mathcal{D}_{k+1}$. Note that we changed our point of view. Now the first iterate on the next subset is well-known, i.e., $u^\nu_{k+1,0} = u^\nu_{k+1,*}$ but the last one is $u^\nu_{k+1,m_{k+1,f}} = u^\nu_{l,m_l}$ where $l = r_k - 1$.
Since we consider a Linesearch framework, the actual implementation of the multiplicative recombination operator is

$$\mathcal{A}^\nu_{\text{MPLS},k}(s_{k,m_k}, u^\nu_{k,m_k}) = u^\nu_{k,m_k} + \alpha^\nu_{k,m_k} s_{k,m_k}$$

where α^ν_{k,m_k} is chosen such that the Armijo condition (5.3.1) holds. Moreover, if $\mathcal{D}_k \neq \mathbb{R}^n$ and the initial iteration on $\mathcal{D}_k$ already took place, i.e., $u^\nu_{k,i} \neq u^\nu_{k,*}$, we demand that also (5.3.4) is satisfied. Thus, the nonlinear multiplicative update operator is given by

$$\mathcal{F}^{(j)}_M(u^\nu_{j,m_j}) = \mathcal{A}^\nu_{\text{MPLS},j}(s_{j,m_j}, u^\nu_{j,m_j}) \tag{5.3.7}$$

for all j with $\mathcal{D}_j = \mathbb{R}^n$.

5.3.3 Convergence to First–Order Critical Points

Like in Section 2.2, we will prove the convergence of the just presented MPLS Algorithm, Algorithm 11 by showing that each multiplicatively computed correction is a descent direction, each correction satisfies a sufficient decrease condition and that the step–length parameters are bounded from below.

Lemma 5.3.3. *Suppose that $s_{k,i}$ is a correction which was computed in Algorithm 11. Then we obtain that the following estimation holds*

$$\|s_{k,i}\|^2_\infty \leq \beta_{ls} \|\hat{g}^\nu_{k,i}\|^2_\infty \tag{5.3.8}$$

for all $k = 1, \ldots, N$ and all $i = 0, \ldots, m$.

Proof. By (5.3.5b) inequality (5.3.8) holds for every correction computed within the Linesearch algorithm.
Now we consider the case $s_{k,i} = s_{k,m_k}$, i.e., a recursively computed correction, and inductively prove that (5.3.8) holds. First, assume that $s_{k,m_k} = I^k_{k+1} s^\nu_{k+1}$ where $\mathcal{D}_{k+1}$ is the lowermost subset. Due to (5.3.5) we obtain

$$\begin{aligned} \|s_{k,m_k}\|^2_\infty &= \|I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0})\|^2_\infty \\ &\leq \|I^k_{k+1}\|^2_\infty \|u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0}\|^2_\infty \leq \|I^k_{k+1}\|^2_\infty \beta_{ls} \Delta_k \\ &\leq \beta_{ls} \min\{\|\hat{g}^\nu_{k,m_k}\|^2_\infty, \Delta_{k-1} - \beta^{-1}_{ls} \|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty\} \end{aligned} \tag{5.3.9}$$

Obviously, (5.3.8) holds.
Now, if $k+1$ is not the lowermost subset, we obtain due to the previous inequality, $\alpha^\nu_{k,m_k} \leq \alpha_0 \leq 1$ and the definition of Δ_{k+1} that

$$\begin{aligned} \|u^\nu_{k,m_k} + \alpha^\nu_{k,m_k} s_{k,m_k} - u^\nu_{k,*}\|^2_\infty &\leq \|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty + \|\alpha^\nu_{k,m_k} s_{k,m_k}\|^2_\infty \\ &\leq \|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty + \alpha_0 \|s_{k,m_k}\|^2_\infty \\ &\leq \|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty + \beta_{ls}\Delta_k - \|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty \\ &= \beta_{ls}\Delta_k \end{aligned}$$

Following the argumentation in (5.3.9) gives rise to $\|u^\nu_{k,m_k} - u^\nu_{k,*}\|^2_\infty \leq \beta_{ls}\Delta_{k-1}$ and proves the proposition. □

Next, we show that the Linesearch parameters which satisfy the Armijo condition are bounded from below by a constant depending on the gradient and the search direction. As we will see, we will prove this result only for initial subset corrections on $\mathcal{D}_k \neq \mathbb{R}^n$. Here, we are just interested in the initial corrections, since condition (5.3.4) enables us to relate subsequent corrections to the initial one.

Lemma 5.3.4. *Suppose that ($A_{mpls}1$) and ($A_{mpls}2$) hold. Then for each Linesearch step on $\mathcal{D}_j = \mathbb{R}^n$ and each initial Linesearch step on $\mathcal{D}_k \neq \mathbb{R}^n$ with $u^\nu_{k,0} = u^\nu_{k,*}$ in the MPLS algorithm the Armijo condition (5.3.1) is satisfied for*

$$\hat{\alpha}^\nu_{k,i} = \frac{2(\rho_A - 1)\langle s_{k,i}, g^\nu_{k,i}\rangle}{L_g \|s_{k,i}\|_2^2}$$

where L_g is the Lipschitz constant for ∇H^ν_k. Moreover, we obtain for the step–length $\alpha^\nu_{k,i}$ of each smoothing step on $\mathcal{D}_k$ the estimation

$$\min\{\alpha_0, 2\tau\hat{\alpha}^\nu_{k,i}\} \leq \alpha^\nu_{k,i} \leq \min\{\alpha_0, 2\hat{\alpha}^\nu_{k,i}\}$$

Proof. Due to the assumption of this lemma, the proof is the same as the one for Lemma 2.2.2. □

Lemma 5.3.5. *Assume that ($A_{mpls}1$) and ($A_{mpls}2$) hold. Then there exists a constant such that for each correction $s_{k,i}$ with $\mathcal{D}_k = \mathbb{R}^n$, computed in the MPLS algorithm, Algorithm 11, the inequality holds*

$$-\langle g^\nu_{k,i}, s_{k,i}\rangle \geq c_{MPLS}\|\hat{g}^\nu_{k,i}\|_2^2$$

where $c_{MPLS} = c_{MPLS}(\beta_{ls}, \eta_{ls}, \alpha_0, \rho_{MP}, m, N) > 0$.

Proof. We prove the proposition inductively by defining c_{MPLS}. By assumption (2.2.2a), we obtain that each Linesearch correction on $\mathcal{D}_k = \mathbb{R}^n$ and each initial Linesearch correction satisfies

$$-\langle g^\nu_{k,i}, s_{k,i}\rangle \geq c^{(k)}_{MPLS}\|\hat{g}^\nu_{k,i}\|_2^2 = \eta_{ls}\|\hat{g}^\nu_{k,i}\|_2^2$$

In this case we define $c^{(k)}_{MPLS} = \eta_{ls} > 0$ which gives us the induction statement.
Now, we consider a recursively computed correction. Due to the definition of the subset objective function we have $g^\nu_{k+1,0} = \nabla H^\nu_{k+1}(u^\nu_{k+1,0}) = R^{k+1}_k g^\nu_{k,m_k}$. Now we use the definition of R^{k+1}_k and s_{k,m_k} and obtain

$$\begin{aligned} -\langle g^\nu_{k,m_k}, s_{k,m_k}\rangle &= -\langle g^\nu_{k,m_k}, I^k_{k+1}\left(u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0}\right)\rangle \\ &\geq -\langle R^{k+1}_k g^\nu_{k,m_k}, u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0}\rangle \\ &= -\langle g^\nu_{k+1,0}, u^\nu_{k+1,m_{k+1,f}} - u^\nu_{k+1,0}\rangle \end{aligned}$$

Since each correction on $\mathcal{D}_{k+1}$ is a descent direction which satisfies the descent condition (5.3.4) we obtain

$$-\langle g^\nu_{k,m_k}, s_{k,m_k}\rangle \geq -\rho^m_{MP}\langle g^\nu_{k+1,0}, \alpha^\nu_{k+1,0}s_{k+1,0}\rangle$$

Now we may employ the induction statement and obtain

$$-\langle g^\nu_{k,m_k}, s_{k,m_k}\rangle \geq \rho^m_{MP}\alpha^\nu_{k+1,0}c^{(k+1)}_{MPLS}\|\hat{g}^\nu_{k+1,0}\|_2^2 \tag{5.3.10}$$

Next, we will derive a lower bound for $\alpha^{\nu}_{k+1,0}$. To this end, we apply Lemma 5.3.4, i.e., the step–length estimation by means of gradient and the correction, and obtain

$$\alpha^{\nu}_{k+1,0} \geq \frac{2\tau(\rho_A - 1)\langle s_{k+1,0}, g^{\nu}_{k+1,0}\rangle}{L_g \|s_{k+1,0}\|_2^2}$$

Now we estimate the correction's size by employing Lemma 5.3.3 and $n \geq n_{k+1} = \dim \mathcal{D}_{k+1}$ which yields

$$\|s_{k+1,0}\|_2^2 \leq n\|s_{k+1,0}\|_\infty^2 \leq n\beta_{ls}\|\hat{g}^{\nu}_{k+1,0}\|_\infty^2 \leq n\beta_{ls}\|\hat{g}^{\nu}_{k+1,0}\|_2^2$$

Employing the induction statement, and the previous inequality yields

$$\alpha^{\nu}_{k+1,0} \geq \min\{\alpha_0, \frac{2\tau(1-\rho_A)c^{(k+1)}_{MPLS}\|\hat{g}^{\nu}_{k+1,0}\|_2^2}{L_g n\beta_{ls}\|\hat{g}^{\nu}_{k+1,0}\|_2^2}\} = \min\{\alpha_0, \frac{2\tau(1-\rho_A)c^{(k+1)}_{MPLS}}{L_g n\beta_{ls}}\}$$

Now, we combine the estimation for the step–length along with our subset gradient condition (5.1.1) and (5.3.10) which yields

$$\begin{aligned} -\langle g^{\nu}_{k,m_k}, s_{k,m_k}\rangle &\geq \rho^m_{MP}\min\{\alpha_0, \frac{2\tau(1-\rho_A)c^{(k+1)}_{MPLS}}{L_g n\beta_{ls}}\}c^{(k+1)}_{MPLS}\|\hat{g}^{\nu}_{k+1,0}\|_2^2 \\ &\geq \rho^m_{MP}\kappa_g^2\min\{\alpha_0, \frac{2\tau(1-\rho_A)c^{(k+1)}_{MPLS}}{L_g n\beta_{ls}}\}c^{(k+1)}_{MPLS}\|\hat{g}^{\nu}_{k,m_k}\|_2^2 \end{aligned}$$

Next, we can define the sought-after constant recursively by employing $c^{(k+1)}_{MPLS}$ and the traditional Linesearch constants as follows

$$c^{(k)}_{MPLS} = \rho^m_{MP}\kappa_g^2\min\{\alpha_0, \frac{2\tau(1-\rho_A)c^{(k+1)}_{MPLS}}{L_g n\beta_{ls}}\}c^{(k+1)}_{MPLS} > 0 \tag{5.3.11}$$

Together this provides

$$-\langle g^{\nu}_{k,m_k}, s_{k,m_k}\rangle \geq c^{(k)}_{MPLS}\|\hat{g}^{\nu}_{k,m_k}\|_2^2$$

Since the number of recursions is limited by N and each constant $c^{(k)}_{MPLS} > 0$ we can choose

$$c_{MPLS} = \min_{k=0,\ldots,N}\{c^{(k)}_{MPLS}\} \tag{5.3.12}$$

This proves the proposition. □

The following lemma considers the estimation of c_{MPLS}.

Lemma 5.3.6. *Assume that all constants in $c^{(k)}_{MPLS}$ are given as defined in Algorithm 11. Then we obtain*

$$c^{(k)}_{MPLS} \geq (\tau\Theta)^{2^{2(N-k)}}\min\{1, \eta_{ls}^{2^{(N-k+1)}}\} \tag{5.3.13}$$

where

$$\Theta = \min\{\alpha_0, \frac{2(1-\rho_A)}{L_g n}\}\rho^m_{MP}\kappa_g^2$$

with $1 > \Theta > 0$.

Proof. If k is the deepest level in the recursion, we obtain $c_{MPLS}^{(k)} = \eta_{ls} \geq \min\{1, \eta_{ls}^2\}$ which proves the statement.
Now we consider $k < N$ and employ the definition of $c_{MPLS}^{(k)}$, (5.3.13) and that

$$1 \geq c_{MPLS}^{(k+1)} = (\tau\Theta)^{2^{(N-k-1)}} \min\{1, \eta_{ls}^{2^{(N-k)}}\} > 0$$

which yields

$$\begin{aligned} c_{MPLS}^{(k)} &= \rho_{MP}^m \kappa_g^2 \min\{\alpha_0, \frac{2\tau(1-\rho_A)c_{MPLS}^{(k+1)}}{L_g n}\} c_{MPLS}^{(k+1)} \\ &\geq \rho_{MP}^m \kappa_g^2 \tau \min\{\alpha_0, \frac{2(1-\rho_A)}{L_g n}\} \left((\tau\Theta)^{2^{2(N-k-1)}} \min\{1, \eta_{ls}^{2^{(N-k)}}\}\right)^2 \\ &\geq (\tau\Theta)^2 (\tau\Theta)^{2^{2(N-k-1)+1}} \min\{1, \eta_{ls}^{2^{(N-k+1)}}\} \\ &= (\tau\Theta)^{2^{2(N-k)}} \min\{1, \eta_{ls}^{2^{(N-k+1)}}\} \end{aligned}$$

This proves the proposition. □

As we have seen in the proof of Lemma 5.3.5, one may also estimate each step–length parameter independent from the gradients and corrections.

Lemma 5.3.7. *Assume that ($A_{mpls}1$) and ($A_{mpls}2$) hold. Then there exists a constant such that each step–length parameter $\alpha_{k,i}^{\nu}$ for recursively computed corrections in the MPLS algorithm is bounded from below, i.e.,*

$$\alpha_{k,i}^{\nu} \geq \min\left\{\alpha_0, \frac{2\tau(1-\rho_A)c_{MPLS}}{L_g n \beta_{ls}}\right\}$$

where $\mathcal{D}_k = \mathbb{R}^n$ and c_{MPLS} is as defined in (5.3.12).

Proof. First, we exploit Lemma 5.3.4 which gives

$$\alpha_{k,i}^{\nu} \geq \min\left\{\alpha_0, \frac{2\tau(\rho_A - 1)\langle s_{k,i}, g_{k,i}^{\nu}\rangle}{L_g \|s_{k,i}\|_2^2}\right\}$$

Now, we employ Lemma 5.3.3 and Lemma 5.3.5 and obtain

$$\alpha_{k,i}^{\nu} \geq \min\left\{\alpha_0, c_{MPLS}\frac{2\tau(1-\rho_A)\|\hat{g}_{k,i}^{\nu}\|_2^2}{nL_g\beta_{ls}\|\hat{g}_{k,i}^{\nu}\|_2^2}\right\} \geq \min\left\{\alpha_0, c_{MPLS}\frac{2\tau(1-\rho_A)}{L_g n\beta_{ls}}\right\}$$

□

In a similar fashion like Theorem 2.2.4, we will now prove the convergence of the nonlinear multiplicatively preconditioned Linesearch algorithm, Algorithm 11.

Theorem 5.3.8. *Suppose that ($A_{mpls}1$) and ($A_{mpls}2$) hold. Furthermore, assume that either on one subset $\mathcal{D}_j = \mathbb{R}^n$ at least $m_j = m_G > 0$ Linesearch steps are computed or an overlapping or non-overlapping domain decomposition is employed. If a domain decomposition is employed, we assume furthermore that $m_k > 0$ holds for each $\mathcal{D}_k \neq \mathbb{R}^n$ and that the constants are given as in (4.1.7) and (4.1.8), respectively.*

Then the MPLS algorithm, Algorithm 11, computes a sequence of iterates converging to a first–order critical point for problem (M), i.e.,

$$\lim_{\nu\to\infty,\mathcal{D}_j=\mathbb{R}^n,i\in\{0,\dots,m\}} \|\hat{g}^\nu_{j,i}\|_2 = 0 \tag{5.3.14}$$

Proof. As in the proof of Theorem 2.2.4, we use that on $\mathcal{D}_j = \mathbb{R}^n$ the Armijo condition (5.3.1) gives rise to

$$J(u^\nu_{j,i}) - J(u^\nu_{j,i+1}) \geq -\alpha^\nu_{j,i}\rho_A\langle s_{j,i}, g^\nu_{j,i}\rangle$$

We employ Lemma 5.3.5 and (2.2.2b), respectively, and obtain

$$J(u^\nu_{j,i}) - J(u^\nu_{j,i+1}) \geq \begin{cases} \alpha^\nu_{j,i} c_{MPLS}\|\hat{g}^\nu_{j,i}\|_2^2 & \text{if } s_{j,i} \text{ was computed recursively} \\ \alpha^\nu_{j,i}\eta_{ls}\|\hat{g}^\nu_{j,i}\|_2^2 & \text{otherwise} \end{cases}$$

Now we employ Lemma 2.2.3 and Lemma 5.3.7, respectively which gives

$$\alpha^\nu_{j,i} \geq \begin{cases} \min\left\{\alpha_0, \frac{2\tau(1-\rho_A)c_{MPLS}}{L_g\beta_{ls}n}\right\} & \text{if } s_{j,i} \text{ was computed recursively} \\ \min\left\{\alpha_0, \frac{2\tau(1-\rho_A)}{L_g n\beta_{ls}}\right\} & \text{otherwise} \end{cases}$$

Note that if on $\mathcal{D}_j = \mathbb{R}^n$ no Linesearch steps are computed, we have a domain decomposition as introduced in Section 3.1.6. In this case, Lemma 4.1.5 is valid and in each iteration ν at least on one subset a correction is computed. Together with the compactness of $\mathcal{L}^0$ and, thus, $J(u^\nu_{j,i}) - J(u^\nu_{j,i+1}) \to 0$ we can conclude that $\|\hat{g}^\nu_{j,i}\|_2 \to 0$ which proves the proposition. □

5.4 Combined Nonlinearly Preconditioned Linesearch Methods

To improve the rates of convergence of the traditional Linesearch method from Section 2.2, we introduced the preconditioned Linesearch variants Algorithm 7 and Algorithm 11. In this section, we will introduce an algorithm which combines both preconditioning strategies by substituting the Linesearch solver within MPLS by the APLS solver. Hence, we obtain the AMPLS algorithm as presented in Algorithm 12.

Convergence to First–Order Critical Points

Similar to the traditional Linesearch analysis, we will once more prove that each correction computed within the AMPLS algorithm satisfies a sufficient decrease condition. In turn, we may use the standard argumentation and obtain the sought–after convergence result.

Lemma 5.4.1. *Assume that $\mathbb{R}^n$ is decomposed into a sequence of nested subspaces, as introduced in Section 3.1.4, and that the respective subsets $\mathcal{D}_k$ are decomposed as in (5.2.1). Moreover, assume that ($A_{mpls}1$), ($A_{mpls}2$) hold on each "multiplicative" subset $\mathcal{D}_k$ and that ($A_{apls}1$) and ($A_{apls}2$) hold on the respective subspaces $\mathcal{D}_{k,l}$. Then for each correction $s_{k,m_k} \in \mathcal{D}_k = \mathbb{R}^n$, computed multiplicatively within the AMPLS algorithm, we obtain*

$$-\langle g^\nu_{k,m_k}, s_{k,m_k}\rangle \geq \alpha^\nu_{k,m_k} c_{MPLS2}\|\hat{g}^\nu_{k,m_k}\|_2^2$$

where $c_{MPLS2} > 0$.

Algorithm: AMPLS – Nonlinear Additively and Multiplicatively Preconditioned Linesearch Strategy

Input: $k \in \{0, \ldots, N\}, \Delta_k \mathcal{B}_k, u^\nu_{k,0}, g^\nu_{k-1} \in \mathcal{D}_{k-1}, \Delta_{k-1} \in \mathbb{R}^+ \cup \{\infty\}$
Output: new iterate $u^\nu_{k,m_{k,f}} \in \mathcal{D}_k$

Smoothing
call a modified *APLS* algorithm with $\underbrace{u^\nu_{k,0}}_{=u_0}$, $\underbrace{H^\nu_k}_{=J}$, $\underbrace{\dim \mathcal{D}_k}_{=n}$
which ensures that (5.3.1), (5.3.4) and (4.2.4) is satisfied
and receive a new iterate u^ν_{k,m_k} and a new Trust-Region radius Δ^ν_{k,m_k}

if $(C_{k+1} \supsetneq C_k)$
return $u^\nu_{k,m_{k,f}} = u^\nu_{k,m_k}$

else if $(C_{k+1} = C_k)$ {
$u^\nu_{k+1,0} = u^\nu_{k,m_k}$
$k = k + 1$
goto *Smoothing*

} **else if** $(C_{k+1} \subsetneq C_k$ **and** (5.1.1) holds) {
computed Δ_{k+1} by means of (5.3.6)
call *MPLS* with $k+1$, Δ_{k+1}, $\underbrace{\mathcal{B}_{k+1}(u^\nu_{k,m_k})}_{=\mathcal{B}_{k+1} \text{ cf. (3.3.2)}}$, $\underbrace{P^{k+1}_k u^\nu_{k,m_k}}_{=u^\nu_{k+1,0}}$, $\underbrace{R^{k+1}_k g^\nu_{k,m_k}}_{g^\nu_k}$ **and** receive $u^\nu_{k+1,m_{k+1,f}}$
set $s_{k,m_k} = I^k_{k+1}(u^\nu_{k+1,m_{k+1,f}} - P^{k+1}_k u^\nu_{k,m_k})$

if $(\mathcal{D}_k \neq \mathbb{R}^n$ and $m^\nu_{k,m_k} \neq u^\nu_{k,*})$
call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,m_k} satisfying (5.3.1) and (5.3.4)
else
call the Backtracking Algorithm 2 to compute a step–length α^ν_{k,m_k} satisfying (5.3.1)

set $u^\nu_{r_k,0} = u^\nu_{k,m_k} + \alpha^\nu_{k,m_k} s^\nu_{k,m_k}$

if $(k \neq 0)$
$k = r_k$
else
$u^{\nu+1}_{0,0} = u^\nu_{r_k,m_k+1}$, $\nu = \nu + 1$
}

Algorithm 12: AMPLS – Nonlinear Additively and Multiplicatively Preconditioned Linesearch Strategy

Proof. Similar to the proof of Lemma 5.3.5, we prove the proposition by induction. But since just the induction statement changes, we can employ major parts of the proof of Lemma 5.3.5.

By assumption (2.2.2a) and the result of Lemma 4.2.4, we obtain that each initial Linesearch and APLS correction satisfies

$$-\langle g_{k,i}^{\nu}, s_{k,i}\rangle \geq \min\left\{\eta_{ls}, \kappa_g^2 \rho_{AP}^m \eta_{ls} \min\left\{\alpha_0, 2\tau\alpha_{\min}\right\}\right\} \|\hat{g}_{k,i}^{\nu}\|_2^2$$

In this case, we define $c_{MPLS2}^{(k)} = \min\{\eta_{ls}, \kappa_g^2 \rho_{AP}^m \eta_{ls} \min\{\alpha_0, 2\tau\alpha_{\min}\}\} > 0$.

Now, we can exploit exactly the same argumentation as in the proof of Lemma 5.3.5 and obtain that for recursively computed corrections the proposition holds with

$$c_{MPLS2}^{(k)} = \rho_{MP}^m \kappa_g^2 \min\{\alpha_0, \frac{2\tau(1-\rho_A)c_{MPLS2}^{(k+1)}}{L_g n \beta_{ls}}\} c_{MPLS2}^{(k+1)} > 0$$

Finally we choose

$$c_{MPLS2} = \min_{k=0,\ldots,N} c_{MPLS2}^{(k)}$$

which concludes the proof. □

Theorem 5.4.2. *Assume that $\mathbb{R}^n$ is decomposed into a sequence of nested subspaces, as introduced in Section 3.1.4 and that the respective subsets $\mathcal{D}_k$ are decomposed as in (5.2.1). Moreover, assume that ($A_{mpls}1$), ($A_{mpls}2$) hold on each "multiplicative" subset $\mathcal{D}_k$ and that ($A_{apls}1$) and ($A_{apls}2$) hold for the respective subspaces $\mathcal{D}_{k,l}$. Moreover suppose that either each subset $\mathcal{D}_j = \mathbb{R}^n$ is decomposed employing an overlapping or non-overlapping or that on one $\mathcal{D}_j = \mathbb{R}^n$ at least $m_j = m_G > 0$ global smoothing steps are computed. If a domain decomposition is employed, we assume furthermore that $m_{j,l} > 0$ holds for the subsets $\mathcal{D}_{j,l}$ of $\mathcal{D}_j = \mathbb{R}^n$ and that the constants are given as in (4.1.7) and (4.1.8), respectively.*

Then the sequence of iterates $(u_{j,i}^{\nu})_{i,\mathcal{D}_j=\mathbb{R}^n,\nu}$ computed in Algorithm 9, satisfies

$$\lim_{\nu\to\infty,\mathcal{D}_j=\mathbb{R}^n,i\in\{0,\ldots,m+1\}} \|\hat{g}_{j,i}^{\nu}\|_2 = 0 \tag{5.4.1}$$

Proof. As in the proof of Theorem 2.2.4, we use that each global Linesearch parameter $\alpha_{j,i}^{\nu}$ satisfies the Armijo condition (5.3.1), i.e.,

$$J(u_{j,i}^{\nu}) - J^{\nu}(u_{j,i+1}^{\nu}) \geq -\alpha_{j,i}^{\nu}\rho_A\langle s_{j,i}, g_{j,i}^{\nu}\rangle$$

Using, Lemma 4.2.4, Lemma 4.2.5 and equation (2.2.2b), respectively, gives,

$$J(u_{j,i}^{\nu}) - J(u_{j,i+1}^{\nu}) \geq \begin{cases} \alpha_{j,i}^{\nu}\kappa_g^2\rho_{AP}^m\eta_{ls}\min\left\{\alpha_0, 2\tau\alpha_{\min}\right\}\|\hat{g}_{j,i}^{\nu}\|_2^2 & \text{if } s_{j,i} \text{ was computed additively} \\ \alpha_{j,i}^{\nu}c_{MPLS2}\|\hat{g}_{j,i}^{\nu}\|_2^2 & \text{if } s_{j,i} \text{ was computed recursively} \\ \alpha_{j,i}^{\nu}\eta_{ls}\|\hat{g}_{j,i}^{\nu}\|_2^2 & \text{otherwise} \end{cases}$$

where $\alpha_{\min} > 0$ was defined in Lemma 4.2.3. Now we employ Lemma 2.2.3, Lemma 4.2.5 and

Lemma 5.3.7, and obtain for each Linesearch parameter

$$\alpha_{j,i}^{\nu} \geq \begin{cases} \min\{\alpha_0, 2\tau c_{\text{APLS}}\alpha_{\min}\} & \text{if } s_{j,i} \text{ was computed additively} \\ \min\left\{\alpha_0, \frac{2\tau(1-\rho_A)c_{MPLS2}}{L_g n \beta_{ls}}\right\} & \text{if } s_{j,i} \text{ was computed recursively} \\ \min\left\{\alpha_0, \frac{2\tau(1-\rho_A)}{L_g n \beta_{ls}}\right\} & \text{otherwise} \end{cases}$$

Note that if, within the APLS algorithm, $m_G = 0$, we have a domain decomposition as introduced in Section 3.1.6. In this case, Lemma 4.1.5 applies and in each iteration at least one subset correction is computed.
Together with the compactness of $\mathcal{L}^0$ and, thus, $J(u_{j,i}^{\nu}) - J(u_{j,i+1}^{\nu}) \to 0$ we conclude that

$$\|\hat{g}_{j,i}^{\nu}\|_2 \to 0$$

which proves the proposition. □

5.5 A Remark on Second-Order Convergence

In order to compute a second–order critical point, the iterative scheme must be able to "detect" and handle negative eigenvalues of the Hessians. As a matter of fact, even if the gradient is zero, one might have just found a saddle point. If a saddle point was computed, one must choose the right search direction, to succeed in computing a local minimizer, as pointed out in Section 2.1.5 and Section 2.2.5.
The presented preconditioning strategies can be considered as subspace correction methods, which may only resolve the eigenvectors and eigenvalues of the Hessian on the respective subspaces. Therefore, employing a multiplicative scheme to compute a search–direction which satisfies

$$\psi_{j,m_j}^{\nu}(s_{j,m_j}) \leq c\psi_{j,m_j}^{\nu}(s_{\min}) \tag{5.5.1}$$

with $c > 0$ and

$$\psi_{j,m_j}^{\nu}(s) = \langle g_{j,m_j}^{\nu}, s\rangle + \frac{1}{2}\langle s, \nabla^2 J(u_{j,m_j}^{\nu})s\rangle$$

is generally impossible. Here, $s_{\min}$ is the solution of

$$s : \psi_{j,m_j}^{\nu}(s) = \min! \text{ w.r.t. } \|s\|_\infty \leq \Delta_{j,m_j}^{\nu} \text{ and } u_{j,m_j}^{\nu} + s_{j,m_j} \in \mathcal{B}$$

We will briefly show that for some (realistic) examples multiplicative strategies are not able to satisfy (5.5.1). Suppose that $\mathbb{R}^n$ is decomposed non-overlappingly and suppose that the local objective function is a quadratic function like

$$H_k^{\nu}(u_{k,0}^{\nu} + s_k) = \psi_k^{\nu}(s_k) = \langle R_k g_{j,m_j}^{\nu}, s_k\rangle + \frac{1}{2}\langle s_k, R_k \nabla^2 J(u_{j,m_j}^{\nu}) I_k s_k\rangle$$

Similar to the argumentation in Section 4.4, we consider the following problem. Suppose that $g_k = 0$ and that $R_k\nabla^2 J(u_{j,m_j}^{\nu})I_k$ is positive definite. Furthermore, let $R_k\nabla^2 J(u_{j,m_j}^{\nu})I_i$ for $k \neq i$ be chosen such that $\nabla^2 J(u_{j,m_j}^{\nu})$ is negative definite. In this case, the local correction s_k is zero, but the solution of (5.5.1) may be the following vector

$$s_{\min} = \alpha x_{\lambda_{\min}}$$

where $x_{\lambda_{\min}}$ with $\|x_{\lambda_{\min}}\|_2 = 1$ is the eigenvector related to the smallest (negative) eigenvalue of $\nabla^2 J(u^\nu_{j,m_j})$. In a Trust-Region setting, the scaling parameter $\alpha > 0$ is chosen such that $u^\nu_{j,m_j} + s_{\min} \in \mathcal{B}$ and $\|s_{\min}\|_\infty \leq \Delta$ holds. In this case, we obtain that

$$\psi^\nu_{j,m_j}(I_k s_k) = 0$$

but

$$\psi^\nu_{j,m_j}(s_{\min}) = \lambda_{\min}\alpha < 0$$

Which shows, that also multiplicative corrections generally cannot solve (5.5.1).
Similarly, in the context of multiplicatively preconditioned Linesearch methods, the search directions must solve (5.5.1) in order to compute a second–order critical point [CL94]. But, the same reasoning shows that this is generally impossible.
Therefore, also multiplicative schemes aim at just improving the convergence of the globalization strategy. On the other hand, only the global smoothing strategy is able to ensure (quadratic) convergence to second–order critical points.
Complex real life simulations in solid mechanics are challenging in two ways. To obtain results which are close to reality, the geometry of the solid, in particular its boundaries must be resolved sufficiently accurate by the computational domain. But, the better the polyhedral mesh approximates the real geometry, the larger becomes the minimization problem. In addition, realistic physical models generally give rise to nonlinear, and in the case of contact, possibly nonsmooth objective functions.
From the engineer's point of view, real life simulations must be computed employing efficient and reliable strategies. As we have seen in Chapter 2, reliable solution strategies for large scale variants of our model problem (M) are the traditional Trust-Region and Linesearch strategies. However, efficiency for both globalization strategies may only be achieved, if the search directions can be computed in parallel. But, if large scale minimization problems with strong nonlinearities have to be solved, these traditional globalization strategies tend to converge slowly. On the other hand, the presented preconditioned globalization strategies truly converge to critical points. Now, in this chapter we will consider the convergence behavior of the presented traditional and nonlinearly preconditioned globalization strategies. To this end, we compare in several examples the rates of convergence of the respective Trust-Region strategies and Linesearch strategies with each other and comment on the convergence rates and computation times.
The presented examples in this chapter arise from the discretization of the PDEs as introduced in Section 1.3. Numerically, the discretization is carried out within the OBSLIB++ framework [Kra07b]. On the other hand, the presented solution strategies are implemented in the NLSOLVERLIB which extends OBSLIB++. A brief outline of technical aspects of the NLSOLVERLIB is given in Chapter 6. OBSLIB++, itself, extends the Finite Element toolbox UG [BBJ$^+$97] in order to assemble and to solve nonsmooth minimization problems.
The computational domains of the presented examples are CAD based unstructured grids provided in EXODUS-II format [SY94]. Moreover, the boundary conditions and necessary parameters are given in EXODUS-II PARAMETER FORMAT [GK08a].

5.6 Non-Linear Elasto-Static PDEs

The convergence analyses for the presented globalization strategies of this thesis have in common, that we assume that the minimization problem (M) has a solution. In the context of the static border case in elasticity it is sufficient to assume that a stored energy function satisfies the assumptions of

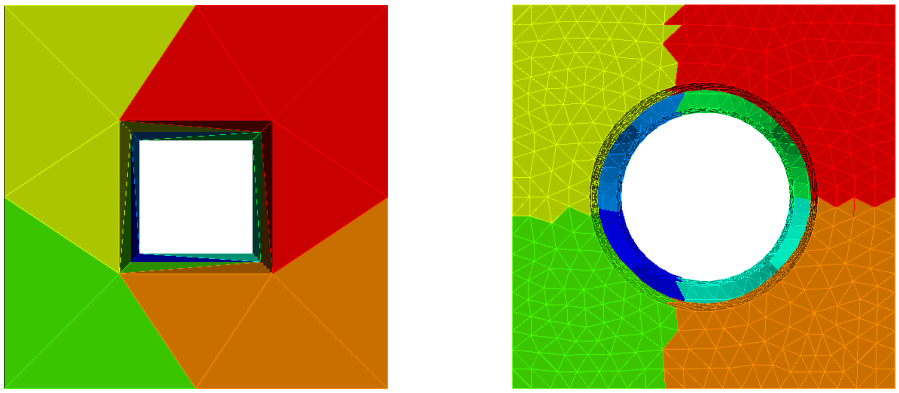

Figure 5.2: Left: the computational domain for the Dirichlet value problem of Section 5.6.3. Right: the computational domain for the contact problem of Section 5.6.6.

Theorem 1.2.3. As it turns out, the well-known and most simple nonlinear stored energy function, for St. Venant-Kirchhoff materials does not satisfy these assumptions [Rao86] and is, thus, not suited for our numerical examples.
Therefore, we will focus on a class of objective functions, introduced by R.W. Ogden [Ogd97], which satisfies the assumptions of Theorem 1.2.3:

$$\widehat{\boldsymbol{W}}(\boldsymbol{C}(\boldsymbol{u})) = 3(a+b) + (2a+4b)\cdot \mathrm{tr}\boldsymbol{E} + 2b\cdot(\mathrm{tr}\boldsymbol{E})^2 - 2b\cdot\mathrm{tr}(\boldsymbol{E}^2) + \Gamma(\det(\nabla\boldsymbol{\varphi})) \tag{5.6.1}$$

where $\boldsymbol{C}(\boldsymbol{u}) = \frac{1}{2}(I+\nabla\boldsymbol{u})^T(I+\nabla\boldsymbol{u})$ is the Green-St.Venant strain tensor, $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C}-I)$ and $\boldsymbol{\varphi} = \mathrm{Id}+\boldsymbol{u}$ is the deformation tensor and $\Gamma(\delta) = c\delta^2 - d\log\delta$ a *logarithmic barrier function*. The constants are chosen as follows

$$a = \frac{\mu}{2} - \frac{\lambda}{8},\ b = \frac{\mu}{2} + \frac{\lambda}{4},\ c = \frac{\lambda}{8} \text{ and } d = \frac{\lambda}{2} + \mu \tag{5.6.2}$$

The Lamé constants $\mu > 0$ and $\lambda > 0$ will be chosen problem dependent.
This material law describes the behavior of a compressible Mooney-Rivlin material (cf. Section 4.10 [Cia88]). But, moreover, following Theorem 4.10-2 [Cia88], this material law and its parameters have the following properties

- for $\|E\| \to 0$ this material law converges to the St. Venant-Kirchhoff material law
- this material law is polyconvex and satisfies the coercivity inequality from Section 1.2.2

Moreover, this stored energy function is twice continuously differentiable in $\boldsymbol{u}$ and satisfies the assumption on the levelsets.

Lemma 5.6.1. *Suppose that $\widehat{\boldsymbol{W}} : \mathcal{X} \to \mathbb{R}$ is given like in (5.6.1), where $\mathcal{X}$ is the Finite Element space from Section 1.3.2. Suppose furthermore that the assumptions on the set of admissible solutions $\Phi_h = \mathcal{X}\cap\Phi$ stated in Theorem 1.2.3[2] hold along with $\Gamma_D \neq \emptyset$ and $J(\boldsymbol{u}_0) < \infty$ with $\boldsymbol{u}_0 \in \Phi_h$. Then the levelset*

$$\mathcal{L} = \{\boldsymbol{u} \in \Phi_h \mid J(\boldsymbol{u}) \leq J(\boldsymbol{u}_0)\}$$

[2]The definition of Φ and that it is non-empty

where

$$J(\boldsymbol{u}) = \int_\Omega \left(\widehat{\boldsymbol{W}}(\boldsymbol{C}(\boldsymbol{u})) - \rho \boldsymbol{F} \cdot \boldsymbol{u} \right) d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f} \cdot \boldsymbol{u} \, ds_{\boldsymbol{x}}$$

is compact.

Proof. Due to Theorem 4.10-2 [Cia88] and the reasoning as in the proof of Theorem 7.7-1 [Cia88], the following coercivity relation holds

$$J(\boldsymbol{u}) \geq c\|\boldsymbol{u} + \mathrm{Id}\|^2_{H^1(\Omega)} + \|\mathrm{Cof}(\nabla \boldsymbol{u} + I)\|^2_{L^2(\Omega)} + (\det(\nabla \boldsymbol{u} + I))^2) + d \qquad \text{for all } \boldsymbol{u} \in \Phi_h$$

where $c > 0$ and $d \in \mathbb{R}$.
Now assume that $\mathcal{L}$ is not bounded. Then there exists a sequence $(\boldsymbol{u}_k)_k$ in $\mathcal{L}$ with $\|\boldsymbol{u}_k\|_{L^2(\Omega)} \to \infty$. Due to the coercivity this implies that $J(\boldsymbol{u}_k) \to \infty$, which contradicts $J(\boldsymbol{u}_0) < \infty$.
Now suppose that $\mathcal{L}$ is not closed. In this case, there exists a sequence of Finite Element functions $(\boldsymbol{u}_k)_k$ in $\mathcal{L}$ such that $\boldsymbol{u}_k \to \tilde{\boldsymbol{u}} \notin \mathcal{L}$. which means that $J(\tilde{\boldsymbol{u}}) > J(\boldsymbol{u}_0)$. This implies that there must exist for all $\varepsilon > 0$ an index ν_0 such that $\|\tilde{\boldsymbol{u}} - \boldsymbol{u}_k\|_{L^2(\Omega)} \leq \varepsilon$ for almost all $k \geq \nu_0$. The continuity of J and the finite dimension of $\mathcal{X}$ now gives rise to the fact that for small ε also $J(\boldsymbol{u}_k) > J(\boldsymbol{u}_0)$ holds, which contradicts $\boldsymbol{u}_k \in \mathcal{L}$. □

Remark 5.6.2. *Since $\mathbb{R}^n$ is isomorphic to $\mathcal{X}$, we obtain that also the discrete levelsets are compact. Also for each subset $\mathcal{D}_k$ the compactness of $\mathcal{L}_k$ can be shown employing the same reasoning, if $P_k u^\nu$ and $P_k^{k+1} u_k$ must satisfy the assumptions of this theorem. Though, as pointed out in [GK08b], restricted iterates in a Finite Element multigrid context might not satisfy these assumptions. In contrast, in the presented examples of this section, we obtain that projected iterates satisfy the assumptions which yields that also for the multiplicative strategies the convergence results hold.*
Moreover, as we have shown in Theorem 1.3.1, also in the dynamic case the coercivity condition holds. Employing the same argumentation as in the previous lemma, one can deduce that each levelset $\mathcal{L}^{(t_i)}$ is compact.

Lemma 5.6.3. *Suppose that $\widehat{\boldsymbol{W}} : \mathcal{X} \to \mathbb{R}$ is given like in (5.6.1), where $\mathcal{X}$ is the Finite Element space with linear basis functions from Section 1.3.2. Suppose furthermore that the assumptions on the set of admissible solutions $\Phi_h = \mathcal{X} \cap \Phi$ stated in Theorem 1.2.3[3] hold along with $\Gamma_D \neq \emptyset$ and $J(\boldsymbol{u}_0) < \infty$ with $\boldsymbol{u}_0 \in \Phi_h$. Then there exists a constant $C > 0$ such that for all $\boldsymbol{u} \in \mathcal{L}$ as defined in the previous lemma,*

$$\|\nabla^2 \widehat{\boldsymbol{W}}(\boldsymbol{C}(\boldsymbol{u}))\| \leq C$$

Proof. Since $\boldsymbol{u}$ is an element from the Finite Element space $\mathcal{X}$ and since $\boldsymbol{E}$ is a polynomial in the components of $\nabla \boldsymbol{u}$, we obtain that there exists an $c > 0$ such that

$$\| \frac{\partial^2}{\partial \boldsymbol{u}^2} \left(3(a+b) + (2a+4b) \cdot \mathrm{tr}\boldsymbol{E} + 2b \cdot (\mathrm{tr}\boldsymbol{E})^2 - 2b \cdot \mathrm{tr}(\boldsymbol{E}^2) \right) \| \leq c$$

for all $\boldsymbol{u} \in \mathcal{L}$. On the other hand, we employ

$$\begin{aligned} \frac{\partial^2}{\partial \boldsymbol{u}^2} \Gamma(\det(\nabla \boldsymbol{\varphi}))(\cdot)(\cdot) \quad = \quad & \Gamma''(\det(\nabla \boldsymbol{\varphi})) \det'(\nabla \boldsymbol{\varphi}) \nabla(\cdot) \det'(\nabla \boldsymbol{\varphi}) \nabla(\cdot) + \\ & \Gamma'(\det(\nabla \boldsymbol{\varphi})) \det''(\nabla \boldsymbol{\varphi}) \nabla(\cdot) \nabla(\cdot) \end{aligned}$$

[3] The definition of Φ and that it is non-empty

Since $\det(\nabla\boldsymbol{\varphi})$ is a polynomial of degree 3 in the components of $\nabla\boldsymbol{u}$, which is in turn a piecewise polynomial, we obtain that $\|\det'(\nabla\boldsymbol{\varphi})\|$ and $\|\det''(\nabla\boldsymbol{\varphi})\|$ have a finite value on $\mathcal{L}$. On the other hand, since $\mathcal{L}$ is bounded, and the coercivity condition holds, we obtain that $\det(\nabla\boldsymbol{\varphi}) \geq \varepsilon > 0$ for all $\boldsymbol{u} \in \mathcal{L}$. This in fact, yields that δ^{-1} and δ^{-2} in $\Gamma'(\delta)$ and $\Gamma''(\delta)$ are bounded in $\mathcal{L}$. Moreover, also the derivatives of δ^2 are bounded in $\mathcal{L}$ yielding that the norms of the barrier terms Γ' and Γ'' are bounded in $\mathcal{L}$.

Alltogether this proves the proposition. □

Remark 5.6.4. *Employing the result of the previous Lemma shows that the gradients of $\widehat{\boldsymbol{W}}$ are bounded and Lipschitz continuous. Now we can employ, once more, that $\mathbb{R}^n$ and $\mathcal{X}$ are isomorphic and obtain that the stated assumptions for the respective globalization strategies hold and convergence can be guaranteed.*

As we have seen, the presented globalization strategies aim at the solution of discretized optimization problems of the kind (M). In this section, we will, therefore, focus on the following minimization problem

$$J(Xu) = \min! \qquad \text{in } \Omega \tag{5.6.3a}$$
$$Xu \cdot \boldsymbol{n} \leq \phi \qquad \text{on } \Gamma_C \subset \partial\Omega \tag{5.6.3b}$$
$$Xu = g \qquad \text{on } \Gamma_D \subset \partial\Omega \tag{5.6.3c}$$

(cf., equation (1.3.7)) where

$$J(\boldsymbol{u}) = \int_\Omega \left(\widehat{\boldsymbol{W}}(\boldsymbol{C}(\boldsymbol{u})) - \rho\boldsymbol{F}\cdot\boldsymbol{u}\right) d\boldsymbol{x} - \int_{\Gamma_N} \boldsymbol{f}\cdot\boldsymbol{u}\, ds_{\boldsymbol{x}}$$

But, note that in all of our examples we will employ $\rho = 1$ and $\boldsymbol{F} = 0$.

As a matter of fact, the resulting objective function realizes an interior point approach (for an introduction see [NW06] and [FM90]) to enforce that element volumes will not be inverted. As it turns out, the logarithmic barrier function is an approximation to the indicator function

$$\chi_{\mathcal{B}^+}(\boldsymbol{u}) = \begin{cases} 0 & \text{if } \boldsymbol{u} \in \mathcal{B}^+ \\ \infty & \text{otherwise} \end{cases}$$

of

$$\mathcal{B}^+ = \{\boldsymbol{u} \in H^1(\Omega) \mid \det(\nabla\boldsymbol{u} + I) > 0\}$$

But, the employed logarithmic barrier term yields that (5.6.1) becomes a highly nonlinear objective function, whenever the material is compressed. Therefore, within the iterative solution of a minimization problem which incorporates this barrier function, undamped iterates may violate $\mathcal{B}^+$. Since the barrier function will depend on the discretization, this constraint is closely related to the mesh size. In turn, for relative coarse meshes this constraint does often not yield a step-length limitation. But the finer the mesh becomes, the more problems can be caused by long corrections. Therefore, this argument along with the possible non-convexity of the objective function, stresses the fact that convergence can only be guaranteed if a globalization strategy is employed.

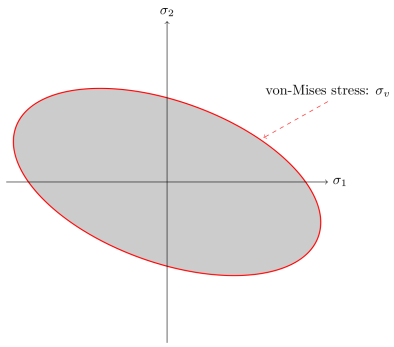

Figure 5.3: Here, we illustrate the von-Mises stress. In case of plane stress, the principal stress components $\hat{\boldsymbol{T}}_{11}$ and $\hat{\boldsymbol{T}}_{22}$, i.e., the first two eigenvalues of the stress tensor, the von-Mises stress describes an ellipse within the $\hat{\boldsymbol{T}}_{11} - \hat{\boldsymbol{T}}_{22}$-plane. In particular, we have $\|\text{dev}\,\hat{\boldsymbol{T}}\|_2 = \hat{\boldsymbol{T}}_{11}^2 - \hat{\boldsymbol{T}}_{11}\hat{\boldsymbol{T}}_{22} + \hat{\boldsymbol{T}}_{22}^2$, where

5.6.1 Visualization

To visualize the computed results, we employ the von-Mises stress distribution which is a well-known tool in plasticity theory. In particular, the von-Mises stress distribution[4] maps from the space of second–order tensors to $\mathbb{R}$ given by

$$\hat{\boldsymbol{T}} \mapsto \|\text{dev}\,\hat{\boldsymbol{T}}\|_2 = \|\hat{\boldsymbol{T}} - \frac{1}{3}(\hat{\boldsymbol{T}} : I)I\|_2$$

In fact, the stress of a material differs under different loading conditions since under our assumptions the stress tensor itself possesses six degrees of freedom. Thus, to each other "equivalent" stresses would normally yield different visualizations. The von-Mises stress, in contrast, maps equivalent stresses to the same distribution.

Though, the von-Mises stress distribution is a fictitious stress distribution and is well-suited to make predictions if a material is bended or skewed. In contrast, this criterion should not be applied, if the stresses in all principal directions, i.e., in direction of the eigenvectors of the stress tensor, are equally large. In this case, it may occur that $\|\text{dev}\,\hat{\boldsymbol{T}}\|_2 \approx 0$ but $\|\hat{\boldsymbol{T}}\|_2 \gg 0$.

However, the visualization itself was carried out in two steps. We employ IOLIB from OBSLIB++ (cf., Section 6.3) to export the current mesh, displacements and the von-Mises stress distribution. This enables us to visualize, in a secnod step, all data employing PARAVIEW [Tea09].

Computing Initial Iterates

Stored energy functions of Ogden-type, such as (5.6.1) often yield particular numerical challenges if displacements are prescribed for the solution at Γ_D, which is the case when solving a so-called Dirichlet value problem. In the case of linear elasticity, the solution of such minimization problems can be carried out straight-forwardly, independent from $\mathcal{B}^+$. The linear elastic material law is given by

$$\widehat{\boldsymbol{W}}(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{u}) \tag{5.6.4}$$

[4] Sometimes also referred to as equivalent tensile stress or distortion strain energy

Figure 5.4: Left: the computational domain for the Dirichlet value problem of Section 5.6.4. Right: the computational domain for the contact problem of Section 5.6.7.

where $\varepsilon(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u}^T + \nabla \boldsymbol{u})$ denotes the linearized Green–St. Venant strain tensor and

$$\boldsymbol{\sigma}(\varepsilon) = \frac{E}{1+\nu}\varepsilon(\boldsymbol{u}) + \frac{\nu}{1-2\nu}\mathrm{tr}(\varepsilon(\boldsymbol{u}))\, I$$

Hooke's tensor [Cia88, Bra07].
However, in the case of nonlinear elasticity and the context of our examples it is of paramount importance that $\boldsymbol{u}_0 \in \mathcal{B}^+$. But, even if the initial solution on a relatively coarse mesh does not violate this constraint, for realistic resolutions of the computational domain, this is generally not the case.
Therefore, two different strategies are often employed to compute admissible iterates for large–scale optimization problems. On the one hand one might employ nested iteration. In this case, (5.6.3) is solved based on a certain discretization. Then the underlying mesh is refined and the current solution interpolated yielding a start iterate for the finer problem. On the other hand, one might first solve an easier computable problem, e.g., with (5.6.4) as stored energy function, which usually provides an admissible start iterate for the fine level solution process. In our examples, we, therefore, solve the linear elastic model problem (employing the same Lame parameters) on the coarsest level and interpolate the computed linear solution to the level, where the nonlinear minimization problem (5.6.3) should be solved.

5.6.2 The Nonlinear Update Operator

In our examples, each local update operator $\mathcal{F}_k(u)$ is constituted by four, in the additive case asynchronous, Trust-Region or Linesearch steps, respectively. On the other hand, we employ four Trust-Region or Linesearch steps as postsmoother in order to compute s^ν in (3.1.6).
The respective search directions are computed employing a parallelized projected cg method along with a parallel symmetric non-linear Gauß-Seidel preconditioner [Kra07a]. Basically this Gauß-Seidel preconditioner works just like a sequential symmetric non-linear Gauß-Seidel method. On each processor, the symmetric Gauß-Seidel iteration is performed but without parallel communication. After the local iteration, a parallel update takes place, enhancing the overall convergence tremendously. In turn, during the iterative solution process for problems like (M) thousands of cg iterations must be computed yielding as many parallel communication calls. Though, the overall method behaves for a small number of processors similar to a cg-method employing the traditional symmetric Gauß-Seidel method as smoother. The cg-method, itself, is parallelized just employing parallelized linear algebra.

In contrast, during the parallel solution process, we employ the cg method in combination with a local sequential symmetric nonlinear Gauß-Seidel method, which does not employ any parallel communication. Both solvers are employed to compute Quasi-Newton corrections by means of possibly outdated Hessians. In fact, we reassemble the exact Hessian whenever the current Hessians become outdated. This, in turn, is measured by means of (6.1.1), a heuristic which will be introduced in Section 6.1. Though, we will employ $\mathcal{F}'(u) = I$ within each global $B^{\nu}_{k,i}$ as proposed in Section 3.1. In turn, we compute several Trust-Region or Linesearch steps in order to obtain a good global correction s^{ν}.
As a matter of fact, this linear solver is generally not suited for the solution of indefinite and negative definite linear systems of equations. But, to guarantee convergence, we employ the Cauchy criterion (2.1.9) and (2.2.3), respectively. This means, that if the correction or search direction does not satisfy these conditions, we simply choose the Cauchy point as direction.
However, the symmetric Gauß-Seidel method cannot handle coupled constraints straight-forwardly. As it turns out, even if the obstacle itself is a plane, the plane's normal generally does not direct in the direction of the employed basis functions which yields coupled constraints. To avoid this problem, one might employ the approach from Section 1.3.2 to rotate the basis functions prior to the solution process into the normal tangential system of the obstacle (cf., for instance [Kra01]).

The Additive Framework

In the additive framework, we decompose the computational domain Ω into N non–overlapping subsets Ω_k where N is the employed number of processors. In fact, in all computed examples we employed $N = 8$ processors. This yields a decomposition of the coefficient space $\mathbb{R}^n$ as presented in Section 3.1.6.
The local objective function is then given by

$$J^{\nu}_k(u_k) = J(u_k, u^{\nu}_{\overline{k}})$$

where $u^{\nu}_{\overline{k}} = (u^{\nu})_{\{1,\dots,n\}-C_k}$ are the coefficients of u_k which are not represented on $\mathcal{D}_k$. Note that, from now on we consider the solution of the discretized system (1.3.7a).
This particular objective function is reasonable in the context of Ogden materials, since the barrier term must be computed employing outdated unknowns at the processor interface. Setting these unknowns to zero would generally cause that the barrier function is not defined at u_k.
Moreover, since each basis function has a strictly local support, the assembling process can also be carried out strictly local. Therefore, in order to asynchronously compute J^{ν}_k, the assembler just needs the (outdated) information of the unknowns at neighboring elements.

Remarks on the Expected Numerical Behavior

In Chapter 4, we have seen that both globalization strategies, APTS and APLS, aim at a solution of local minimization problems which are closely related to (M). In case of a domain decomposition, as presented in Section 3.1.6, the additive preconditioning strategies quickly smooth the local nonlinear residuals. As a consequence, within the interior of each $\mathcal{D}_k$ the error becomes small. But on the other hand, at the domain interfaces, the residual might increase since on two different domains corrections were computed without parallel communication.
As it will turn out, in our computations this might have different effects. In most cases the convergence rates are significantly improved, even if step-length limitations at the domain interfaces might occur. On the other hand, low frequency contributions of the solution, such as rigid body motions,

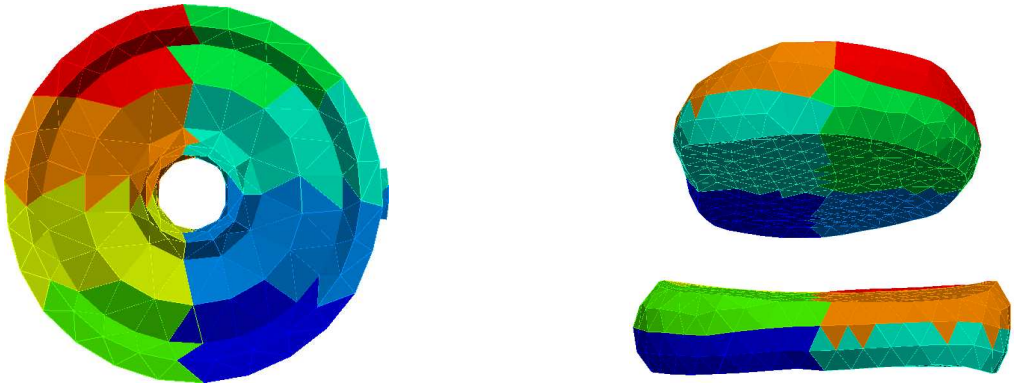

Figure 5.5: Left: the computational domain for the Dirichlet value problem of Section 5.6.5. Right: the computational domain for the contact problem of Section 5.6.8.

are poorly resolved by the additive framework which causes a disturbing effect. Therefore, employing also a nonlinear multigrid strategy as a multiplicative preconditioning scheme to resolve these motions improves the rates of convergence significantly.

The Multiplicative Framework

In the multiplicative framework, the Finite Element space $\mathcal{X}$ is hierarchically decomposed as follows

$$\mathcal{X} = \mathcal{X}_0 \supset \ldots \supset \mathcal{X}_N$$

The transfer operator for the primal variables is the approximation to the discretized L^2 projection as presented in (3.1.18). The interpolation operator is given as in (3.1.12) and the local objective function is given by

$$J_k^\nu(u_k) = J(X_k u_k)$$

As a matter of fact, the coarse level objective function can be evaluated directly on the current level employing a quadrature rule on the current grid. Moreover, the coarse level problems are solved employing two iterations of the traditional Trust-Region or Linesearch schemes.

Remarks on the Expected Numerical Behavior

In our numerical examples, we observe that the application of a coarse level generally speeds up the convergence with less computational overhead. In particular, if we combine additive and multiplicative schemes, in most examples we observe the fastest measured convergence. This results from the well-balanced combination of an "exact" solution of local problems in combination with an improved resolution of low frequency contributions, such as rigid body motions.

Parameter Choices

Within the sufficient decrease condition (2.1.9) of the Trust-Region algorithm we employ $\beta = 0.5$. Together with $\eta = 0.1$ and $\gamma_1 = 0.1$ and $\gamma_2 = 2$, the constants of the Trust-Region method are given. Within the Linesearch algorithm we employ $\rho_A = 0.1$, as well as $\rho_{AP} = \rho_{MP} = 0.9$. Moreover, we define $\beta_{ls} = 100$ and $\eta_{ls} = 0.1$. In all algorithms, the additive and multiplicative schemes, we employ $\kappa_g = \frac{1}{n\sqrt{n}} \leq \frac{1}{\sqrt{n}}$ in (5.1.1) to ensure a "uniform" convergence of the first-order conditions.

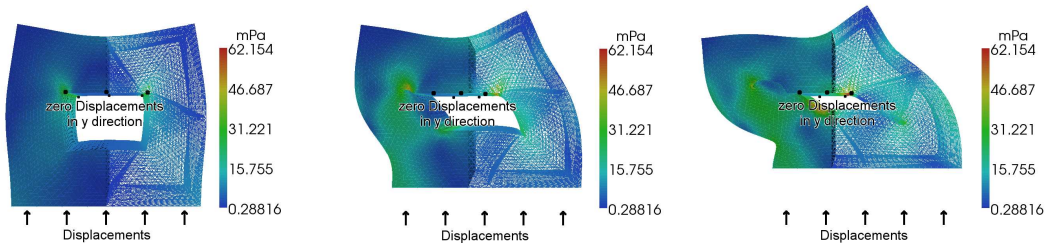

Figure 5.6: **Unconstrained Minimization Problem: Compression of a Cube**. The computed solution of problem from Section 5.6.3 with $135,456$ degrees of freedom. At the hole's upper side we applied zero Dirichlet values in all directions (as indicated by the dots), and at the cube's lower side displacements of 10%, 20%, 30% respectively, of the cube's size (as indicated by the arrows). Colors are the local von-Mises stresses.

Finally, we define a stop-criterion for the respective globalization strategies. This means, that we stop our computation if the first–order conditions $\|g^{\nu}_{j,m_j}\|_2 \leq \varepsilon$ for problem-dependently chosen $\varepsilon > 0$.

Comparing the Schemes

Due to the computational overhead of the preconditioned schemes, it becomes hard to compare the traditional and the preconditioned globalization strategies with each other. In order to compare the schemes we measure the employed outer iterations, the cg iterations and the normed, expected computation time of the schemes. To derive the expected computation time we employ the worst-case scenario: experiments show that the asynchronous strategies employing 4 Trust-Region or Line-search steps along with 25 cg iterations for the computation of search directions need 135% of the time which traditional schemes with 100 cg iterations and 4 globalization steps consume. Therefore the additive schemes need per cycle 1.35 times the computation time of the traditional scheme. Similarly the multiplicative strategies need generally 1.15 times longer per iteration than the traditional schemes.
In fact, in our comparison each asynchronous cg iteration is weighted like a synchronous iteration. As we have seen in our numerical experiments, this observation holds true for eight-core machines, like the employed ones. Therefore, due to the massively employed parallel communication, we expect that on faster machines with competitive implementations, the computation time for each APTS/APLS cycle is considerably faster than the factor of 1.35.

5.6.3 Unconstrained Minimization Problem: Compression of a Cube

As the first numerical example, we consider the solution of problem (5.6.3) employing a discretization with $135,456$ unknowns. In this example, we apply displacements at the lower side of the domain shown in the left image of Figure 5.2. On the other hand, at the upper side of the geometry's hole, we apply zero displacements. All other boundary conditions are chosen as natural conditions. Therefore, $\partial\Omega$ is divided into $\Gamma_D = \{(x,y,z) | -0.5 = x \vee 0.15 = x\} \cap \partial\Omega$, $\Gamma_N = \partial\Omega \backslash \Gamma_D$ with

$$g((x,y,z)^T) = \begin{cases} (d,0,0)^T & \text{if } x = 0.5 \\ 0 & \text{otherwise} \end{cases}$$

where $d = 0.2, 0.25, 0.3$. Due to the absence of contact conditions, we choose $\underline{\phi} = (-10^6, \ldots, -10^6)^T$ and $\overline{\phi} = (10^6, \ldots, 10^6)^T$. The material parameters, i.e., the Lamé constants, are chosen as $E = 300[mPa]$ and $\nu = 0.3$. To derive λ and μ one can employ the following

	Example	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	20%	11	4,400	0	1.0
APTS		7	2,800	700	0.85
MPTS		7	2,800	0	0.73
AMPTS		7	2,800	700	0.98
Trust-Region	25%	30	12,000	0	1.0
APTS		22	8,800	2,200	0.99
MPTS		27	10,800	0	1.01
AMPTS		18	7,200	1,800	0.93
Trust-Region	30%	64	25,600	0	1.0
APTS		42	16,800	4,200	0.88
MPTS		62	24,800	0	1.11
AMPTS		38	15,200	3,800	0.92
Linesearch	20%	8	3,200	0	1.0
APLS		8	3,200	800	1.35
MPLS		7	2,800	0	1.0
AMPLS		6	2,400	600	1.16
Linesearch	25%	21	8,400	0	1.0
APLS		16	6,400	1,600	1.02
MPLS		19	7,600	0	1.09
AMPLS		11	4,400	1,100	0.81
Linesearch	30%	42	16,800	0	1.0
APLS		31	12,400	3,100	0.99
MPLS		34	13,600	0	0.93
AMPLS		20	8,000	2,000	0.73

Table 5.1: **Unconstrained Minimization Problem: Compression of a Cube**. Runtime comparisons of the globalization strategies for the respective examples

formulas

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \text{ and } \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$$

Figure 5.6 shows the numerical result of this simulation: the reaction of this cube-like geometry to three different kinds of pressure.
This simulation is carried out employing the traditional Linesearch and Trust-Region strategies, as well as the preconditioned strategies. The stop criterion is $\varepsilon = 1e-4$. In Figure 5.7 we compare the numerical behavior of the Trust-Region and preconditioned Trust-Region strategies. In Figure 5.8 we compare the Linesearch schemes. Table 5.1 shows the runtime comparisons for the respective schemes.

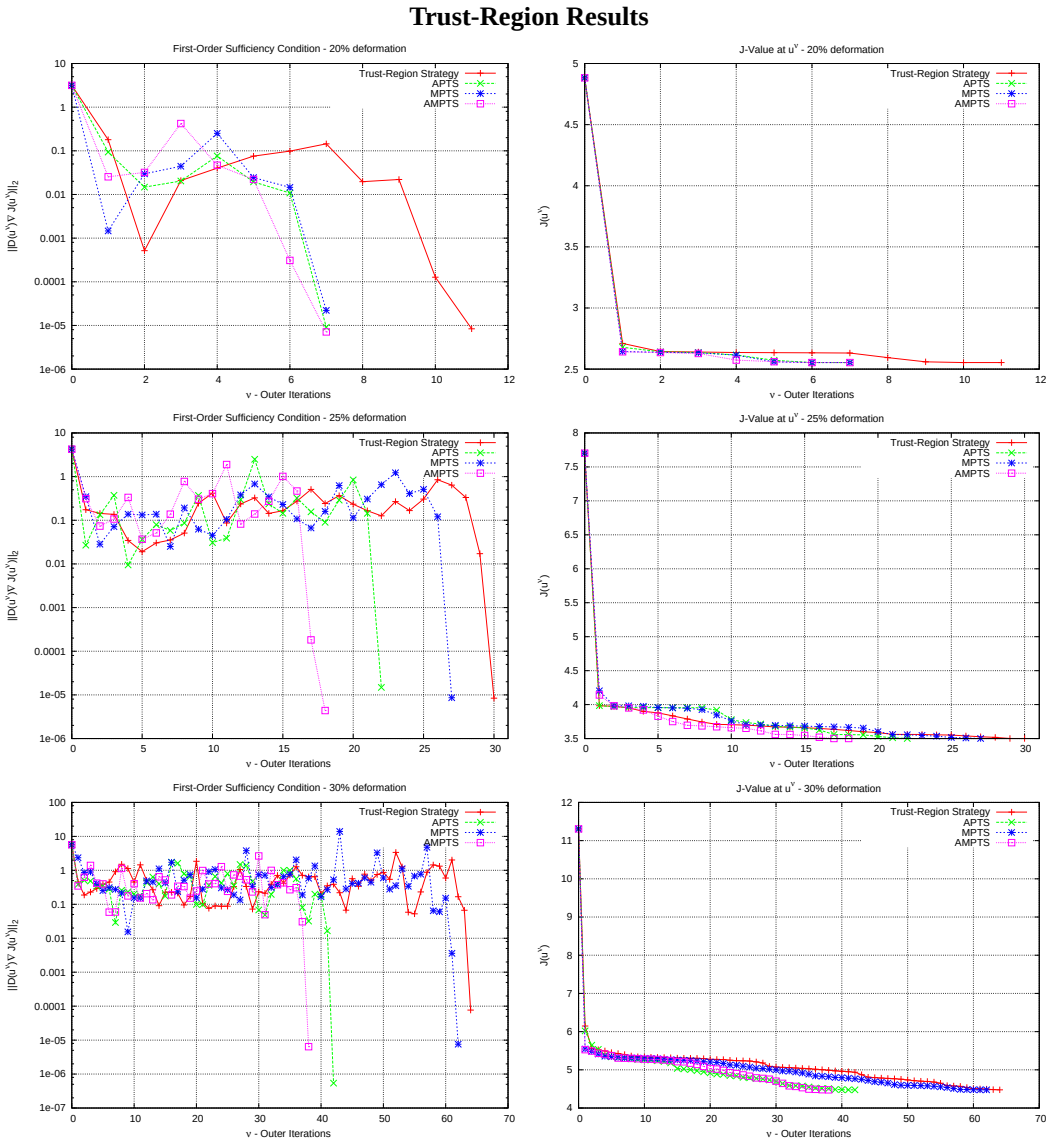

Figure 5.7: **Unconstrained Minimization Problem: Compression of a Cube**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.3 with a **Trust-Region strategy** and the **preconditioned Trust-Region strategies**, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

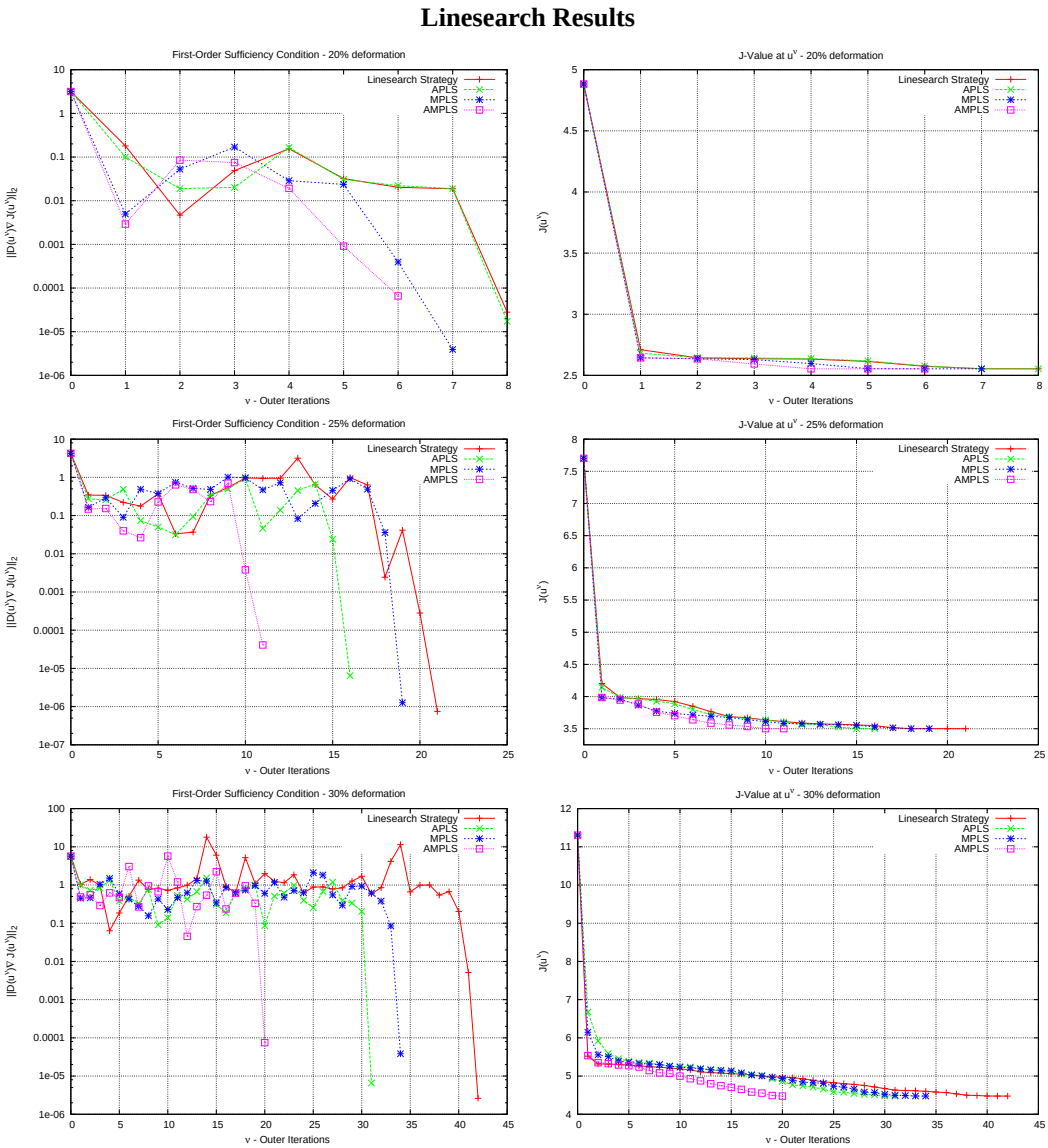

Figure 5.8: **Unconstrained Minimization Problem: Compression of a Cube**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.3 with a **Linesearch strategy** and the **preconditioned Linesearch strategies**, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

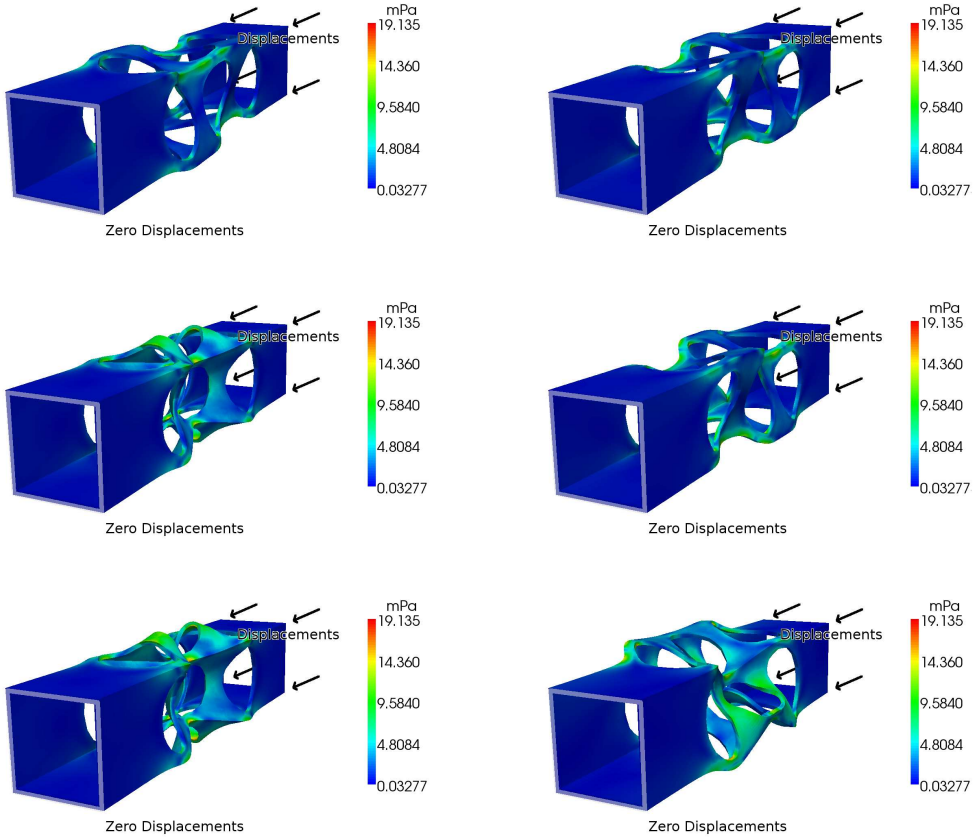

Figure 5.9: **Unconstrained Minimization Problem: Simulation of a Can**. The computed solution of the problem from Section 5.6.4 with $330,999$ degrees of freedom. At the visible end of the geometry, we apply zero Dirichlet values (as indicated by the grey lines). Though, at the opposite side we apply displacements of 10%, $12,5\%$ and 15% of the geometries length respectively, (as indicated by the arrows). In each computation we obtain two different solution. In the first row, we see two computed solutions for 10% displacements. Similarly the second and third line show two possible solutions for 12.5% and 15% displacements. Colors are the local von-Mises stresses.

5.6.4 Unconstrained Minimization Problem: Simulation of a Can

Within this example, we simulate a rectangular structure, as shown in the left image of Figure 5.4. Obviously, a mesh, which provides a good approximation to the depicted circular holes has thousands of degrees of freedom. In our case, the coarse mesh provides approximately 10,000 unknowns. After uniformly refining this mesh twice we obtain the final problem with 330,999 unknowns. Therefore, our multilevel hierarchy consists of three levels and the domain decomposition of eight domains.

In our computations, we prescribe displacements at the left side of the geometry and zero displacements at the opposite side. All other boundary values are left natural. As it turns out, the applied boundary values cause a compression of the geometry of 10%, 12.5% and 15%. In particular the

	Example	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	10%	73	29,200	0	1.0
APTS		60	24,000	6,000	1.10
MPTS		62	24,800	0	0.97
AMPTS		56	22,400	5,600	1.19
Trust-Region	12.5%	156	62,400	0	1.0
APTS		109	43,600	10,900	0.94
MPTS		48	19,200	0	0.35
AMPTS		40	16,000	4,000	0.39
Trust-Region	15%	156	62,400	0	1.0
APTS		107	42,800	10,700	0.92
MPTS		105	42,000	0	0.77
AMPTS		84	33,600	8,400	0.83
Linesearch	10%	46	18,400	0	1.0
APLS		43	17,200	4,300	1.26
MPLS		39	15,600	0	0.97
AMPLS		35	14,000	3,500	1.18
Linesearch	12.5%	74	29,600	0	1.0
APLS		90	36,000	9,000	1.64
MPLS		36	14,400	0	0.55
AMPLS		38	15,200	3,800	0.79
Linesearch	15%	118	47,200	0	1.0
APLS		100	40,000	10,000	1.14
MPLS		72	28,800	0	0.70
AMPLS		58	23,200	5,800	0.76

Table 5.2: **Unconstrained Minimization Problem: Simulation of a Can**. Runtime comparisons of the traditional and preconditioned strategies for the respective examples. Note that bifurcations take place, which yield heavily varying runtimes.

boundary values are at $\Gamma_D = \{(x, y, z) \mid x \in \{-1, 1\}\}$ given by

$$g((x, y, z)^T) = \begin{cases} (d, 0, 0)^T & \text{if } x = 1.0 \\ 0 & \text{if } x = -1.0 \end{cases}$$

where $d = 0.20, 0.25, 0.3$. In this example, similarly to the previous one, we employ $E = 300[mPa]$ and $\nu = 0.1$, i.e., material parameters for a very soft material. Here, we also choose $\underline{\phi} = (-10^6, \ldots, -10^6)^T$ and $\overline{\phi} = (10^6, \ldots, 10^6)^T$.
This boundary value problem is sensitive for large strains. In fact, each computation led to two different solutions of the minimization problems. In turn, this influences the convergence behavior of the respective globalization strategy. In particular, the multiplicative schemes compute a different solution than the additive and the traditional schemes. This fact, makes the respective strategies, in particular the computation times, hard to compare. However, a survey of the convergence behavior of the respective methods is given in Figure 5.10 and Figure 5.11. Here, the stop criterion was chosen as $\varepsilon = 1e - 5$. Computation times, cg iterations and outer iterations are shown in Table 5.2.

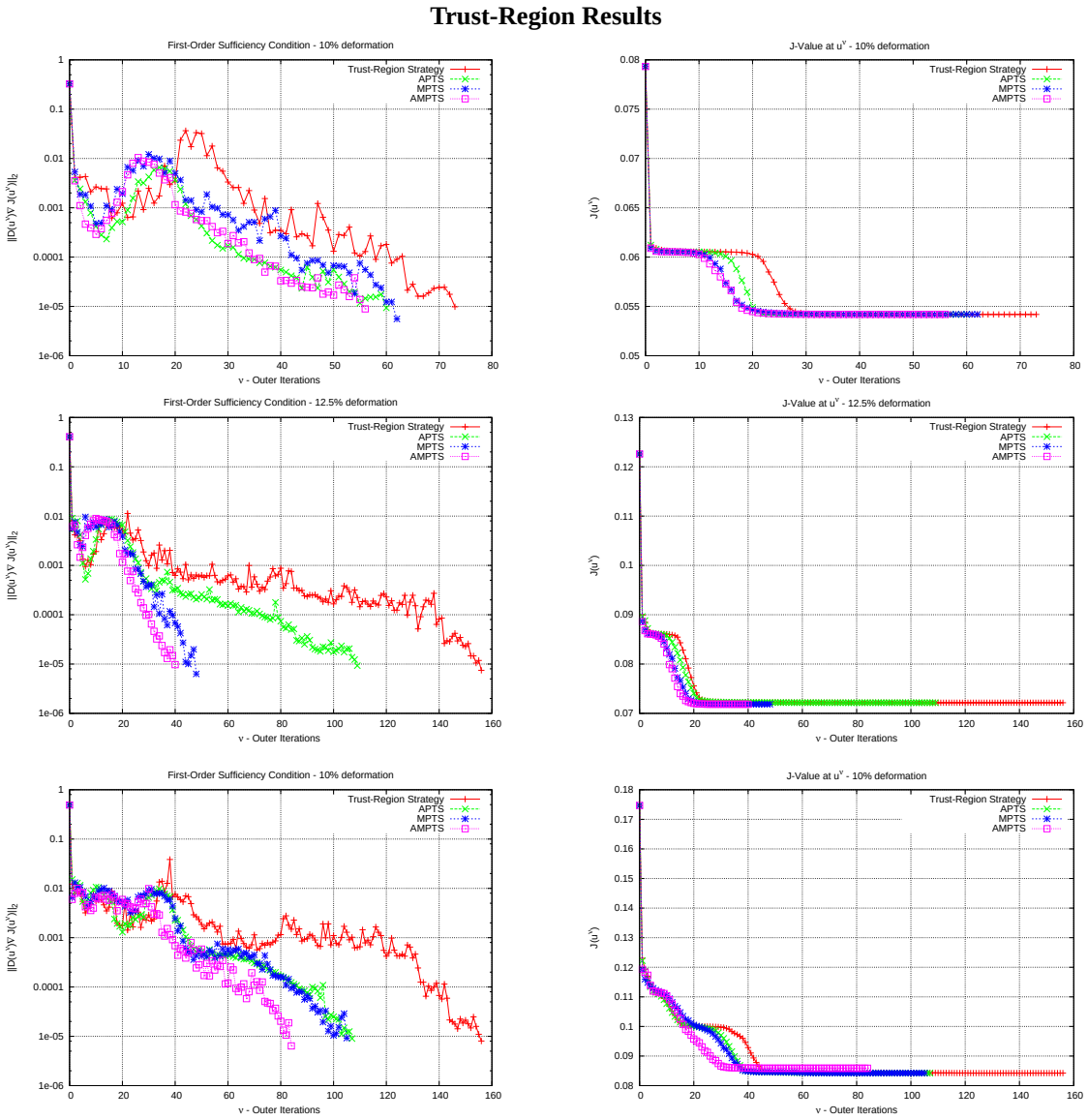

Figure 5.10: **Unconstrained Minimization Problem: Simulation of a Can**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.4 with a traditional and the preconditioned **Trust-Region** strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

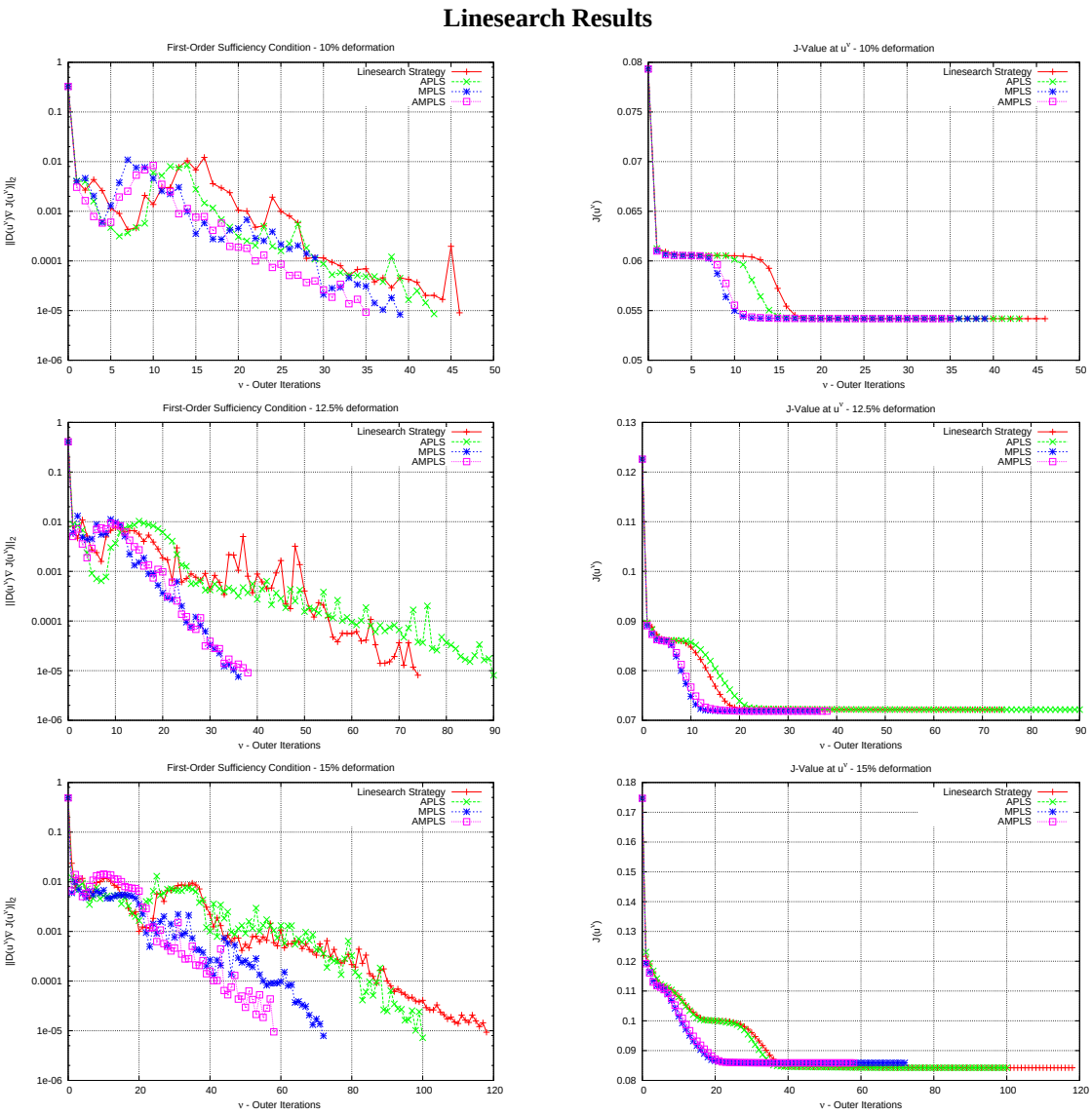

Figure 5.11: **Unconstrained Minimization Problem: Simulation of a Can**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.4 with a traditional and the preconditioned **Linesearch** strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

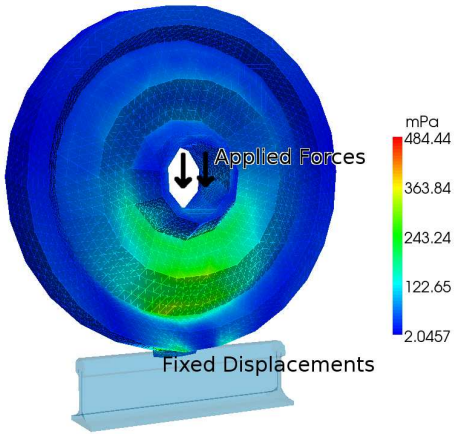

Figure 5.12: **Unconstrained Minimization Problem: Simulation of an Iron wheel**. This boundary value problem is solved employing 40,488 unknowns and eight processors. As indicated in this figure, we apply forces (Neumann values) at the inner side of the wheel's shaft. Moreover, in a small region on the lower side of the wheel we have fixed displacements, approximating contact subject to friction. The colors are the von-Mises stresses, as introduced at the beginning of this chapter.

5.6.5 Unconstrained Minimization Problem: Simulation of an Iron wheel

The simulation of stresses within tires and wheels is of enormous relevance for material scientists. In some applications it is of particular interest, how a tire or wheel reacts on strains, as in our example. Here, we employ a wheel-shaped geometry [NZ01] and Ogden's material law to compute strain-induced stresses, as indicated in Figure 5.12. To simulate the contact between wheel and track, we employ Dirichlet values at the lower side of the geometry, but apply forces at the interior of the geometry.
The employed material parameters are $E = 21[gPa]$, $\nu = 0.3$. The forces are applied at the inner surface of the axis shaft. Here, the force vector itself is given by

$$f((x, y, z)^T) = (-2, 0, 0)^T$$

At all other boundaries, except for a small region next to the track, we applied zero boundary values. Here, we also choose $\underline{\phi} = (-10^6, \ldots, -10^6)^T$ and $\overline{\phi} = (10^6, \ldots, 10^6)^T$.
Figure 5.13 shows that due to the geometry itself and the stated problem, the multiplicative schemes rapidly compute solutions for the local minimization problem. It seems that the coarse level problem itself provides a good solution for the fine level problem. On the other hand, even if the APTS strategy is five times slower than the multiplicative schemes, it succeeds in computing a local minimizer in iteration 197, right before the limit of 200 outer iterations. Moreover, as it can be seen, the stop criterion is $\|\hat{g}^{\nu}_{j,i}\|_2 \leq 0.1$. This is due to the chosen large Young's modulus which yields large function values. In turn, we reach the regions of computational accuracy when the error gets into the region of 10^{-1}.

	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	>200	> 80,000	1.0	
APTS	197	78,800	19,700	< 1.32
MPTS	30	12,000	0	<0.17
AMPTS	28	11,200	2,800	<0.21
Linesearch	181	72,400	0	1.0
APLS	187	74,800	18,700	1.39
MPLS	32	12,800	0	0.2
AMPLS	31	12,400	3,100	0.26

Table 5.3: **Unconstrained Minimization Problem: Simulation of an Iron wheel**. Runtime comparisons of the globalization strategies for the example of Section 5.6.5.

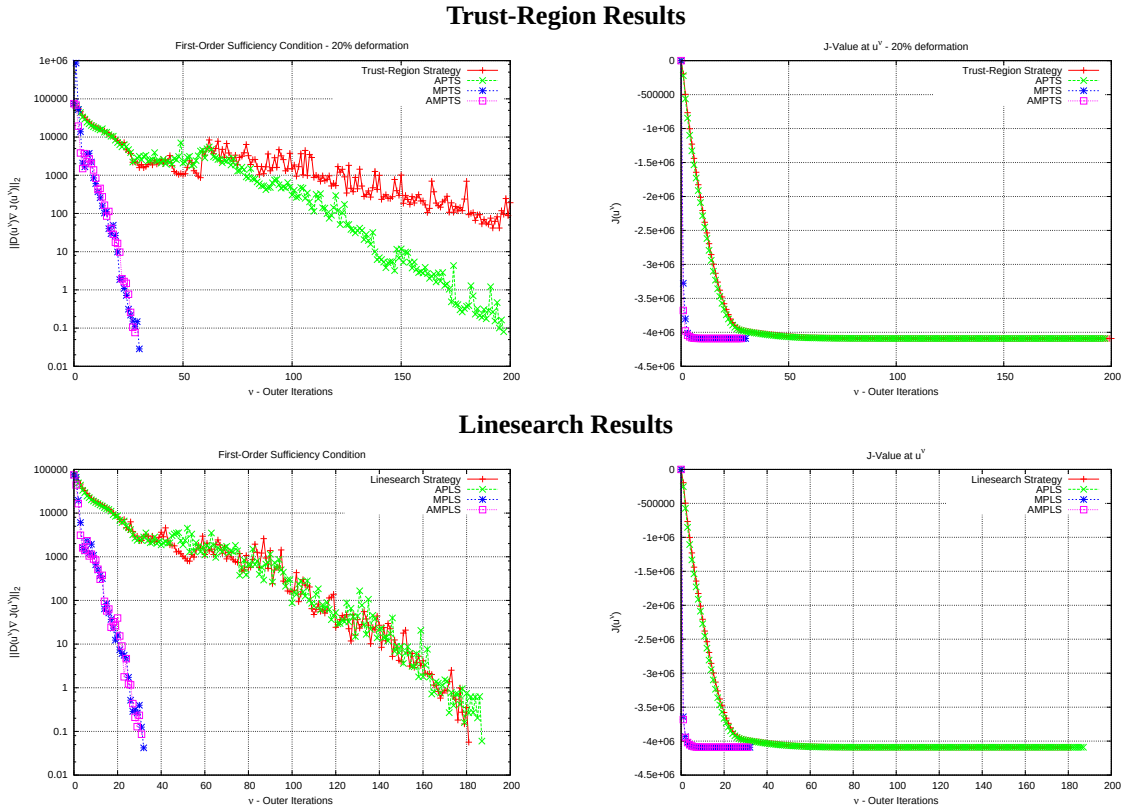

Figure 5.13: **Unconstrained Minimization Problem: Simulation of an Iron wheel**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.5 with a Trust-Region and Linesearch strategy and the preconditioned strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

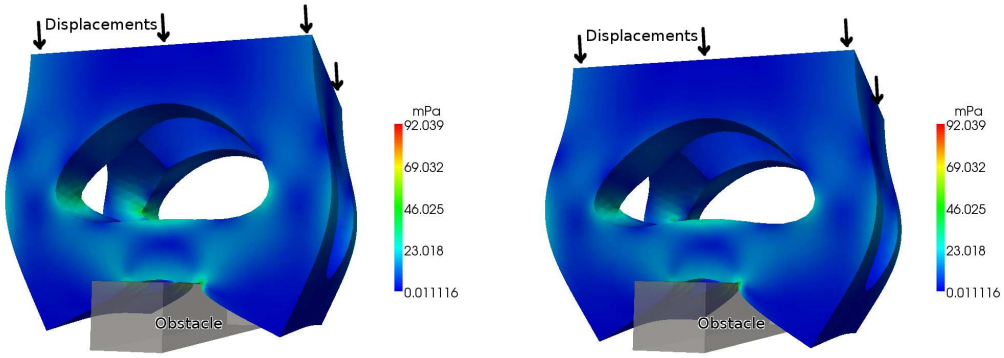

Figure 5.14: **Contact with a Small Obstacle**. *Left image:* Solution of the problem from Section 5.6.6 with $988,392$ degrees of freedom. As indicated, we apply displacements of 10% of the cube's length at the top of the cube (indicated by the arrows). On the other hand, an obstacle is located at the middle of the cube's bottom (as indicated by the grey rectangle). *Right image:* Here, we double the applied displacements to 20% yielding the displayed result.

5.6.6 Constrained Minimization Problem: Contact with a Small Obstacle

This it the first example where we employed a linearized obstacle along with prescribed displacements at the Dirichlet boundary. Both together yields a compression of the given cube-like geometry of up to 20% of the length of the geometry.
The problem description is as follows. In this example, we solved the minimization problem (5.6.3) on the domain

$$\begin{aligned}\overline{\Omega} = \{(x,y,z) | -0.5 \leq x,y,z \leq 0.5 \quad &\wedge \quad ((x = \pm 0.5) \wedge \neg(y^2 + z^2 \leq 0.5)) \\ &\wedge \quad ((y = \pm 0.5) \wedge \neg(x^2 + z^2 \leq 0.5)) \\ &\wedge \quad ((z = \pm 0.5) \wedge \neg(x^2 + y^2 \leq 0.5))\}\end{aligned}$$

as shown in the right image of Figure 5.2. The Dirichlet boundary is the entire upper side of the cube, i.e., $\Gamma_D = \{(x,y,z) \mid z = 0.5\} \cap \partial\Omega$. The boundary values are $\boldsymbol{g}(x,y,z) = (0,0,-d)$ with $d = 0.1$ and $d = 0.2$. All other boundaries have natural Neumann conditions. The contact boundary is an unsymmetrical obstacle (visualized by the bar in Figure 5.14) at the bottom of the geometry. The geometry and the obstacle stay initially in contact, i.e., $\underline{\phi}_k = 0$ and $\overline{\phi}_k = 10^6$. Here, we choose $\underline{\phi}_k = -10^6$ and $\overline{\phi}_k = 10^6$ at all unknowns k which are not related to Γ_C. Similar to other examples of this chapter, the material parameters were given by $E = 300[mPa]$ and $\nu = 0.1$, i.e., material parameters for soft materials. Here, the stop criterion was chosen $\varepsilon = 1e-4$.

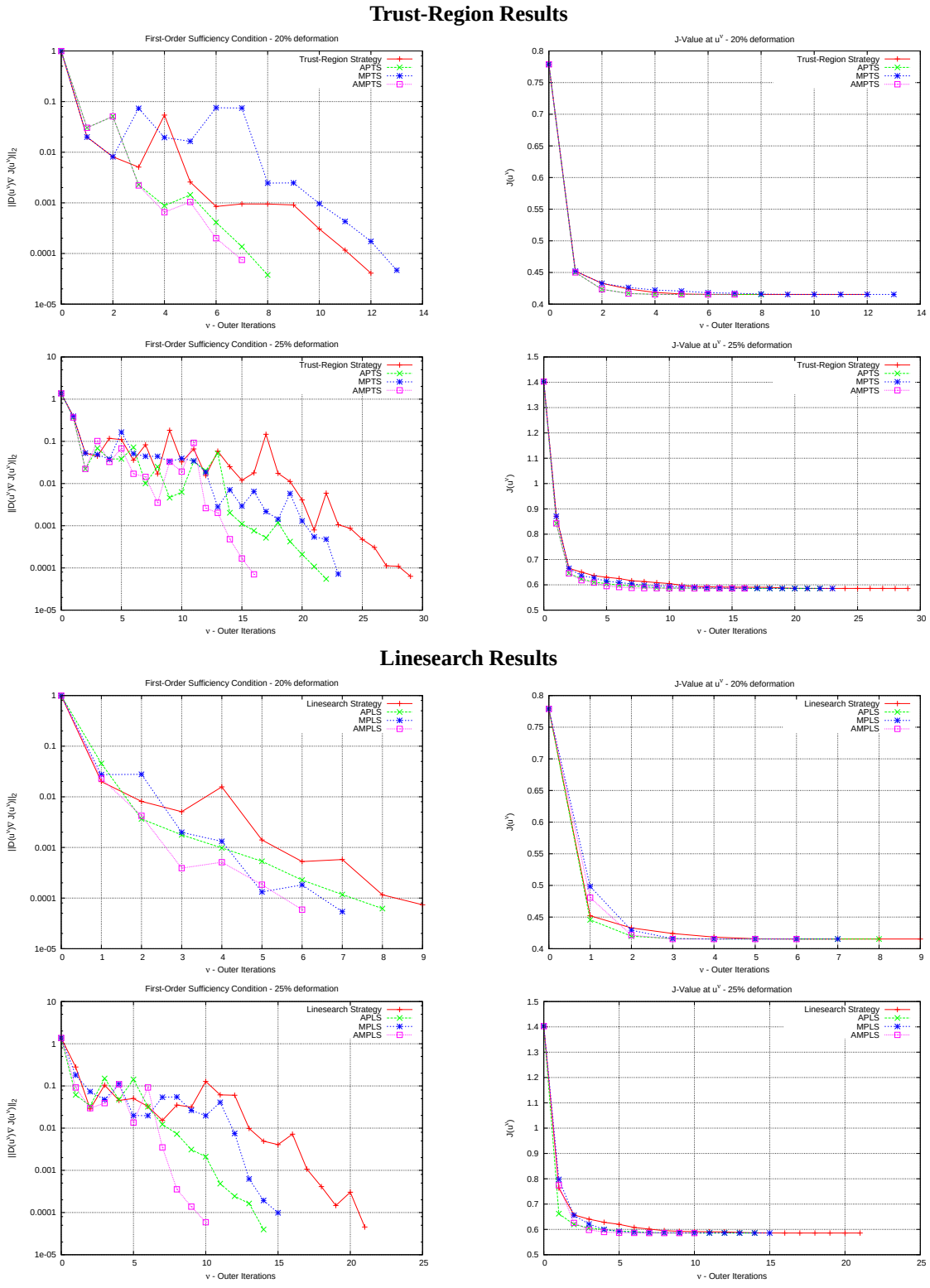

Figure 5.15: **Contact with a Small Obstacle**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.6 with a traditional and preconditioned strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies.

	Example	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	10%	13	5,200	0	1.0
APTS		8	3,200	800	0.83
MPTS		12	4,800	0	1.06
AMPTS		7	2,800	700	0.83
Trust-Region	20%	29	11,600	0	1.0
APTS		22	8,800	2,200	1.02
MPTS		23	9,200	0	0.91
AMPTS		16	6,400	1,600	0.85
Linesearch	10%	9	3,600	0	1.0
APLS		8	3,200	800	1.2
MPLS		7	2,800	0	0.89
AMPLS		6	2,400	600	1.03
Linesearch	20%	15	6,000	0	1.0
APLS		21	8,400	2,100	1.89
MPLS		14	5,600	0	1.07
AMPLS		10	4,000	1,000	0.76

Table 5.4: **Contact with a Small Obstacle**. Runtime comparisons of the globalization strategies for the respective examples.

	Example	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	5%	151	60,400	0	1.0
APTS		58	23,200	5,800	0.51
MPTS		52	20,800	0	0.39
AMPTS		51	20,400	5,100	0.52
Trust-Region	10%	137	54,800	0	1.0
APTS		112	44,800	11,200	1.10
MPTS		73	29,200	0	0.61
AMPTS		45	18,000	4,500	0.50
Linesearch	5%	$>$ 148 (out of time)	$>$ 59,200	0	1.0
APLS		70	28,000	7,000	$<$ 0.63
MPLS		35	14,000	0	$<$ 0.27
AMPLS		75	30,000	7,500	$<$ 0.78
Linesearch	10%	103	41,200	0	1.0
APLS		78	31,200	7,800	1.02
MPLS		80	32,000	0	0.89
AMPLS		44	17,600	4,400	0.66

Table 5.5: **Obstacle Problem: Simulation of a Can**. Runtime comparisons of the traditional and preconditioned globalization strategies for different loads. Note that bifurcations take place, which yield heavily varying necessary iterations.

5.6.7 Constrained Minimization Problem: Simulation of a Can

Within this example, we compute a constrained boundary value problem. Here, a can-like structure is pressed against an obstacle, as shown in Figure 5.16. Due to the fact, that the employed material parameters, $E = 300[mPa]$ and $\nu = 0.1$, describe a soft material, the applied deformations yield two possible minimizers as indicated in the same figure. In turn, the resulting computation times for the respective minimization strategies vary tremendously.
In fact, we apply at $\Gamma_D = \{(x, y, z) \mid x = 0.50\}$ the following displacements

$$g((x, y, z)^T) = (d, 0, 0)$$

where $d = -0.1, -0.2$. All other boundaries have natural boundary conditions. The contact boundary is given by $\Gamma_C = \{(x, y, z) \mid x = -0.50\}$. Here, similarly to the previous example, the geometry and the obstacle stay initially in contact , i.e., $\underline{\phi}_k = 0$ and $\overline{\phi}_k = 10^6$. At all unknowns which are not related to Γ_C, we choose $\underline{\phi}_k = -10^6$ and $\overline{\phi}_k = 10^6$. In this example, the stop criterion was chosen $\varepsilon = 1e - 5$.

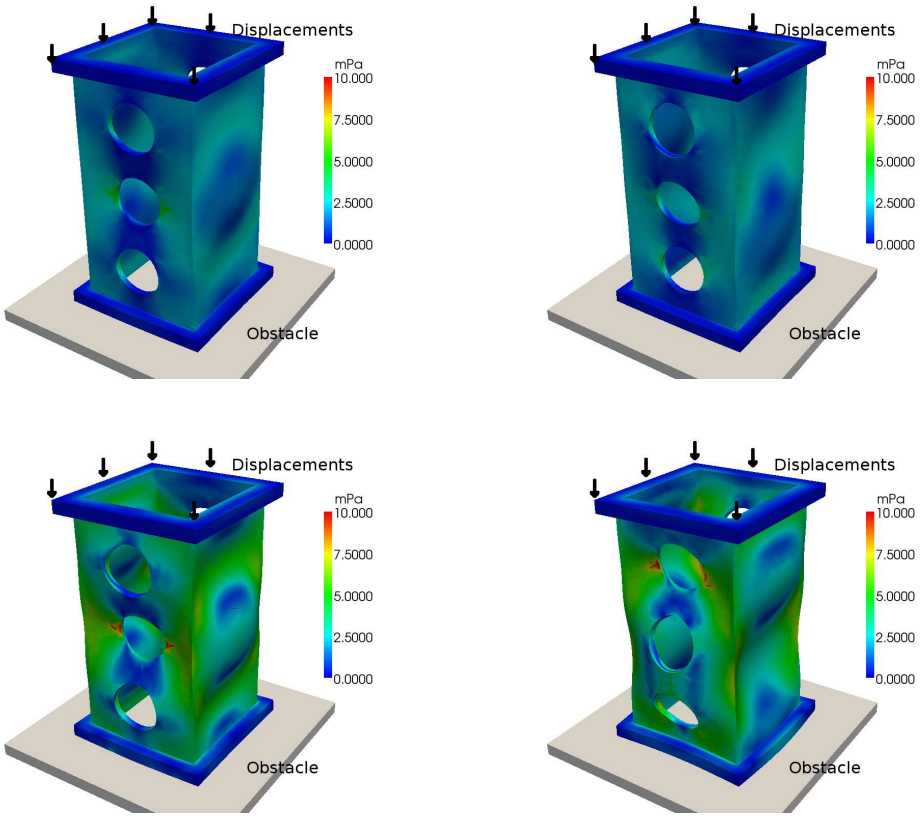

Figure 5.16: **Constrained Minimization Problem: Simulation of a Can**. In this figure different results for the problem of Section 5.6.7 with 323,994 unknowns are presented. The upper images show two possible energy optimal solutions for the obstacle problem with 5% applied deformations. The lower images are the results for 10% displacements.

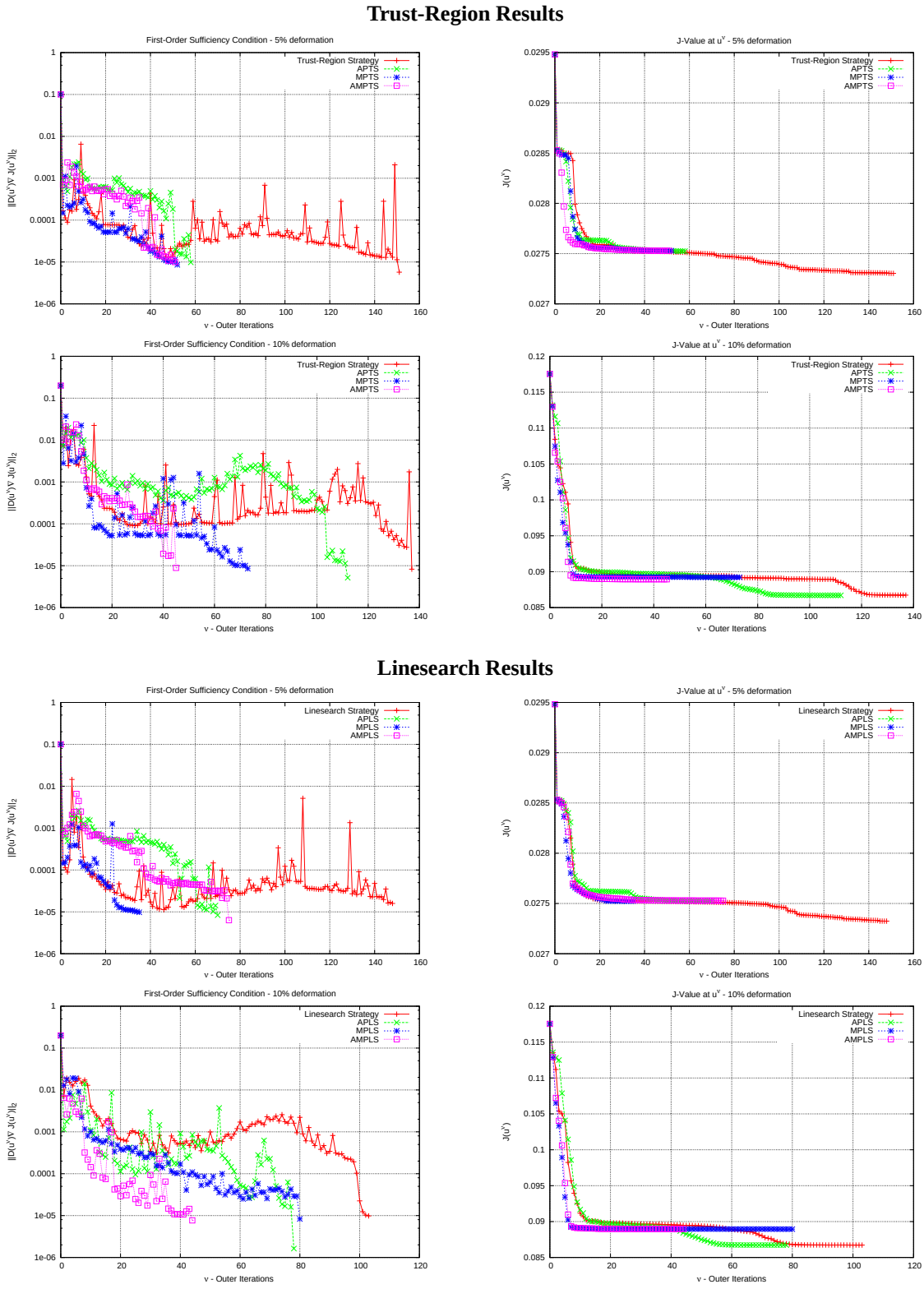

Figure 5.17: **Constrained Minimization Problem: Simulation of a Can**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu) + s^\nu)\|_2$, for the solution of the problem in Section 5.6.7 with the traditional and the preconditioned globalization strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu) + s^\nu)$ for both strategies. Note that bifurcations take place, which yield heavily varying runtimes.

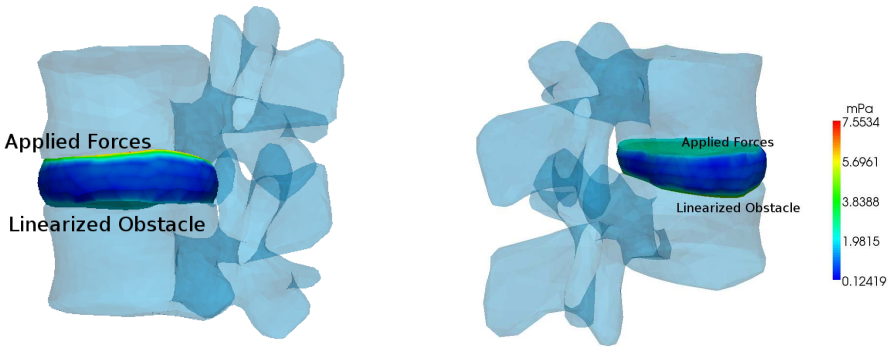

Figure 5.18: **Constrained Minimization Problem: Intervertebral Disk**. This is the annotated result of the computation employing an intervertebral disc geometry with 1,032,000 unknowns. Here, we apply forces at the upper side of the geometry, standing for forces induced from the upper vertebra (as indicated by the upper blue geometry). On the lower side, simple, linearized non-penetration conditions simulate the lower vertebra (as indicated by the lower blue geometry). Note, that for a more correct simulation an elastic multi-body contact must be taken into account, such as proposed in [DGK$^+$08].

5.6.8 Constrained Minimization Problem: Simulation of an Intervertebral Disk

This is the second example with a more realistic context. Here, we employ Ogden's material law with parameters $E = 1500[mPa]$ and $\nu = 0.15$ to compute stresses within an intervertebral disc. In particular, the globalization strategies are employed to compute a traction problem, where we apply forces at the upper side of the geometry, i.e., where the disc stays in contact with the upper vertebra along with obstacle conditions at the lower side, where the lower vertebra would be in contact with the disc. However, note that computing stresses within an intervertebral disc employing Ogden's material law is a poor approximation due to the fluids located within the disc. Often, different material laws are employed to compute stresses within cartilage-like structures, for instance, poro-viscoelastic material laws [WvDvR$^+$04].
At Γ_N we apply the following forces

$$f((x,y,z)^T) = \begin{cases} (0,-10,0) & \text{if } y \approx 0.368 \\ 0 & \text{otherwise} \end{cases}$$

The contact boundary is given by $\Gamma_C \approx \{(x,y,z) \mid y = -0.773\}$. Also in this example, the reference configuration touches the obstacle, which means that $\underline{\phi}_k = 0$ and $\overline{\phi}_k = 10^6$. All other components are chosen $\underline{\phi}_k = -10^6$ and $\overline{\phi}_k = 10^6$.

	Outer it.	cg it. (fine level)	acg it. (fine level)	Time
Trust-Region	4	1,600	0	1.0
APTS	4	1,600	400	1.35
MPTS	1	400	0	0.28
AMPTS	1	400	100	0.38
Linesearch	5	2,000	0	1.0
APLS	5	2,000	500	1.35
MPLS	3	1,200	0	0.69
AMPLS	1	400	100	0.31

Table 5.6: **Obstacle Problem: Constrained Minimization Problem: Intervertebral Disk**. Runtime comparisons of the globalization strategies for the intervertebral disc examples.

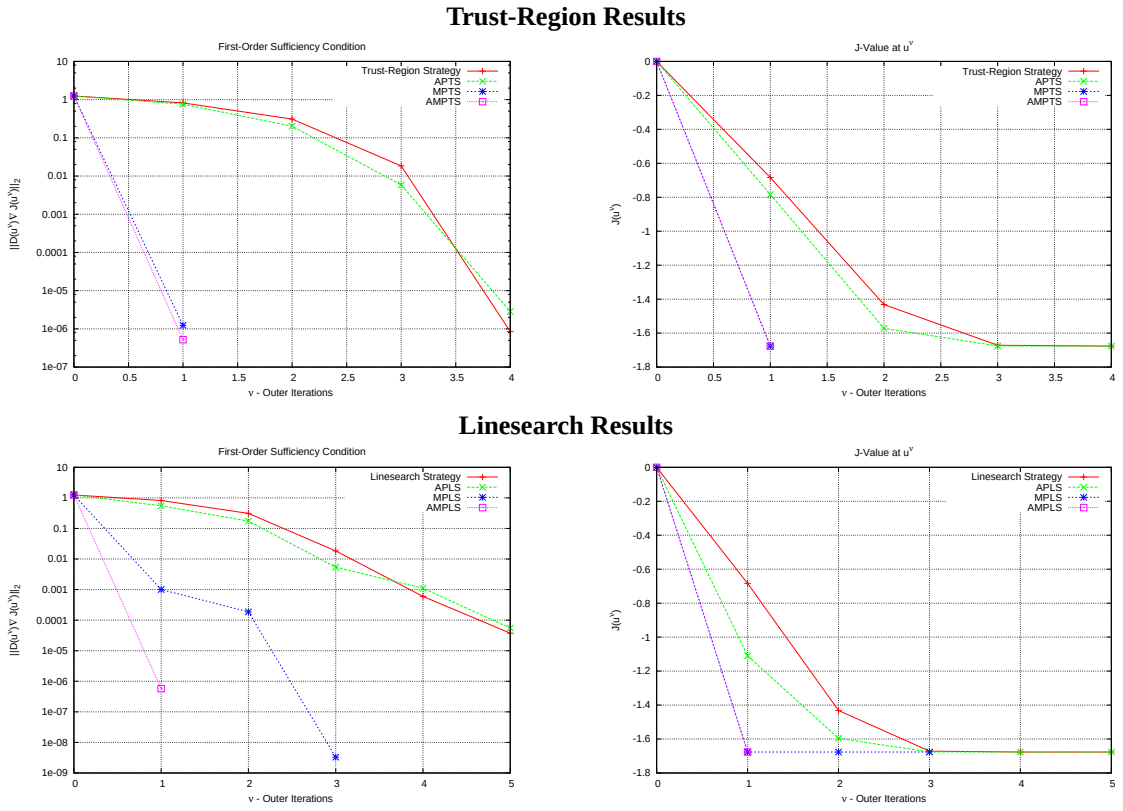

Figure 5.19: **Constrained Minimization Problem: Intervertebral Disk**. The *left diagrams* show the first order sufficient conditions vs. the number of iterations, ν, i.e., $\|\hat{g}(\mathcal{F}(u^\nu)+s^\nu)\|_2$, for the solution of the problem in Section 5.6.8 with a Trust-Region strategy and the preconditioned Trust-Region strategies, respectively. The *right diagrams* show the value of the objective function vs. the number of iterations, ν, i.e., $J(\mathcal{F}(u^\nu)+s^\nu)$ for both strategies.

5.7 Non-Linear Elasto-Dynamic PDEs

In this section we will focus on large deformations in the case of elasto-dynamic contact problems. As we have seen in Section 1.3.1 in each timestep a PDE of the kind (1.3.1b) has a solution $\boldsymbol{u}^i$, if the assumptions of Theorem 1.3.1 are satisfied. Moreover, employing a stored energy function of Ogden type, in particular (5.6.1) yields the solvability of (1.3.1b). Furthermore, as far as in each time step the spatial discretized minimization problem (1.3.4b) and the initial iterate satisfy the assumptions of Lemma 5.6.1 and Lemma 5.6.3 we obtain that the assumptions on the globalization strategies are satisfied and convergence can be ensured. Therefore, as in the previous sections, we will focus on the solution of a fully discretized variant of (1.3.1a). In our case, we employ Finite Elements to derive the minimization problem (1.3.4).

Parameter Choice

As we have pointed out in Section 1.3.1, Newmark's scheme becomes unconditionally stable if $2\beta = \gamma = \frac{1}{2}$. In this case, the time discretization is (partially) implicit and a nonlinear minimization problem must be solved which is carried out employing the AMPLS algorithm, Algorithm 12. The respective constants within this algorithm are chosen as in Section 5.6.2. Even if we employ Rothe's method to discretize the original system of PDEs, we will initially choose a Finite Element discretization, which stays fixed during the computation. The computation of the predictor step (1.3.4a) was carried out employing the projected cg method along with a nonlinear symmetric Gauß-Seidel smoother.

5.7.1 Example: Dynamic Simulation of a Can

In this example, we employ the geometry from Section 5.6.7 as shown in Figure 5.4. Here we are interested in the deformations which occur if this geometry "crashes" against an rigid obstacle, as shown in Figure 5.20. Here, we employed $\Gamma_N = \partial\Omega - \Gamma_C$ where $\Gamma_C = \{(x, y, z) | z = -0.5\}$ with all natural boundary conditions. On the other hand, the initial velocity is given by $(\boldsymbol{u}_0)_k = (0, 0, -0.05)$ for all k, yielding a movement in direction of the obstacle. Initially the displacements are given by $\boldsymbol{u}_0 = 0$ and the gap between geometry and obstacle is slightly larger than zero. At all unknowns which are not related to Γ_C, we choose $\underline{\phi}_k = -10^6$ and $\overline{\phi}_k = 10^6$.
Here, we computed 1,000 timesteps with $\tau = 0.01$. The geometry itself is uniformly refined once giving rise to a nonlinear programming problem, equation (1.3.4b), with approximately $54,000$ unknowns. The employed material parameters are $E = 1000[mPa]$ and $\nu = 0.3$.

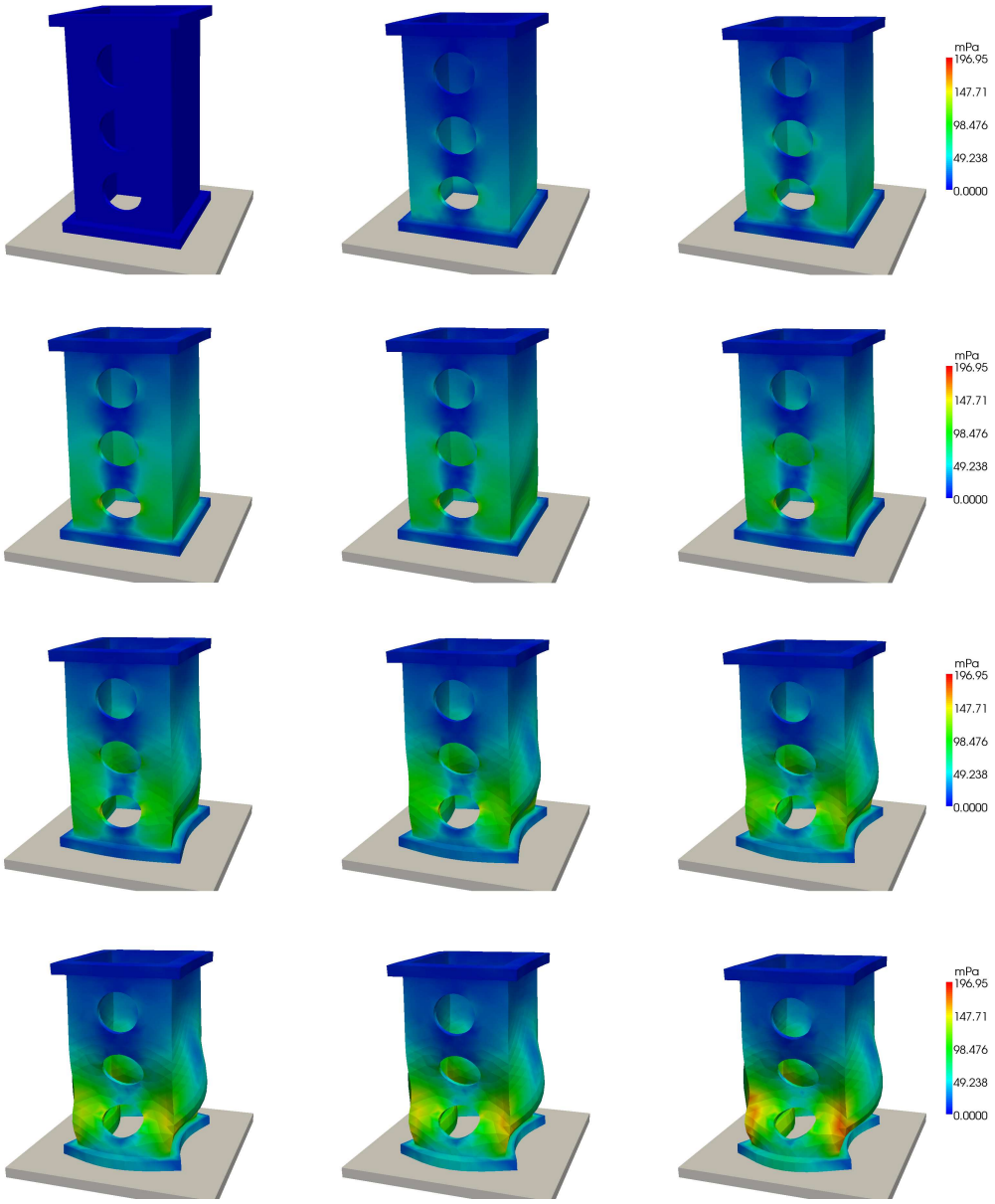

Figure 5.20: **Dynamic Simulation of a Can**. Here, the solution of the problem of Section 5.7.1 is shown. As one can see, the can-like geometry moves in direction of the obstacle as indicated by the grey plane. Soon, the geometry and the obstacle stay in contact and the geometries momentum yields the shown large deformations. The last shown figure is the final configuration in this simulation.

5.7.2 Example: Dynamic Simulation of a Hollow Geometry

In this example, we employ the geometry as shown in Figure 5.4. Similar to the previous example, this geometry moves towards a planar, rigid obstacle. As one can see in Figure 5.20, the geometry is a hollow cube with a circular structure on top. As can be seen in Figure 5.22, the momentum of this circular structure yields that the whole geometry somehow collapses.

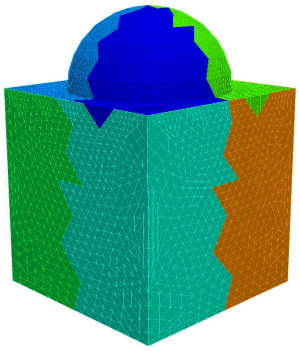

Figure 5.21: The initial geometry of the example from Section 5.7.2. Similar to the other examples of this chapter, we decompose the initial geometry into eight domains.

Here, we employed $\Gamma_N = \partial\Omega - \Gamma_C$ where $\Gamma_C = \{(x, y, z)|z = -0.5\}$. Similar to the previous example, we apply natural boundary conditions on Γ_N. The initial velocity is given by $(\dot{\boldsymbol{u}}_0)_k = (0, 0, -3)$ for all k, yielding a movement in direction of the obstacle. Initially the displacements are given by $\boldsymbol{u}_0 = 0$ and the gap between geometry and obstacle is slightly larger than zero.
Here, we computed 200 timesteps with $\tau = 0.005$. The geometry itself is twice uniformly refined giving rise to a nonlinear programming problem, equation (1.3.4b), with $60,042$ unknowns. The material parameters are $E = 10000[mPa]$ and $\nu = 0.3$.

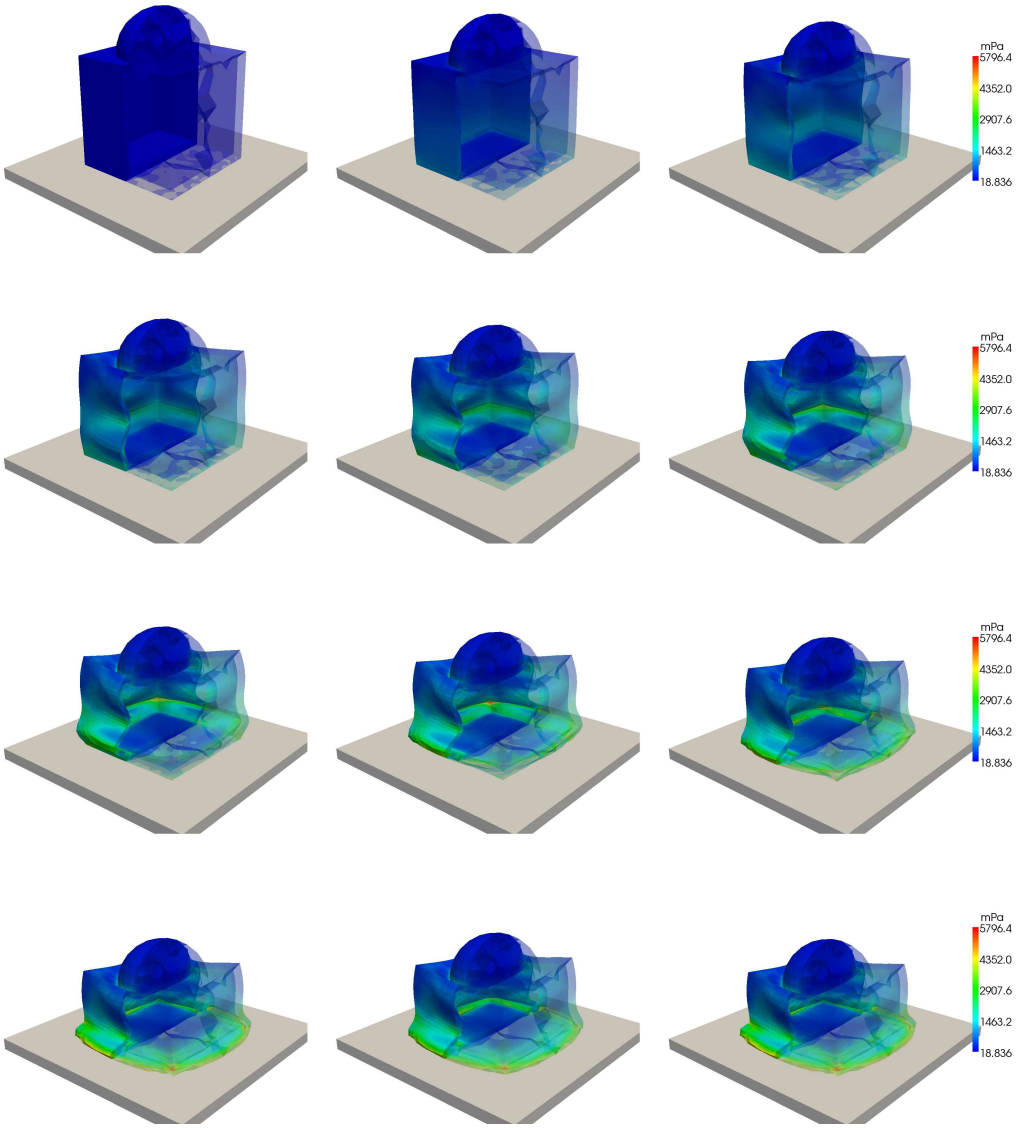

Figure 5.22: **Dynamic Simulation of a Hollow Geometry**. Here, the solution of the problem of Section 5.7.2 is shown. As one can see, a hollow geometry moves in direction of the obstacle as indicated by the grey plane. Soon, the geometry and the obstacle stay in contact and the geometries momentum yields the shown large deformations. The last shown figure is the final configuration in this simulation.

6 Appendix: Implementational Aspects

The development of new algorithms within modern Finite Element software toolboxes, like e.g., DUNE (in combination with UG) [BBD+08] or OBSLIB++ (in combination with UG), has the major advantage that necessary core functionalities are already provided. For instance, the OBSLIB++ toolbox provides

- a grid manager
- linear algebra
- numerical methods to assemble the objective functions, gradients and Hessians
- treatment of the set of admissible solutions $\mathcal{B}$, the obstacles respectively
- parallelization

In this chapter, we will consider implementational aspects of the NLSOLVERLIB toolbox, as well as necessary changes in the OBSLIB++ and in the UG core.

6.1 NLSolverLib

All presented algorithms in this thesis, beginning from the Trust-Region and Linesearch framework up to the combination of the additive and multiplicative frameworks are implemented within the NLSOLVERLIB. This library is a set of numerical procedures called *num-proc*s which may be instantiated during runtime and employed to solve arbitrary minimization problems.

The Respective C++ Classes

Object oriented programming allows for inheriting interfaces and functionalities from already implemented classes in UG. In particular, each instantiated num-proc of a certain UG class can be employed as a black-box. In our implementation, we mostly consider nonlinear solvers inheriting from *NP_NL_SOLVER*. To allow for solving problems like (M), our solvers receive a so-called *ObstacleBase* num-proc, which is able to generate and handle an obstacle on an algebraic level. Moreover, the nonlinear solvers must receive an *NP_NL_ASSEMBLE* num-proc which allows for evaluating J and its derivatives.
In particular, during this dissertation project, the following solver classes were implemented

1. trSolver
2. lineSearchSolver
3. APTS
4. APLS

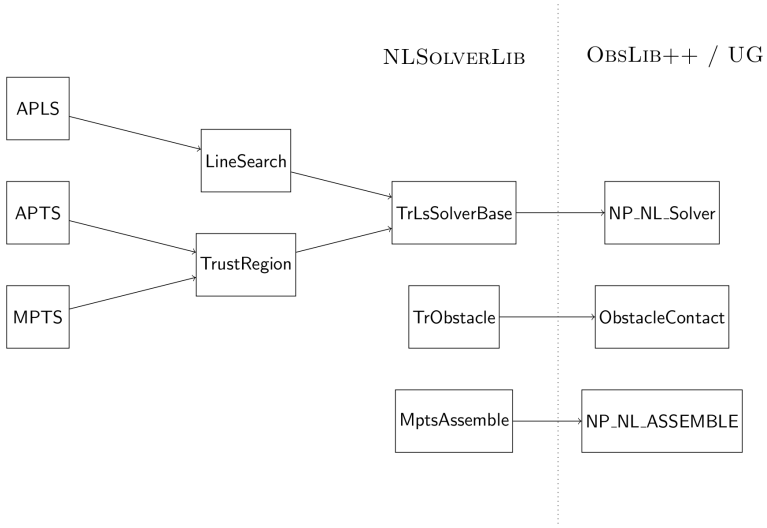

Figure 6.1: The class structure within NLSolverLib. Each of the classes employs the linear algebra provided by UG and ObsLib++. Here, we highlight which class inherits from other classes, and the interface between ObsLib++/UG and the new NLSolverLib.

5. MPTS (realizes MPLS/MPTS within a multigrid framework)

On the other hand, UG allows – independentend from inheritances – for encapsulating num-procs into each other. Therefore, the MPTS solver receives an *NP_NL_SOLVER* num-proc which is employed to solve the local minimization problems. As it turns out, depending on the solver, the MPTS strategy employs the MPLS or the MPTS control routines.

The TrLsSolverBase-Class.
This class provides common methods and fields employed in both, the Trust-Region and Linesearch solver. In particular, this class implements the solution of the quadratic model (2.1.2), and allows for treating complex obstacles employing an SQP approach.
The solution of the quadratic model problem is currently carried out by employing a linear solver. As pointed out before, we employ the projected cg method in combination with a symmetric nonlinear Gauß-Seidel smoother. To guarantee convergence to a first-order critical point we have to ensure that a sufficient decrease takes place. In our implementation, this is realized by computing the Cauchy-point (2.1.10) or (2.2.3) and checking (2.1.9) and (2.2.2b), respectively. If the iteratively computed search direction does not satisfy the respective condition, it is discarded and the respective Cauchy point is employed as search direction.

The TrustRegion-Class. This class realizes along with an *TrObstacle* num-proc Algorithm 1. Some special features, such as the treatment of numerical instabilities in the case of rounding errors have also been added. For instance, if

$$\frac{|J(u) - J(u+s)|}{|J(u)|} \leq \varepsilon$$

holds for $\varepsilon = 10^{-12}$, we cannot trust the decrease ratio (2.1.3). In this case, we have to trust the correction and simply add it. More information on this topic and different strategies can be found in Chapter 10.6 of the monograph [CGT00].

The LineSearch-Class. This class implements the Linesearch algorithm, Algorithm 3. In contrast to just treating the Armijo condition (2.2.5), this algorithm is also able to treat the Armijo conditions for subdomains, i.e., (4.2.4) and (5.3.4). Finally, also the Linesearch algorithm may suffer from rounding errors, which is similarly treated as in the Trust-Region class.

The APTS/APLS-Classes. Here, we implemented certain variants of Algorithm 5 and Algorithm 7. As a matter of fact, the APTS and APLS implementations themselves do not depend on the particular implementation of the domain decomposition but as pointed out in Chapter 5.5, we aim at a real parallel speed-up by decomposing the domain in non–overlapping subdomains as introduced in Section 3.1.6. Therefore, we slightly altered the UG load balancing command such that not only all necessary *master* elements are transferred from one processor to another but also all *ghost* elements which have a node in common with a master element. Therefore, we have a theoretical overlap of one element such that quadrature over all basis functions of a master element is possible without communicating in parallel.

In the preconditioning step $\mathcal{F}_A$, the APTS/APLS solvers directly employ the respective inherited Trust-Region or Linesearch solvers as a nonlinear solver. Since no parallel communication is allowed to take place during the solution process, the nonlinear solver itself, the employed linear solver, the assembler and the obstacle are not allowed to employ parallel communication. In turn, we have an asynchronous solution phase, as described in Algorithm 5 and in Algorithm 7.

After the asynchronous solution process, the APTS and APLS method employ a given parallel nonlinear solver num-proc to compute s^ν from (3.1.6).

The MPTS-Class. This class implements the multiplicative frameworks from Chapter 5. This num-proc switches from Trust-Region to Linesearch behavior depending on the class of the given coarse level NP_NL_SOLVER.

The TrObstacle-Class. This class extends the functionality of the ObstacleContact class by adding an additional constraint, the Trust-Region radius. Moreover, the *TrObstacle*-Class updates and handles the Trust-Region and Linesearch step-length constraints.

The MPTSAssemble-Class. This assembler class aims at the computation and management of the objective function, gradient and Hessian. Due to the formulation of the subset objective functions (3.2.1) and (3.3.1), it receives an NP_FE_ASSEMBLE num-proc, which is able to compute J_k^ν and its derivatives. Therefore, the MPTSAssembler just has to have the ability to compute and add the linear correction term to the objective function and gradient. Moreover it incorporates caching strategies to prevent avoidable recomputations of gradients and Hessians. In fact, the Hessian is recomputed only if a certain threshold is exceeded. Following Taylor's theorem we may estimate the quality of the recently computed Hessian by the following expansion.

$$\nabla H_k^\nu(u) - \nabla H_k^\nu(u+s) - \nabla^2 H_k^\nu(u)s = \nabla^2 H_k^\nu(u+\tau s)s - \nabla^2 H_k^\nu(u)s$$

where $u \in \mathcal{D}_k$ is the iterate, when the Hessian was the last time reassembled, $\tau \in (0,1)$ and $s \in \mathcal{D}_k$ the difference between the most recent iterate and u. Therefore, we reassemble the Hessian if

$$\frac{1}{\|s\|_2}\|\nabla H_k^\nu(u) - \nabla H_k^\nu(u+s) - \nabla^2 H_k^\nu(u)s\|_2 \geq \eta_R \tag{6.1.1}$$

with $\eta_R > 0$. In our examples, we employed $\eta_R = 0.1$. Otherwise we return the more or less up to date Hessian $\nabla^2 H_k^\nu(u)$.
Unfortunately, the original UG implementation was not designed for the evaluation of the objective function and by now only the Finite Element based assemblers allow for evaluating the respective objective functions J_k^ν.

6.2 Asynchronous Linear Solvers

In addition to the already existing (parallelized) linear solvers and preconditioners in OBSLIB++, an asynchronous projected cg-method (acgpl), an asynchronous linear solver (als) and asynchronous symmetric and classical nonlinear Gauß-Seidel variants were implemented. In contrast to the original solvers, the asynchronous versions als and acgpl do not employ parallel communication at all. Therefore, acgpl and als are suitable as a linear solver within the APLS and APTS methods to solve the arising quadratic minimization problems – as far as these incorporate a symmetric positive definite Hessian.

6.3 IOLib

To import CAD-based geometries, an OBSLIB++ plugin has been developed to import EXODUS-II geometries. In combination with the EXODUS-II PARAMETER FORMAT [GK08a] this allows for easily defining the computational domain Ω and the boundary values.
More important is the developed export library ANIMATIONSUITE, which is designed for exporting geometry data and numerical results in EXODUS-II format. Even if UG itself has a visualization unit, since Revision 265 the ANIMATIONSUITE allows to export computed results as CAD data. In turn, one can visualize the computed results in professional toolkits like, e.g., PARAVIEW [Tea09]. Besides the fact that EXODUS-II can be employed to store nodal and element values, it also enables us to export time dependent nodal data frame-wise. Therefore, in the initialization process of ANIMATIONSUITE, we just declare the respective vectors, whose values should be exported. Each time the writeFrame command is called, the current values of all registered vectors are written into a temporary file. With the call of writeMesh the geometry information and all data in the temp file is then written into an EXODUS-II file.
ANIMATIONSUITE also enables us to handle geometries which have been exported at the end of an parallel computation process. It turns out that each domain Ω_i on the i-th processor is exported separately. Therefore, in order to obtain one result, the C++ - mergeMesh method was implemented. Here, on one processor the exported meshes and data are merged yielding exactly one file.

6.4 InterpreterLib

The particular treatment of algebraic expression within UG-scripts was not implemented in UG until OBSLIB++ revision 143. This made the statement of simplest expressions like, $A \cdot u - f$ complicated and barely readable. Therefore, we started to implement a special parser called SYNTAXPARSER which allows for creating a syntax tree whose recursive evaluation yields the sought-after evaluation of the algebraic expression. In particular, the evaluation of an expression splits into two steps:

1. Parsing and creating the syntax tree

Figure 6.2: **Application of** INTERPRETLIB **for Boundary Values**. Here, we describe displacements by means of the coordinates at the respective quadrature points, yielding a rotation of the upper plane of about $45°$. The left image shows the result for linear elasticity, the right image for an Ogden material, as employed in Chapter 5.5.

2. evaluating the syntax tree by successively evaluating intermediate results

Each of the respective nodes of the syntax tree may be a binary operator or a variable of type integer, double, string, vector and matrix. Today, all popular unary and binary operators, such as multiplication, scalar product, modulo, comparison operators, pointwise multiplication for vectors, string-concatenation, etc., are implemented within a *weakly typed* framework. Finally, also boundary values can now be stated algebraically allowing for boundary value problems as shown in Figure 6.2.

Index

$B_i = B(u_i) \approx \nabla^2 J(u_i)$ – symmetric approximation to the Hessian, 17, 102, 129
C, C_i set of indices, 38
C^ν – subsets where corrections were computed, 53, 55
$D_i = D(u_i)$, $D^\nu_{k,i} = D_k(u^\nu_{k,i})$ – scaling matrix, 19
H^ν_k – local objective function, 40, 45
$I^k_{k+1} : \mathcal{D}_{k+1} \to \mathcal{D}_k$ – interpolation operator, 34
$I_k : \mathcal{D}_k \to \mathbb{R}^n$ – interpolation operator, 33
J – global objective function, 5
N – number of subsets, 33
$P^{k+1}_k : \mathcal{D}_k \to \mathcal{D}_{k+1}$ – projection operator, 35
$P_k : \mathbb{R}^n \to \mathcal{D}_k$ – projection operator, 34
$R^{k+1}_k = (I^k_{k+1})^T$ – restriction operator, 34
$R_k = (I_k)^T$ – restriction operator, 34
$X : \mathbb{R}^n \to \mathcal{X}$ – coordinate isomorphism, 33
X^i – coordinate isomorphism, 13
$X_k : \mathcal{D}_k \to \mathcal{X}_k$ – coordinate isomorphism, 33
$\boldsymbol{C} = \nabla\boldsymbol{\varphi}^T \nabla\boldsymbol{\varphi}$ – right Cauchy-Green strain tensor, 7, 98
$\mathcal{D}_k = \mathbb{R}^{n_k}$ – subset, 33
Δ_i – Trust-Region radius (traditional scheme), 18
$\Delta^\nu_{k,i}$ – Trust-Region radius (APTS/MPTS), 52
$\Delta^\nu_{k,i}$ – Trust-Region radius (MPTS), 75
$\boldsymbol{F}$ – volume force densities, 6
Γ_D, Γ_N – Dirichlet/Neumann boundaries, 6
Ω – computational domain, 6
$\hat{\boldsymbol{T}}$ – response function, 7
$\widehat{\boldsymbol{W}}$ – stored energy function, 7
α_i – Linesearch parameter (traditional scheme), 23
$\alpha_{\min}$ – step–length satisfying the Armijo condition, 27, 65
$\alpha^\nu_{k,i}$ – Linesearch parameter (APLS), 62
$\alpha^\nu_{k,i}$ – Linesearch parameter (MPLS), 84
β_{ls} – step–length threshold (APLS), 65
β_{ls} – step–length threshold (MPLS), 86
β_{ls} – step–length threshold (Linesearch), 24
η – threshold for accepting corrections, 18
η_{ls} – threshold for descent (Linesearch), 24
$\boldsymbol{f}$ – surface force density, 6
$\gamma_1, \gamma_2 \in (0, 1)$ – Trust-Region rescaling parameters, 18
$\hat{g}^\nu = \hat{g}^\nu_{G,0}$ – first–order conditions before additive prec. (APTS), 55
$\hat{g}_i = D(u_i)\nabla J(u_i)$ – first–order conditions (traditional scheme), 19
$\hat{g}^\nu_{k,i} = D^\nu_{k,i} \nabla H^\nu_k(u^\nu_{k,i})$ – first–order conditions (subset/global context), 55, 65, 73
$\kappa_g \in (0, 1)$ – threshold for coarse level gradient, 55, 73
λ_i – nodal basis function, 13
$\mathcal{A}$ – additive recombination operator, 41, 53
$\mathcal{A}_k$ – multiplicative recombination operator, 47, 75
$\mathcal{B}$ – set of admissible solutions, 5
$\mathcal{B}_k(u^\nu)$ – admissible solutions, 43
$\mathcal{B}_k(u_{k-1})$ - set of admissible solutions, 49
$\mathcal{F}_M$ – multiplicative update operator, 46
$\mathcal{F}^{(j)}_M$ – nonlinear update operator (MPTS), 75
$\mathcal{F}_k$ – local update operator, 41, 46
$\mathcal{X}$ – original solution space, 33
$\mathcal{X}, \mathcal{X}^i$ – Finite Element space, 12
$\mathcal{X}_k \subset \mathcal{X}$ – local version of the original solution space, 33
ν – index of current outer iteration, 40
$\overline{\phi}, \underline{\phi}$ – obstacle vectors, 5
$\psi_i(s) = \langle g_i, s\rangle + \frac{1}{2}\langle s, B_i s\rangle$ – quadratic model, 17, 25
ρ – mass density, 6
ρ^ν – decrease ratio for additively computed corrections, 53
ρ_A – threshold Armijo condition, 25, 62, 84
ρ_R – descent threshold (APLS), 62

ρ_R – descent threshold (MPLS), 84
ρ_i – decrease ratio (traditional scheme), 17
ρ_{AP} – descent threshold (APLS/MPLS), 64
ρ_{MP} – descent threshold (APLS/MPLS), 85
$\rho^\nu_{k,i}$ – local decrease ratio (APTS), 54
ρ^ν_{k,m_k} – decrease ratio for multiplicatively computed corrections (MPTS), 74
τ – reduction factor in the backtracking algorithm, 25
τ – time step parameter in Rothe's method, 10
$\boldsymbol{u} = \boldsymbol{\varphi} - \mathrm{Id}$ – displacements, 6
$\boldsymbol{u}_h$ – discretized displacements, 12
$\boldsymbol{\varphi}$ – deformation of the ref. configuration, 6
$g_i = \nabla J(u_i)$ – gradient at u_i, 17
$g^\nu_{k,i} = \nabla H^\nu_k(u^\nu_{k,i})$ – gradient, 62
m – maximal number of subset iterations, 73
m_k – number of subset iterations, 73, 88
s_i – correction vector, 17
$s_{k,i}$ – subset correction (APTS/MPTS), 52, 75
$s_{k,i}$ – subset search direction (APLS/MPLS), 62
$s_{k,i}$ – subset search–direction (APLS/MPLS), 84
s_{k,m_k} – multiplicatively computed correction, 73, 87
u^ν – global iterate, 40
$u^\nu_{k+1,m_{k+1,f}}$ – the final iterate on $\mathcal{D}_{k+1}$, 88
$u^\nu_{k,0}$ – initial iterate (additive), 40
$u^\nu_{k,0}$ – initial iterate (multiplicative), 45
$u^\nu_{k,i}$ – subset iterate, 52, 62, 73, 84

additive preconditioning, 39
additive subset obstacles, 43
advance conditions (APTS/APLS), 54
advance conditions (MPTS/MPLS), 73
AMPLS – Combined Nonlinearly Preconditioned Linesearch Methods, 92
AMPLS algorithm, 93
AMPTS – Combined Nonlinearly Preconditioned Trust-Region Methods, 81
AMPTS algorithm, 82
APLS – Nonlinear Additively Preconditioned Linesearch Methods, 60
APLS algorithm, 66
APTS – Nonlinear Additively Preconditioned Trust–Region Methods, 51
APTS algorithm, 53
Armijo condition, 15, 25
Armijo condition (APLS), 62
Armijo condition (MPLS), 84
ASPIN, 3, 29, 39, 45
assumption Trust-Region method, 16
assumptions Linesearch , 24
assumptions APLS, 61
assumptions APTS, 51
assumptions MPLS, 83
assumptions MPTS, 72

backtracking algorithm, 25
barrier function, 98, 100, 103
bisection algorithm, 85
boundedness α (Linesearch), 26
boundedness α (APLS), 65
boundedness α (MPLS), 91

Cauchy Point, 102
Cauchy point, 15, 19, 128
coarse grid, 71
coercive, 97, 98
coercivity of a stored energy function, 8
compressible Mooney-Rivlin material, 98
computation of Trust-Region corrections, 17
conservation of impulse, 6
contact stabilization, 10, 123

decrease ratio (APTS), 53
descent condition - I (APLS), 62
descent condition -II (APLS), 64
descent direction (Linesearch), 24
descent direction (APLS), 66
descent direction (MPLS), 89
domain decomposition, 71

exact update strategy, 31

FAS, 2, 29, 45
filter–based Linesearch strategy, 29
Finite Differences, 12
Finite Elements, 12, 33, 35, 48, 98, 99, 104
first–order conditions, 5, 19
forget-me-not approach, 40

Gauß-Seidel, 47, 72

Hilbert space, 35
Hooke's law, 101

horizontal decomposition, 33

indicator function, 100
initial iterate (additive), 34, 40
initial iterate (multiplicative), 45
initial subset iterate, 33
interpolation operator, 33, 38, 39, 43, 48, 53, 62, 74, 75, 86, 88

Krylov-Schwarz, 29

Lamé constants, 98
left preconditioning, 30
linear coupling term, 40, 45
linear multigrid methods, 35
linearized Green–St. Venant strain tensor, 101
linearized update strategy, 31
Linesearch algorithm, 27
Linesearch method, 24
local Trust-Region constraint, 54
local function J_k^ν, 40, 103
local objective function (additive), 40
local objective function (multiplicative), 45
local update operator $\mathcal{F}_k$, 52, 61, 84, 102
lumped mass matrix, 37

mass matrix, 14, 37
Meshfree Methods, 12
method of lines, 9
MG/Opt, 2, 29, 45, 71
minimization problem (M), 4
MLS, 2, 72, 84
Modified Linesearch Algorithm (APLS), 64
modified Linesearch algorithm (MPLS), 86
MPLS – Nonlinear Multiplicatively Preconditioned Linesearch Methods, 83
MPLS algorithm, 87
MPLS step–length criterion, 86
MPTS – Nonlinear Multiplicatively Preconditioned Trust-Region Methods, 72
MPTS algorithm, 74
multigrid method, 37
multiplicative descent condition I, 84
multiplicative descent condition II, 85
multiplicative subset obstacles, 49
multiplicatively computed correction, 47

Newmark scheme, 10, 123
Newton's method, 1
non-overlapping domain decomposition, 38, 55, 58
nonlinear multigrids, 72
nonlinear recombination operator $\mathcal{A}$, 41, 46, 53, 62, 75, 88
nonlinear Schwarz method, 39
nonlinear symmetric Gauß–Seidel method, 102, 123
nonlinear update operator, 30, 45
nonlinear update operator $\mathcal{F}_A$ (APLS), 62
nonlinear update operator $\mathcal{F}_A$ (APTS), 53
nonlinear update operator (concept), 32, 41, 46
nonlinear update operator (MPLS), 88
normal equation, 35

Ogden material, 97
orthogonal projection, 34
overlapping domain decomposition, 38, 55, 58

parallel gradient distribution, 3, 39, 60
parallel variable distribution, 3, 29, 39, 40, 42, 51
ParaView, 101
polyconvex, 97, 98
polyconvexity of a stored energy function, 8
post-smoothing, 32
projection operator, 33, 34, 37–39, 48

recursively computed correction, 47
residual δg_k^ν (additive), 40
residual δg_k^ν (multiplicative), 45
restriction operator, 35, 38
right preconditioning, 32
RMTR, 2, 71
Rothe's method, 9

second–order coupling (additive), 41
second–order coupling (multiplicative), 46
second–order critical points, 23, 27, 69, 95
sequential-quadratic-programming, 29
St. Venant-Kirchhoff material law, 97, 98
step–length condition Linesearch , 24
subset correction, 33
successful correction, 18
sufficient decrease, 81, 82
sufficient decrease (APTS), 55
sufficient decrease (MPLS), 76

sufficient decrease condition, 19

transfer operator, 33
Trust–Region algorithm, 18
Trust–Region method (traditional), 16
Trust–Region update, 75
Trust–Region update (APTS), 54

update Trust-Region radius, 18
update operator, 48

V-Cycle, 47, 48
vertical and horizontal decomposition, 81
vertical decomposition, 33
von-Mises stress, 100

W-Cycle, 47
Wavelets, 12, 71
Wolfe condition, 25, 84, 86

Bibliography

[AMPS08] J. Arnal, V. Migallòn, J. Penadès, and D. B. Szyld. Newton additive and multiplicative Schwarz iterative methods. *IMA Journal of Numerical Analysis*, 28(3):143–161, 2008.

[Arm66] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pac. J. Math.*, 16:1–3, 1966.

[Bal77] J. M. Ball. Convexity conditions and existence theorems in nonlinear elasticity. *Arch. Ration. Mech. Anal.*, 63:337–403, 1977.

[Bas96] P. Bastian. *Parallele adaptive Mehrgitterverfahren*. Teubner Skripten zur Numerik. Teubner-Verlag, 1996.

[BBD+08] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klfkorn, M. Ohlberger, and O. Sander. A generic grid interface for parallel and adaptive scientific computing. part i: Abstract framework. *Computing*, 82(2-3):103–119, 2008.

[BBJ+97] P. Bastian, K. Birken, K. Johannsen, S.Lang, N. Neuß, H. Rentz-Reichert, and C.Wieners. UG – a flexible software toolbox for solving partial differential equations. *Computing and Visualization in Science*, 1:27–40, 1997.

[Bra81] A. Brandt. Guide in multigrid development. In U. Trottenberg W. Hackbusch, editor, *Multigrid Methods*, volume 960 of *Lect. Notes Math.*, pages 220–312, 1981.

[Bra07] D. Braess. *Finite elements. Theory, fast solvers and applications in solid mechanics. Translated from German by Larry L. Schumaker.* Cambridge: Cambridge University Press. 17, 2007.

[Bro70] C.G. Broyden. The convergence of a class of double-rank minimization algorithms. I: General considerations. *J. Inst. Math. Appl.*, 6:76–90, 1970.

[CGT00] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[Cia88] P. G. Ciarlét. Mathematical elasticity, volume I: Three-dimensional elasticity. *Studies in Mathematics and its Applications*, 20(186):715–716, 1988.

[CK02] X-C. Cai and D. E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.*, 24(1):183–200, 2002.

[CKM02] X.-C. Cai, D. E. Keyes, and L. Marcinkowski. Nonlinear additive Schwarz preconditioners and application in computational fluid dynamics. *Int. J. Numer. Methods Fluids*, 40(12):1463–1470, 2002.

[CKY02] X.-C. Cai, D. E. Keyes, and D. Young. A nonlinear additive Schwarz preconditioned inexact Newton method for schocked duct flows. Debit, N. (ed.) et al., Domain decomposition methods in science and engineering. Papers of the thirteenth international conference on domain decomposition methods, Lyon, France, October 9–12, 2000. Barcelona: International Center for Numerical Methods in Engineering (CIMNE). Theory Eng. Appl. Comput. Methods, 345-352 (2002)., 2002.

[CL94] T. F. Coleman and Y. Li. On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, 67(1-3), 1994.

[CL96] T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM J. Optim.*, 6:418–445, May 1996.

[Cur44] H. B. Curry. The method of steepest descent for non-linear minimization problems. *Q. Appl. Math.*, 2:258–261, 1944.

[Dah97] W. Dahmen. Wavelet and multiscale methods for operator equations. Iserles, A. (ed.), Acta Numerica Vol. 6, Cambridge: Cambridge University Press. 55-228 (1997)., 1997.

[DGK+08] T. Dickopf, C. Groß, R. Krause, C. Mohr, and J. Steiner. Efficient simulation techniques for heterogeneous models in biomechanics. INS preprint 0806, Institute for Numerical Simulation, University of Bonn, July 2008. Published in Proceedings of the 8th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering.

[DH08] P. Deuflhard and A. Hohmann. *Numerical mathematics. 1: An algorithmically oriented introduction. (Numerische Mathematik 1. Eine algorithmisch orientierte Einführung.) 4th revised and extended ed.* de Gruyter Lehrbuch. Berlin: de Gruyter. xii, 2008.

[DKE08] P. Deuflhard, R. Krause, and S. Ertel. A contact–stabilized newmark method for dynamical contact problems. *International Journal for Numerical Methods in Engineering*, 73(9):1274 – 1290, 2008. Available as INS Preprint No 0602.

[DS83] J. E. Dennis and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.* Prentice-Hall Series in Computational Mathematics. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. XIII, 1983.

[DS98] J. E. Dennis, , and T. Steihaug. A Ferris-Mangasarian technique applied to linear least squares problems. Technical report, Rice University, 1998.

[EGK08] C. Eck, H. Garcke, and P. Knabner. *Mathematical modelling. (Mathematische Modellierung.).* Springer-Lehrbuch. Berlin: Springer. xiv, 2008.

[Fle70] R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.

[FM90] A. V. Fiacco and G. P. McCormick. *Nonlinear programming: sequential unconstrained minimization techniques. Unabridged, corrected republication.* Classics in

Applied Mathematics, 4. Philadelphia, PA: SIAM, Society for Industrial and Applied Mathematics. xvi, 1990.

[FM94] M. C. Ferris and O. L. Mangasarian. Parallel variable distribution. *SIAM J. Optim.*, 4(4):815–832, 1994.

[FR99] A. Frommer and R. A. Renaut. Parallel space decomposition for minimization of nonlinear functionals. In T. Yang, editor, *Parallel numerical computation with applications. Proceedings of the workshop on Frontiers of parallel numerical computations and applications, organized in the IEEE 7th symposium on the Frontiers on massively parallel computers (Frontiers '99) at Annapolis, MD, USA, February 20-25, 1999. Boston: Kluwer Academic Publishers. Kluwer Int. Ser. Eng. Comput. Sci. 515, 53-61*, 1999.

[GK08a] C. Groß and R. Krause. Import of geometries and extended informations into obslib++ using the exodus ii and exodus parameter file formats. Technical Report 712, Institute for Numerical Simulation, University of Bonn, Germany, January 2008.

[GK08b] C. Groß and R. Krause. On the convergence of recursive trust–region methods for multiscale non-linear optimization and applications to non-linear mechanics. INS preprint 0710, Institute for Numerical Simulation, University of Bonn, 03 2008. To appear in SIAM J. Numer. Anal.

[GK08c] C. Groß and R. Krause. A recursive trust–region method for non-convex constrained minimization. INS preprint 0713, Institute for Numerical Simulation, University of Bonn, April 2008.

[GM90] E. Gelman and J. Mandel. On multilevel iterative methods for optimization problems. *Math. Program., Ser. B*, 48(1):1–17, 1990.

[GMS+09] S. Gratton, M. Mouffe, A. Sartenaer, Ph. L. Toint, and D. Tomanos. Numerical experience with a recursive trust-region method for multilevel nonlinear optimization. *Optimization Methods and Software*, to appear, 2009.

[GMTWM08] S. Gratton, M. Mouffe, Ph. L. Toint, and M. Weber-Mendonca. A recursive trust-region method in infinity norm for bound-constrained nonlinear optimization. *IMA Journal of Numerical Analysis*, 28(4):827–861, 2008.

[Gol62] A. A. Goldstein. Cauchy's method of minimization. *Numer. Math.*, 4:146–150, 1962.

[Gol70] D. Goldfarb. A family of variable metric updates derived by variational means. *Math. Comp.*, 24:23–26, 1970.

[GQT66] S. M. Goldfeld, R. E. Quandt, and H.F. Trotter. Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551, 1966.

[GS61] R. E. Griffith and R. A. Stewart. A nonlinear programming technique for the optimization of continuous processing systems. *Manage. Sci.*, 7:379–392, 1961.

[GST08] S. Gratton, A. Sartenaer, and P. L. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM Journal on Optimization*, 19(1):414–444, 2008.

[HXC05a] F-N. Hwang and X.C.-Cai. A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations. *J. Comput. Phys.*, 204(2):666–691, 2005.

[HXC05b] F-N. Hwang and X.C.-Cai. A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations. *J. Comput. Phys.*, 204(2):666–691, 2005.

[JS04] F. Jarre and J. Stoer. *Optimization. (Optimierung.)*. Berlin: Springer. xii, 475 S. EUR 29.95/net; sFr 48.00 , 2004.

[Kra01] R. Krause. *Monotone Multigrid Methods for Signorini's Problem with Friction*. PhD thesis, Freie Universität Berlin, 2001.

[Kra07a] R. Krause. On the multiscale solution of constrained minimization problems. *Proceedings of the 17th International Conference on Domain Decomposition Methods, Graz, Austria*, 2007.

[Kra07b] R. Krause. A parallel decomposition approach to non-smooth minimization problems - concepts and implementation. Technical Report 709, Institute for Numerical Simulation, University of Bonn, Germany, 2007.

[Lev44] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Q. Appl. Math.*, 2:164–168, 1944.

[Lev02] R. J. Leveque. *Finite volume methods for hyperbolic problems.* Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press. xix, 2002.

[Liu03] G.-R. Liu. *Mesh free methods. Moving beyond the finite element method.* Boca Raton, FL: CRC Press. xviii, 2003.

[LN05a] R. M. Lewis and S. G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.*, 26(6):1811–1837, 2005.

[LN05b] R. M. Lewis and S.G. Nash. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. Sci. Comput.*, 26(6):1811–1837, 2005.

[LN06] R. M. Lewis and S.G. Nash. Factors affecting the performance of optimization-based multigrid methods. Hager, William W. (ed.) et al., Multiscale optimization methods and applications. Selected papers based on the presentation at the conference, Gainesville, FL, USA, February 26–28, 2004. New York, NY: Springer. Nonconvex Optimization and Its Applications 82, 151-172 (2006)., 2006.

[Man84] J. Mandel. A multilevel iterative method for symmetric, positive definite linear complementarity problems. *Appl. Math. Optimization*, 11:77–95, 1984.

[Man95] O.L. Mangasarian. Parallel gradient distribution in unconstrained optimization. *SIAM J. Control Optimization*, 33(6):1916–1925, 1995.

[Mar63] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.*, 11:431–441, 1963.

[Mor60] D.D. Morrison. Methods for nonlinear least squares problems and convergence proofs. *Proceedings of the Seminar on Tracking Programs and Orbit Determination*, pages 1–9, 1960.

[Mor78] J. J. More. The Levenberg-Marquardt algorithm: implementation and theory. Numer. Anal., Proc. bienn. Conf., Dundee 1977, Lect. Notes Math. 630, 105-116 (1978)., 1978.

[Nas00] S. G. Nash. A multigrid approach to discretized optimization problems. *Journal of Optimization Methods and Software*, 14:99–116, 2000.

[New59] N.M. Newmark. A method of computation for structural dynamics. *Journal of Engineering Mechanics Division*, 8, 1959.

[NW06] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006.

[NZ01] U. Nackenhorst and B. Zastrau. Numerische Parameterstudien zur Beanspruchung von Rad und Schiene beim Rollkontakt. *Der Eisenbahningenieur*, 52:48–52, 2001.

[Ogd97] R. W. Ogden. *Non-linear elastic deformation*. Dover Publications; New Edition, 1997.

[Pow70] M. J. D. Powell. A new algorithm for unconstrained optimization. *Nonlinear Programming*, 1970.

[Rao86] A. Raoult. Non-polyconvexity of the stored energy function of a Saint Venant- Kirchhoff material. *Apl. Mat.*, 31:417–419, 1986.

[Reu88a] A. Reusken. Convergence of the multigrid full approximation scheme for a class of elliptic mildly nonlinear boundary value problems. *Numer. Math.*, 52(3):251–277, 1988.

[Reu88b] A. Reusken. Convergence of the multilevel full approximation scheme including the V- cycle. *Numer. Math.*, 53(6):663–686, 1988.

[Sch90] H. A. Schwarz. *Gesammelte mathematische Abhandlungen. 2 Bände.* Berlin. Springer. Bd. I. XI u. 338 S., Bd. II. VII u. 370 S. gr 8° , 1890.

[Sha70] D.F. Shanno. Conditioning of quasi-Newton methods for function minimization. *Math. Comp.*, 24:647–656, 1970.

[Sol97] M. V. Solodov. New inexact parallel variable distribution algorithms. *Computational Optimization and Applications*, 7:165–182, 1997.

[SSB85] G. A. Shultz, R. B. Schnabel, and R. H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985.

[ST92] J.C. Simo and N. Tarnow. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. *Z. Angew. Math. Phys.*, 43(5):757–792, 1992.

[SY94] L. Schoof and V. Yarberry. Exodus II: A finite element data model. Technical report, Sandia National Laboratories, SAND92–2137, Albuquerque, NM., 1994.

[Tea09] Paraview Developer Team. Paraview – Open Source Scientific Visualization, 2009.

[Toi88] Ph. L. Toint. Global convergence of a class of trust-region methods for nonconvex minimization in Hilbert space. *IMA J. Numer. Anal.*, 8(2):231–252, 1988.

[TOP08] http://top500.org, 11 2008.

[TW05] Andrea Toselli and Olof Widlund. *Domain decomposition methods – algorithms and theory.* Springer Series in Computational Mathematics 34. Berlin: Springer. xv, , 2005.

[UUH99] M. Ulbrich, S. Ulbrich, and M. Heinkenschloss. Global convergence of trust-region interior-point algorithms for infinite-dimensional nonconvex minimization subject to pointwise bounds. *SIAM Journal on Control and Optimization*, 37(3):731–764, 1999.

[Vav91] S. A. Vavasis. *Nonlinear optimization. Complexity issues.* International Series of Monographs on Computer Science. 8. New York: Oxford University Press. xii, 1991.

[WB06] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[WG08] Z. Wen and D. Goldfarb. Line search multigrid methods for large-scale non-convex optimization. Technical report, IEOR Columbia University, 2008.

[Win73] D. Winfield. Function minimization by interpolation in a data table. *J. Inst. Math. Appl.*, 12:339–347, 1973.

[Wri08] P. Wriggers. *Nonlinear finite element methods.* Berlin: Springer. xii, 2008.

[WT02] S. J. Wright and M. J. Tenny. A feasible trust-region sequential quadratic programming algorithm. optimization. Technical report, SIAM Journal on Optimization, 2002.

[WvDvR+04] W. Wilson, C.C. van Donkelaar, B. van Rietbergen, K. Ito, and R. Huiskes. Stresses in the local collagen network of articular cartilage: a poroviscoelastic fibril-reinforced finite element study. *J Biomech*, 37(3):357–66, 3 2004.

[YZ01] Y. Ye and S. Zhang. New results on quadratic minimization. *SIAM Journal on Optimization*, 14:245–267, 2001.