Universidade de Lisboa
Faculdade de Ciências
Departamento de Matemática

Theoretical and Numerical Analysis of Optimization Problems with Applications to Continuum Mechanics

Sérgio Paulo Fino de Sousa Lopes

Doutoramento em Matemática

(Análise Numérica e Matemática Computacional)

2012

Universidade de Lisboa
Faculdade de Ciências
Departamento de Matemática

Theoretical and Numerical Analysis of Optimization Problems with Applications to Continuum Mechanics

Sérgio Paulo Fino de Sousa Lopes

Tese orientada pelo Prof. Doutor Cristian Angel Barbarosie especialmente elaborada para a obtenção do grau de doutor em Matemática (Análise Numérica e Matemática Computacional)

2012

Acknowledgements

First and foremost I wish to thank Prof. Cristian Barbarosie for his continued supervision over the past several years, back to the days of my Master's thesis. His support and insight have been invaluable.

I express my deepest gratitude to Prof. Luís Sanchez and all the staff at *Centro de Matemática e Aplicações Fundamentais* for providing the best working conditions possible.

I would also like to thank my colleagues at *Instituto Superior de Engenharia de Lisboa*, namely Bruno and Prof. Canto de Loura, and Prof. Luís Silva for giving me the chance of having a semester off teaching duties (and another one with a reduced teaching schedule).

Many thanks to all my friends at *Complexo Interdisciplinar*: Ricardo and Filipa, Nuno and Hugo (the mathematicians), Paulo, Pedro (the physicist) and Pedro (the mathematician), Nuno and Hugo (the chemists), and Sérgio Cláudio for the technical support.

A big thank you to my sister and to granny *Maria*; and to my parents: without their support over the years I would have never had the opportunity to thank all of the above.

Finally, I wish to thank the *Mathematical and Numerical Methods in Mechanics* research group at *Centro de Matemática e Aplicações Fundamentais* for the financial support granted via *Fundação para a Ciência e a Tecnologia, Financiamento Base 2010 – ISFL/1/209*.

This work was also supported directly by *Fundação para a Ciência e a Tecnologia* through the grant SFRH/BD/44343/2008, which I gratefully acknowledge.

Abstract

This thesis consists of four chapters sharing the underlying theme of optimization, though approached from different perspectives.

In the first chapter, algorithms based solely on gradient information are developed to address constrained problems of the form $\min_{x\in\mathbb{R}^n} f(x)$, subject to $g(x) = 0$, where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$ are smooth functions. A linear rate of (local) convergence is deduced for one of those methods.

The second chapter targets inequality constraints $g(x) \in]-\infty, 0]^m$, $m \in \mathbb{N}$, by combining the algorithms designed in the first chapter with an active-set strategy. Departing from this methodology, a special class of nonsmooth problems is then adressed, namely minimax problems $\min_{x\in\mathbb{R}^n} F(x)$, where $F(x) = \max\{f_1(x), f_2(x), \ldots, f_m(x)\}$, $m \in \mathbb{N}$, or $F(x) = \max_{y\in Y} f(x, y)$, with $Y \subset \mathbb{R}^p$ compact. The functions $f_1, f_2, \ldots, f_m : \mathbb{R}^n \to \mathbb{R}$ and $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ are supposed smooth. No convergence results are stated.

The third chapter revolves around the properties of certain integral functionals defined over sets of measurable matrix functions $A : \Omega \to \mathbb{R}^{n,n}$, $A(x) = A(x)^T$, where $\Omega \subset \mathbb{R}^n$ is a bounded domain:

$$\Phi(A) = \int_\Omega \phi(A(x))\, dx.$$

The integrand ϕ depends only on the eigenvalues (or equivalently, on the invariants) of its matrix argument. Such functionals typically arise in *free material design* frameworks, in the context of structural optimization. Emphasis is placed on lower semicontinuity of Φ with respect to H-convergence (and its relation to the convexity of ϕ), but other properties are investigated, such as subadditivity and positive homogeneity.

The fourth and final chapter introduces a notion of *compliance* for linearly elastic structures occupying a bounded domain $\Omega \subset \mathbb{R}^n$, whose boundary $\partial\Omega$ is split into disjoint parts Γ_D and Γ_N, and governed by the elliptic partial differential equation

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = f & \text{in } \Omega, \\ u = \bar{u} & \text{on } \Gamma_D, \\ E\varepsilon(u)\nu = g & \text{on } \Gamma_N, \end{cases}$$

where either f or g (eventually both) and $\bar{u}$ *are nonzero*. The proposal is supported by numerical evidence.

Keywords

Gradient methods, active-set methods, worst-case optimization, homogenization, free material optimization, H-convergence, lower semicontinuity, minimum compliance design.

Resumo

Esta tese consiste de quatro capítulos que partilham o tema recorrente da otimização, embora abordado de diferentes perspetivas.

A sequência pela qual os capítulos são apresentados não reflete a ordem cronológica do trabalho. Esta tese foi motivada por problemas de otimização estrutural (por exemplo, maximizar a rigidez de uma estrutura elástica sujeita a certos esforços e sob certos constrangimentos). No seguimento dos resultados conseguidos durante o Mestrado, é aprofundado no terceiro capítulo o estudo de um tipo particular de *funcionais custo* que é usado para impor determinadas restrições quando o principal parâmetro a otimizar é o próprio material de que a estrutura é constituída.

Ao longo deste estudo teve que se enfrentar o problema de como definir a *flexibilidade* de uma estrutura em presença de condições de esforço gerais, questão para a qual não existia uma resposta concreta na literatura à data do início dos trabalhos. É este o tema do quarto capítulo.

Estando em causa, como se vê, problemas de otimização com constrangimentos, o passo natural seguinte foi o de desenvolver algoritmos apropriados, e preferencialmente "baratos" do ponto de vista computacional, o que foi feito nos primeiro e segundo capítulos.

Segue-se uma descrição mais pormenorizada.

No primeiro capítulo são concebidos algoritmos para encontrar mínimos locais de uma função continuamente diferenciável $f : \mathbb{R}^n \to \mathbb{R}$, sujeitos a constrangimentos de igualdade $g = 0$ expressos por meio de uma função vetorial $g : \mathbb{R}^n \to \mathbb{R}^m$ $(m < n)$, também ela continuamente diferenciável. Sublinha-se desde já que, regra geral, as iteradas produzidas por estes algoritmos podem violar (largamente até) a condição $g = 0$; elas vão gradualmente aproximando-se do nível zero de g e só assintoticamente satisfazem os constrangimentos.

A abordagem insere-se numa classe de métodos que itera, partindo de um dado inicial $x_0 \in \mathbb{R}^n$, segundo um esquema $x_{k+1} = x_k + \Delta_k$ em que no incremento Δ_k podem ser separados os dois objetivos do problema: minimizar a função f e satisfazer a equação $g = 0$. Isto é feito dando, em cada iterada x_k, um passo de descida (para f) numa direção τ_k tangente em x_k à variedade $g^{-1}(\{g(x_k)\})$, conjuntamente com um passo ν_k de tipo Newton (para g) ortogonal a τ_k; mais exatamente, pode escrever-se $\Delta_k = \eta_k \tau_k + \nu_k$, onde $\eta_k > 0$ designa o comprimento do passo dado na direção tangencial.

São exclusivamente considerados métodos em que a direção τ_k é de "máxima descida" para f, o que no enquadramento estudado traduz-se por tomar a projeção ortogonal de $-\nabla f(x_k)$ sobre o espaço tangente à variedade $g^{-1}(\{g(x_k)\})$ em x_k.

É provada uma taxa linear de convergência (local) do método obtido com passo constante $\eta_k \equiv \eta > 0$, para η suficientemente pequeno (Teorema 1.3.4). Este resultado é de

certo modo o equivalente para problemas com constrangimentos de igualdade ao resultado clássico sobre o método do gradiente para problemas sem constrangimentos.

Seguidamente é desenvolvida uma estratégia a passo variável (isto é, η_k não está fixo) com base nas ideias dos *métodos de gradiente espetral*. Nesses métodos, para problemas sem constrangimentos ou apenas com constrangimentos lineares, o comprimento de passo η_k (que se determina de uma forma inspirada nos métodos de quasi-Newton) pode ser interpretado como o inverso de um cociente de Rayleigh para uma matriz hessiana média da função f. Deste modo conseguem incorporar de forma bastante "barata" alguma informação de segunda ordem na direção $\delta_k = -\nabla f(x_k)$. Esta abordagem é generalizada a problemas com constrangimentos de igualdade (Secção 1.4), associando à direção τ_k um comprimento de passo η_k em que a informação de segunda ordem está ligada a uma hessiana média, mas de uma "função lagrangiana". Não são apresentados quaisquer resultados teóricos (esta é ainda uma questão em desenvolvimento).

No segundo capítulo, os algoritmos introduzidos no primeiro são estendidos a problemas com constrangimentos de desigualdade através de uma estratégia de *conjunto ativo* (Secções 2.2 e 2.4). Tal estratégia passa por definir a cada iteração, de entre o lote de desigualdades, aquelas que são consideradas como *ativas* e as que são *inativas*; as últimas são essencialmente ignoradas, enquanto que as primeiras são impostas como constrangimentos de igualdade. Na metodologia proposta, uma desigualdade é ativada assim que é violada; todavia, a sua desativação dependerá não do facto de deixar de ser violada, mas exclusivamente do sinal do "multiplicador de Lagrange" que lhe está associado (este tipo de critério inspira-se nas condições necessárias de otimalidade de Karush- Kuhn-Tucker).

Como já ficou patente na descrição prévia, e tal como no primeiro capítulo, às iteradas é permitido violarem os constrangimentos. Os métodos obtidos não se destinam portanto a lidar com constrangimentos ditos *essenciais*, isto é, cuja violação torne o problema de otimização mal posto. A exceção a esta regra prende-se com o caso em que as variáveis do problema estão circunscritas a intervalos fechados, uma situação para a qual é delineado um procedimento de modo a que as iteradas cumpram esse tipo de restrição (Secção 2.3).

Os algoritmos de conjunto ativo estabelecidos são ainda generalizados a problemas de otimização robusta em que a função objetivo F do problema é dada pelo máximo entre múltiplas funções (Secção 2.5). Mais concretamente: pode ser o máximo entre um número finito de funções, $F(x) = \max\{f_1(x), \ldots, f_m(x)\}$, ou entre um número infinito de funções, $F(x) = \max_{y \in Y} f(x, y)$ com $Y \subset \mathbb{R}^p$ compacto. As funções $f_1, \ldots, f_m$ e f supõem-se continuamente diferenciáveis (apesar disso, F não é habitualmente diferenciável).

Todos os algoritmos obtidos usam apenas os gradientes das funções envolvidas no problema, inclusive no enquadramento da otimização robusta! Neste, a minimização de F é feita recorrendo apenas aos gradientes $\nabla f_1, \ldots, \nabla f_m$, ou a $\nabla_x f$ e $\nabla_y f$, conforme o caso tratado. Ao longo deste segundo capítulo são apenas tocadas questões práticas, não sendo deduzidos quaisquer resultados teóricos.

No terceiro capítulo é estudada uma classe de funcionais integrais que surge no âmbito de problemas em otimização estrutural com *livre escolha de materiais*. Esses funcionais tomam a forma genérica

$$\Phi(A) = \int_\Omega \phi(A(x))\, dx,$$

onde $A : \Omega \to \mathbb{R}$ varia num conjunto de funções matriciais mensuráveis e essencialmente limitadas, $A \in L^\infty(\Omega; \mathbb{R}^{n,n})$, que verificam em quase todos os pontos x de um aberto limitado $\Omega \subset \mathbb{R}^n$ a condição $A(x) = A(x)^T$. Supõe-se também que dada $A \in \mathbb{R}^{n,n}$ simétrica, o valor $\phi(A)$ da função $\phi : \mathbb{R}^{n,n} \to \mathbb{R}$ depende *exclusivamente* dos valores próprios (ou de forma equivalente, dos invariantes) da matriz A. Uma outra propriedade fundamental imposta sobre ϕ traduz-se por um requisito de *"monotonia"*: ter-se-á sempre $\phi(B) \geqslant \phi(A)$ quando a matriz $B - A$ for semidefinida positiva (Secção 3.2).

Em aplicações práticas o funcional Φ costuma refletir constrangimentos sobre os recursos disponíveis, impostos tipicamente por meio de uma condição do tipo $\Phi(A) \leqslant C$, onde C é uma constante positiva. Assim, num problema de otimização, é essencial que Φ seja semicontínuo inferiormente relativamente a alguma topologia de $L^\infty(\Omega; \mathbb{R}^{n,n})$, sendo principalmente em torno deste tema que giram os resultados do capítulo (Secções 3.3 e 3.4). A topologia adequada para analisar esta questão *não é a fraca*$*$, mas sim a da *convergência no sentido da homogeneização* (ou *H-convergência*).

Em [1] já tinha sido provado que a convexidade de ϕ é condição suficiente para Φ ser H-semicontínuo inferiormente (Corolário 3.3.3), e até mesmo necessária no caso particular em que há apenas dependência do traço (Teorema 3.3.4): $\phi(A) = \varphi(\mathrm{tr}(A))$; para além disso, foram também excluídas noções mais fracas de convexidade como condição suficiente de H-semicontinuidade inferior (Observação 3.3.6). Mostra-se agora que, em geral, a convexidade da integranda não é efetivamente necessária; a confirmação deste facto é feita com recurso a um exemplo vindo da teoria da homogeneização, nomeadamente o do funcional que representa a mistura mais "barata" entre dois materiais isótropos (Secção 3.4). O leque de propriedades possíveis é reduzido a algo *estritamente* entre a convexidade e a policonvexidade de ϕ.

São ainda abordadas algumas propriedades relevantes do ponto de vista prático, a subaditividade e positiva homogeneidade de Φ (Secção 3.5), especialmente no que concerne ao supramencionado funcional provindo da teoria da homogeneização (agora no caso em que se admitem "buracos", ou seja, as misturas são feitas entre material e "vazio"). Contra todas as expetativas, prova-se que este funcional é apenas subaditivo no caso bidimensional (Teorema 3.5.3 e Observação 3.5.5)!

No quarto e último capítulo é generalizada uma grandeza recorrente em otimização estrutural que se destina a avaliar a flexibilidade de uma estrutura sujeita quer a forças aplicadas, quer a deslocamentos impostos sobre parte da sua superfície. O corpo sólido está confinado a $\overline{\Omega}$, sendo $\Omega \subset \mathbb{R}^n$ um conjunto aberto e limitado cuja fronteira $\partial\Omega$ é constituída por pedaços disjuntos Γ_D e Γ_N. O estado desse corpo é caracterizado pelo campo de deslocamentos u que, considerando o modelo da elasticidade linear, satisfaz

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = f & \text{em } \Omega, \\ u = \bar{u} & \text{sobre } \Gamma_D, \\ E\varepsilon(u)\nu = g & \text{sobre } \Gamma_N. \end{cases}$$

Nestas equações $\varepsilon(u)$ denota a parte simétrica da matriz jacobiana de u, f são forças (volúmicas) exercidas em Ω, g são forças (superficiais) actuando sobre Γ_N (ν é a normal exterior unitária a $\partial\Omega$) e $\bar{u}$ é um deslocamento prescrito sobre Γ_D; E denota um tensor de quarta ordem, representante das características físicas do material usado.

Quando o deslocamento imposto $\bar{u}$ é nulo, o *trabalho efetuado pelas forças aplicadas*,

$$\mathcal{W} = \int_\Omega f \cdot u \, dx + \int_{\Gamma_N} g \cdot u \, dx,$$

é o dobro da *energia interna de deformação*

$$\mathcal{E} = \frac{1}{2} \int_\Omega E\varepsilon(u) : \varepsilon(u) \, dx,$$

sendo que a minimização de qualquer destas quantidades equivale de facto a minimizar a flexibilidade (ou se quisermos, a maximizar a rigidez) da estrutura.

Por outro lado, quando não existem forças ($f = g = 0$) e as deformações no corpo são devidas a um deslocamento $\bar{u} \neq 0$ imposto em Γ_D, é conhecido que para maximizar a rigidez da estrutura deve maximizar-se a energia $\mathcal{E}$; ou doutra forma, a medida de flexibilidade é neste caso não $\mathcal{E}$, mas sim $-\mathcal{E}$.

Contudo, quando as forças aplicadas são não nulas e simultaneamente $\bar{u} \neq 0$, é sabido que qualquer das medidas previamente mencionadas é inadequada para representar flexibilidade. Para obviar a esta situação é proposta uma grandeza, designada por *flexibilidade generalizada* (), que combina o trabalho das forças aplicadas com a energia de deformação; mais precisamente:

$$\mathcal{C} = \int_\Omega f \cdot u \, dx + \int_{\Gamma_N} g \cdot u \, dx - \frac{1}{2} \int_\Omega E\varepsilon(u) : \varepsilon(u) \, dx,$$

ou seja, $\mathcal{C} = \mathcal{W} - \mathcal{E}$. É óbvio que $\mathcal{C}$ coincide com as medidas de flexibilidade adotadas nos dois primeiros casos descritos; para além disso, testes computacionais parecem suportar a tese de que $\mathcal{C}$ é efetivamente a grandeza adequada ao caso geral (Secção 4.4). São ainda determinados os *estados adjuntos* necessários para o cálculo das derivadas estruturais de $\mathcal{W}$, $\mathcal{E}$ e $\mathcal{C}$ quando f, g e $\bar{u}$ são não nulos (Secção 4.3).

Palavras-chave

Métodos de gradiente, métodos de conjunto ativo, projeto robusto, homogeneização, projeto com livre escolha de materiais, H-convergência, semicontinuidade inferior, projeto de rigidez máxima.

Contents

Introduction

This thesis touches different aspects of optimization problems, from the design of minimization algorithms to the adoption of adequate measures of a structure's performance in applied fields of structural optimization, passing through more theoretical aspects surrounding a particular type of functional.

The chapters unfold in a way that does not reflect the work's cronological order. This thesis was motivated by structural optimization problems (for instance, to maximize the stiffness of an elastic structure subject to certain efforts and under some constraints). Following the developments achieved during the Master's thesis, the third chapter deepens the study of a particular kind of *cost functionals* used to impose certain restrictions when the main optimization parameter is the very own material from which the structure is made of (a framework commonly known as *free material optimization*).

Through the course of this study the question of how to define the *compliance* of a structure in presence of general loading conditions had to be adressed, an issue for which there was no concrete proposal in the literature by the time the work went underway. This is the fourth chapter's motif.

Constrained optimization being involved, the next logical step was to develop appropriate algorithms, with a preference for "cheap" ones from the computational point of view, which is the subject of the first and second chapters.

A more detailed description follows.

In the first chapter a couple of algorithms is designed to find local minimizers of a continuously differentiable function, which are subject to equality constraints $g = 0$ expressed by means of an also continuously differentiable vector function $g : \mathbb{R}^n \to \mathbb{R}^m$ ($m < n$). It is called to attention that, in general, the iterates produced by these algorithms may violate (even by a large margin) the equation $g = 0$; the constraints become satisfied only asymptotically, after convergence.

The approach falls under a class of methods that, departing from some initial guess $x_0 \in \mathbb{R}^n$, iterate according to $x_{k+1} = x_k + \Delta_k$, where the increment Δ_k may be split in two components corresponding to the twin goals of decreasing f and fulfiling $g = 0$. This can be done, at each iterate x_k, by performing a descent step (for f) in a direction τ_k tangent at x_k to the manifold $g^{-1}(\{g(x_k)\})$, together with a Newton-like step ν_k (for g) orthogonal to τ_k; more precisely, $\Delta_k = \eta_k \tau_k + \nu_k$, with $\eta_k > 0$ designating the step length along the tangential direction.

Only methods in which the direction τ_k is of "steepest descent" for f are considered. In the adressed framework this amounts to choose the orthogonal projection of $-\nabla f(x_k)$ onto the tangent space to the manifold $g^{-1}(\{g(x_k)\})$ at x_k.

A linear (local) rate of convergence is proven for the method with constant step length $\eta_k \equiv \eta > 0$, for sufficiently small η (Theorem 1.3.4). This result is in a certain sense, for equality constrained problems, the equivalent of the classical result for the steepest descent method in unconstrained optimization.

A variable step size strategy (that is, η_k is not fixed) is then devised, based on the ideas behind *spectral gradient methods*. In these methods, when no constraints or only linear constraints are present, the step length η_k can be interpreted as an inverse Rayleigh quotient for some average hessian matrix of f. They are thus able to incorporate in a very "cheap" fashion some second order information into the direction $\delta_k = -\nabla f(x_k)$. This approach is generalized to problems with equality constraints (Section 1.4) ascribing to τ_k a step size η_k in which the second order information is connected to an average hessian of a "lagrangian function". No theoretical results are presented (this is ongoing work).

In the second chapter, the algorithms previously introduced are extended to problems with inequality constraints via an *active-set* strategy (Sections 2.2 and 2.4). Such a strategy consists of defining at each iteration, among all the inequalities, those to be considered as *active* and the ones that are *inactive*; the latter are essentially ignored, while the former are imposed as equality constraints. In the proposed methodology, an inequality is activated as soon as it becomes violated; however, its deactivation depends exclusively on the sign of the corresponding "Lagrange multiplier" (this kind of criteria draws inspiration from the Karush-Kuhn-Tucker necessary conditions of optimality).

As it can be perceived from the previous description, and just like in the first chapter, the obtained methods are *infeasible* (the constraints are not necessarily satisfied). Hence, they are not suited to deal with *essential* constraints, that is, the kind whose violation renders the problem ill-posed. The exception to this rule is the case where the variables implicated in the problem are confined to closed real intervals (bound constraints, or simply *bounds*), a case in which a procedure is delineated so that the iterates are kept *feasible* with respect to this type of restrictions (Section 2.3).

The active-set algorithms established are generalized to *worst-case optimization* problems, in which the objective function F is the maximum between multiple functions (Section 2.5): it may be the maximum $F(x) = \max\{f_1(x), \ldots, f_m(x)\}$ among a finite number of functions, or between an infinite number of them, $F(x) = \max_{y \in Y} f(x, y)$ with $Y \subset \mathbb{R}^p$ compact. The functions $f_1, \ldots, f_m$ and f are always supposed continuously differentiable (despite this fact, F is usually not an everywhere differentiable function).

All the designed algorithms use solely gradient information, worst-case optimization included! In this last setting, the minimization of F is made by resorting to the gradients $\nabla f_1, \ldots, \nabla f_m$, or to $\nabla_x f$ and $\nabla_y f$, depending on the case. During the entire chapter only practical aspects are accounted for and no theoretical results are given.

The third chapter addresses a class of integral functionals arising in applied fields of structural optimization, namely *free material design*. These functionals take the form

$$\Phi(A) = \int_\Omega \phi(A(x))\, dx,$$

where $A : \Omega \to \mathbb{R}$ varies in a subset of bounded measurable matrix functions, and so $A \in L^\infty(\Omega; \mathbb{R}^{n,n})$, which verify for almost every point x of a bounded open set $\Omega \subset \mathbb{R}^n$

the condition $A(x) = A(x)^T$. It is also assumed that $\phi : \mathbb{R}^{n,n} \to \mathbb{R}$ is a *spectral function*, meaning that given a symmetric matrix $A \in \mathbb{R}^{n,n}$, the value $\phi(A)$ depends *exclusively* on the eigenvalues (or equivalently, on the invariants) of the matrix A. Another fundamental property imposed on ϕ translates as a *"monotonicity"* requirement: one has $\phi(B) \geqslant \phi(A)$ whenever $B - A$ is positive semidefinite (Section 3.2).

In practical applications the functional Φ reflects restrictions on the available resources (*"costs"*), typically imposed by means of a constraint $\Phi(A) \leqslant C$, where C is a positive constant. Therefore, regarding some optimization problem, it is essential that Φ satisfies a lower semicontinuity property with respect to some topology on $L^\infty(\Omega; \mathbb{R}^{n,n})$. This is the main subject in the chapter (Sections 3.3 and 3.4). The proper topology to perform the analysis *is not the weak∗ topology*, but that of *convergence in the sense of homogenization* (also known as *H-convergence*).

In [1] it had already been proved that the convexity of ϕ is a sufficient condition for Φ to be lower H-semicontinuous (Corollary 3.3.3), and even a necessary one in the particular case of trace dependence (Theorem 3.3.4): $\phi(A) = \varphi(\mathrm{tr}(A))$; moreover, weaker notions of convexity, like polyconvexity (and thus quasiconvexity and rank-one convexity as well), were also excluded as sufficient conditions for lower H-semicontinuity (Remark 3.3.6). It is now shown that, in general, convexity of the integrand is indeed not necessary; the confirmation is made by introducing an example from the theory of homogenization, namely the functional representing the "cheapest" mixture between two isotropic materials (Section 3.4). The spectrum of possibilities is therefore reduced to some property *strictly* between the convexity and the polyconvexity of ϕ.

Two additional properties, relevant from the practical point of view, are also investigated: subadditivity and positive homogeneity of Φ (Section 3.5), with a special emphasis on the aforementioned functional from the homogenization theory (in the particular case of mixtures between material and "void"). Against all expectations, this functional is proved to be subadditive only in the bidimensional case (Theorem 3.5.3 and Remark 3.5.5)!

In the fourth and final chapter, a recurrent quantity in structural optimization known as *compliance* (to be understood as the opposite of *stiffness*) is generalized. A solid body (the structure) is confined to $\overline{\Omega}$, being $\Omega \subset \mathbb{R}^n$ a bounded open set whose boundary $\partial\Omega$ is split into disjoint pieces Γ_D and Γ_N. The state of that body is described by the field of displacements u which, considering the framework of linear elasticity, satisfies

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = f & \text{in } \Omega, \\ u = \bar{u} & \text{on } \Gamma_D, \\ E\varepsilon(u)\nu = g & \text{on } \Gamma_N. \end{cases}$$

In these equations $\varepsilon(u)$ is the symmetric part of the jacobian matrix of u, f and g denote applied forces, respectively, over Ω and Γ_N (ν is the outward unit normal to $\partial\Omega$), $\bar{u}$ is a prescribed displacement on Γ_D and E designates a fourth order tensor, representing the physical properties of the material from which the structure is made of.

When the imposed displacement $\bar{u}$ is null (*i.e.* the structure is clamped on Γ_D), the *work done by the applied loads*,

$$\mathcal{W} = \int_\Omega f \cdot u \, dx + \int_{\Gamma_N} g \cdot u \, dx,$$

doubles the *elastic energy* stored in the body

$$\mathcal{E} = \frac{1}{2}\int_\Omega E\varepsilon(u) : \varepsilon(u)\, dx.$$

In this case, either quantity is a good measure of compliance and their minimization truly is equivalent to maximizing the structure's stiffness.

On the other hand, when no applied forces exist ($f = g = 0$) and the deformation is caused by a prescribed displacement $\bar{u} \neq 0$ on Γ_D, it is known that in order to obtain a stiff structure one should maximize $\mathcal{E}$; to put it another way, the quantity measuring compliance is now, not $\mathcal{E}$, but $-\mathcal{E}$ instead.

However, when both the applied forces and the prescribed displacement are nonzero, it is a well known fact that none of the previous quantities is an adequate measure of compliance. To overcome this situation a quantity is introduced, termed *generalized compliance*, that combines the work done by the external loads and the elastic energy stored in the body; more precisely:

$$\mathcal{C} = \int_\Omega f \cdot u\, dx + \int_{\Gamma_N} g \cdot u\, dx - \frac{1}{2}\int_\Omega E\varepsilon(u) : \varepsilon(u)\, dx,$$

that is, $\mathcal{C} = \mathcal{W} - \mathcal{E}$. Obviously, $\mathcal{C}$ coincides with the adopted measures of compliance in the first two cases described; furthermore, numerical tests seem to support the claim that $\mathcal{C}$ is indeed appropriate to the general case (Section 4.4). Also presented are the *adjoint states* necessary for the computation of the structural derivative of $\mathcal{W}$, $\mathcal{E}$ and $\mathcal{C}$ when f, g e $\bar{u}$ are all nonzero (Section 4.3).

The first and second chapters of this thesis constitute the preliminary stages of some (hopefully) successful algorithms in the near future. The content consists of original but, to date, unpublished work. The third and fourth chapters are based, respectively, on the work developed in [2,3] and [4].

Chapter 1

Gradient methods for equality constrained problems

The present chapter extends a couple of well known methods in unconstrained minimization to the framework of equality constrained minimization (an extension to inequality constraints is dealt with in the next chapter). The approach works in the general case, meaning that the functions involved may be nonlinear and nonconvex. It is underlined that just *nonessential constraints* are adressed, that is, *constraints whose violation does not render the problem ill-posed*. In the proposed methods, the constraints are usually violated during the optimization process, and become satisfied only asymptotically. One is particularly interested in algorithms with low storage requirements and that rely on first order information, in the sense that only first derivatives are needed (although for theoretical purposes higher order differentiability is sometimes assumed). This stems from the fact that, in several fields of application, gradients are often available at an affordable computational cost, while second derivatives frequently are not.

Sections 1.1 and 1.2 introduce some indispensable background material. Section 1.3 describes the first of the proposed algorithms, complete with a convergence theorem under the hypothesis that a certain hessian-like matrix is positive definite at the solution. Section 1.4 presents the second algorithm, for which the convergence theory is currently motive for research. Several observations are made on possible ways of improving the method. Numerics are postponed until Section 2.6.

On notations

$x \in \mathbb{R}^n$ is the vector of *variables* (also called *unknowns* or *parameters*) and the components of vector or matrix quantities will be denoted with superscripts. The canonical dot product in euclidean spaces is denoted by $\langle\ ,\ \rangle$ and the respective euclidean norm by $\|\ \|$, unless stated or implicitly suggested otherwise; if for some reason one wants to emphasize the canonical euclidean norm, the subscript "2" will be appended at the lower right end. Matrix norms that are associated with vector norms, *e.g.* the ℓ^2 norm

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2, \qquad (A \in \mathbb{R}^{m,n})$$

are denominated *natural norms*.

f is the *objective function*, a scalar function of x that we want to minimize (or maximize); the *constraints* will be modelled by a vector function $g : \mathbb{R}^n \to \mathbb{R}^m$. The jacobian matrix of a vector function g will be denoted by Jg while its transpose will be denoted by ∇g. In particular, for a scalar function f, ∇f will be the usual gradient. Regarding partial derivatives, the *comma notation* is preferred: if f is a scalar function of x, its derivative with respect to the variable x^i will be denoted by $f_{,i}$. The hessian matrix of a scalar function f will be denoted by $\nabla^2 f$.

1.1 Preliminaries

In this section, some well known and essential concepts related to optimization are reviewed. The main algorithms which are central to the understanding of the subsequent developments along the present chapter will also be revisited.

Rates of convergence

One of the basic measures of an algorithm's performance is its rate of convergence. The terminology associated with different kinds of convergence is now recalled.

Let (u_k) be a sequence in a Banach space E (with norm $\|\ \|$) converging to u_*. The convergence is said to be *Q-linear* if there is a constant $r \in]0, 1[$ such that,

$$\text{for all sufficiently large } k, \qquad \frac{\|u_{k+1} - u_*\|}{\|u_k - u_*\|} \leqslant r.$$

When the above condition holds for any $r > 0$, that is, when

$$\lim_{k\to\infty} \frac{\|u_{k+1} - u_*\|}{\|u_k - u_*\|} = 0,$$

the convergence is termed *Q-superlinear*, while if it holds only for some $r \geqslant 1$ it is called *Q-sublinear*. *Q-quadratic* convergence is obtained if there is a positive constant C (not necessarily less than 1) such that,

$$\text{for all sufficiently large } k, \qquad \frac{\|u_{k+1} - u_*\|}{\|u_k - u_*\|^2} \leqslant C.$$

The prefix "Q" stands for "'quotient", since this type of convergence is defined in terms of the quotient of successive errors. Obviously, Q-quadratic convergence is the strongest between the four: it implies Q-superlinear convergence, which in turn implies Q-linear convergence, that implies Q-sublinear convergence (the weakest of the lot). It is possible to define higher rates of convergence, but those are less interesting in practical terms.

A slightly weaker type of convergence, denoted with the prefix "R" for "root", is concerned with the overall decrease in the error, rather than the decrease over each term of the sequence. The convergence of (u_k) to u_* is said to be *R-linear* if there is a sequence of nonnegative scalars (a_k), converging Q-linearly to zero, such that,

$$\text{for all } k, \qquad \|u_k - u_*\| \leqslant a_k.$$

The sequence of errors $(\|u_k - u_*\|)$ is said to be *dominated* by (α_k). Likewise, the convergence is *R-superlinear*, *R-sublinear*, or *R-quadratic*, when the sequence of errors is dominated by a sequence of scalars converging Q-superlinearly, Q-sublinearly, or Q-quadraticly, to zero, respectively.

Unconstrained problems

Unconstrained optimization is the minimization (or maximization) of a scalar function f defined on the whole $\mathbb{R}^n$, or on an open subset of $\mathbb{R}^n$. The best outcome one could hope for in such a problem is to find a *global minimum point*, or *global minimizer*, x_* of f:

$$\text{for all } x \in \mathbb{R}^n, \qquad f(x_*) \leqslant f(x).$$

If the above inequality is satisfied strictly whenever $x \neq x_*$, then x_* is called a *strict global minimum point*, or *strict global minimizer*, of f. Such points can be difficult to spot, since the knowledge of f is usually just local. Most algorithms are able to find only a *local minimum point*, or *local minimizer*, x_* of f:

$$\text{for all } x \in U, \qquad f(x_*) \leqslant f(x),$$

where U is an open neighbourhood of x_* (that is, an open set containing x_*). Again, if the inequality is satisfied strictly whenever $x \neq x_*$, then x_* is called a *strict local minimum point*, or *strict local minimizer*, of f.

Minimizers of a function are characterized by what is commonly known as *optimality conditions* [5, Sec. 2.1]:

1.1.1 Theorem (First-order necessary conditions)**.** *If x_* is a local minimizer and f is continuously differentiable in an open neighbourhood of x_*, then $\nabla f(x_*) = 0$.*

1.1.2 Theorem (Second-order necessary conditions)**.** *Assume that f is a twice continuously differentiable function in an open neighbourhood of a stationary point x_*, i.e. $\nabla f(x_*) = 0$. Then $\nabla^2 f(x_*)$ is positive semidefinite.*

1.1.3 Theorem (Second-order sufficient conditions)**.** *Suppose that f is a twice continuously differentiable function in an open neighbourhood of a stationary point x_*. If $\nabla^2 f(x_*)$ is positive definite, then x_* is a strict local minimizer of f.*

Minimization algorithms require the user to supply a starting point, which will be denoted by x_0. Then, at each iteration, the algorithm chooses a direction δ_k, the *step*, and a positive scalar η_k, the *step length*, to move along from the current iterate x_k towards a new iterate with a (typically) lower function value. There are two main strategies to achieve this goal, usually involving a finite number of trials: *trust-region methods* prescribe the step length η_k and then search in a η_k-ball (with respect to some norm) for a direction δ_k; *line search* methods are in some sense dual, as they first prescribe the direction and then determine the distance η_k to move along δ_k. These kind of procedures are useful for designing *globally convergent* algorithms, meaning that convergence to a stationary point is guaranteed even from remote starting points x_0, which will be not the main concern here (see the book of Nocedal and Wright [5, Chaps. 3,4] for a more in depth analysis). Besides, once a locally convergent algorithm has been devised, with constant step length, one can always modify it to encompass line search or trust-region methodology in order to enhance its convergence properties (this is usually the order things are done anyway).

The steepest descent method

It will be henceforth assumed that f is at least continuously differentiable. It is then quite natural to look for a *descent direction*, that is, a direction δ_k such that $\langle \nabla f(x_k), \delta_k \rangle < 0$. The *steepest descent direction* $\delta_k = -\nabla f(x_k)$ is the most obvious choice, as is the optimal (but in general forbidding) step size $\eta_k = \arg\min_{\eta>0} f(x_k + \eta\delta_k)$, yielding the iteration

$$x_{k+1} = x_k - \eta_k \nabla f(x_k), \qquad \eta_k = \arg\min_{\eta>0} f(x_k - \eta \nabla f(x_k)). \tag{1.1}$$

This *steepest descent method*, introduced by Cauchy [6], is the oldest method for multidimensional unconstrained minimization and its poor practical behaviour is well known. When the level sets of f resemble steep valleys, the sequence (x_k) generated by (1.1) typically exhibits a zigzagging pattern and the speed of convergence becomes very slow. Even in the simplest case of a strictly convex quadratic f, the method converges to the solution with a Q-linear convergence rate that approches 1 as the condition number of $\nabla^2 f$ tends to infinity [5, Sec. 3.3]; hence, despite the eventual strict convexity of the problem, ill-conditioning is enough to prevent the method of making noticeable progress, or even to break it down. Nevertheless, the simplicity of iteration (1.1) is still quite attractive, especially when dealing with large-scale (several variables) problems and if coupled with "cheaper" step size choices (*e.g.* fixed), since it requires solely the computation of $\nabla f(x_k)$.

The following standard results will be used (they can be easily found in textbooks on Functional Analysis):

1.1.4 Theorem (Banach fixed-point theorem)**.** *Assume that K is a nonempty closed set in a Banach space E (with norm $\|\ \|$), and further, that $S : K \to K$ is a contractive mapping (i.e. a Lipschitzian mapping with Lipschitz constant L strictly lower than one). Then there exists a unique $x_* \in K$ such that $x_* = S(x_*)$ and, for any $x_0 \in K$, the sequence (x_k) defined by $x_{k+1} = S(x_k)$, $k \in \mathbb{N}_0$, stays in K and converges Q-linearly to x_*. Furthermore, the following estimate holds: $\|x_k - x_*\| \leqslant L^k \|x_0 - x_*\|$, for all $k \in \mathbb{N}_0$.*

1.1.5 Corollary. *Let $S : E \to E$ be a continuously Fréchet differentiable operator and $x_* \in E$ a point such that $S(x_*) = x_*$. If the Fréchet derivative $DS(x_*)$ of S at x_* has operator norm strictly lower than one, then the conclusions of the previous theorem hold with $K = \{x \in E : \|x - x_*\| \leqslant r\}$, for some $r > 0$. In the finite dimensional case $E = \mathbb{R}^n$, this is equivalent to the requirement that $\|JS(x_*)\| < 1$ for some natural norm.*

The classical local convergence result for the steepest descent method is now presented. The assumptions, as well as the proof, are somewhat different than the usual ones in most of the literature, in the sense that we regard the method as a fixed-point iteration. This choice suits best our reasoning for the algorithm to come in Section 1.3.

1.1.6 Theorem. *Assume that f is a twice continuously differentiable function whose hessian matrix $\nabla^2 f(x_*)$ at a stationary point x_* is positive definite. Then there exists $r > 0$ such that, given $x_0 \in B_r(x_*) = \{x \in \mathbb{R}^n : \|x - x_*\|_2 \leqslant r\}$, the sequence (x_k) generated by the steepest descent method with constant step size*

$$x_{k+1} = x_k - \eta \nabla f(x_k), \quad k \in \mathbb{N}_0,$$

converges Q-linearly to x_* for sufficiently small step lengths $\eta > 0$.

Proof. Taking $S : \mathbb{R}^n \to \mathbb{R}^n$ defined as $S(x) = x - \eta\nabla f(x)$, the steepest descent method becomes $x_{k+1} = S(x_k)$, $k \in \mathbb{N}_0$. Since we are not interested in proving global convergence, the contractivity property will not be needed on the whole $\mathbb{R}^n$, but only locally near x_*. By Corollary 1.1.5, it suffices to check that $\|JS(x_*)\| < 1$ for some natural norm.

It is clear that $JS(x_*) = I - \eta\nabla^2 f(x_*)$ is a symmetric matrix; then we know that the ℓ_2 norm of $JS(x_*)$ coincides with the spectral radius of this same matrix [7, Sec. 1.4]. The eigenvalues of $JS(x_*)$ take the form $1 - \eta\mu_*^i$, where $\mu_*^1 \geqslant \cdots \geqslant \mu_*^n$ are the eigenvalues of $\nabla^2 f(x_*)$; given that the latter are all positive, we have $1 - \eta\mu_*^i \in [1 - \eta\mu_*^1, 1 - \eta\mu_*^n]$ $(1 \leqslant i \leqslant n)$ and the choice $0 < \eta < \frac{2}{\mu_*^1}$ implies that $[1 - \eta\mu_*^1, 1 - \eta\mu_*^n] \subset]-1, 1[$. Hence, one gets $\|JS(x_*)\|_2 = \rho(JS(x_*))$ strictly lower than one. □

Newton and quasi-Newton methods

It was already stressed that the theoretically optimal properties of the steepest descent direction, $\delta_k = -\nabla f(x_k)$, do not have a corresponding match numerically. In practice, other directions perform far better.

Newton's method for unconstrained optimization is a particular case of its counterpart for solving a nonlinear equation $g(x) = 0$ [5, Sec. 11.1], which iterates as

$$x_{k+1} = x_k - [Jg(x_k)]^{-1} g(x_k),$$

being $g : \mathbb{R}^n \to \mathbb{R}^n$. When applied to $g(x) = \nabla f(x)$, one gets the *Newton direction* $\delta_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$; this direction is guaranteed to be of descent if $\nabla^2 f(x_k)$ is positive definite, a property that is usually verified only near a minimizer of f. Hence, Newton's method is typically globalized via a line search procedure that modifies the search direction in order to satisfy the descent property [5, Sec. 3.4]. Methods that use the Newton direction have usually a Q-quadratic rate of local convergence [5, Sec. 3.3]. The main drawback is the need for the hessian $\nabla^2 f(x_k)$, which can be a highly expensive and cumbersome computation.

Quasi-Newton methods [5, Chap. 6] offer an interesting alternative to Newton's method, since they do not require the computation of second derivatives and still attain a fast rate of local convergence (typically Q-superlinear). Instead of the true hessian $\nabla^2 f(x_k)$, they use an approximation B_k that is updated after every new iterate in order to incorporate the knowledge gained during the step. The update of the matrices B_k are based on the fact that variations in the gradient ∇f give information about the second derivative of f along the search direction; more precisely, when x_k and x_{k+1} are close to a minimizer of f (where $\nabla^2 f$ is bound to be positive definite), a simple Taylor expansion shows that

$$\nabla^2 f(x_k)(x_{k+1} - x_k) \approx \nabla f(x_{k+1}) - \nabla f(x_k).$$

Therefore, the new hessian approximation B_{k+1} is chosen to mimic the above property, that is, it is required to verify the following *secant equation*:

$$B_{k+1} s_k = y_k, \tag{1.2}$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$; the *quasi-Newton direction* is then defined as $\delta_k = -B_k^{-1}\nabla f(x_k)$. However, for practical purposes, it is much more advantageous to update the inverse $H_k = B_k^{-1}$ instead, so that δ_k is obtained by a simple

matrix-vector multiplication and not through the solution of a linear system. The updated approximation H_{k+1} must satisfy the secant equation (1.2), now written

$$H_{k+1}y_k = s_k. \tag{1.3}$$

The most effective update formula known is the *BFGS formula*

$$H_{k+1} = (I - \rho_k s_k y_k^T)H_k(I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T, \qquad \rho_k = \langle s_k, y_k\rangle^{-1},$$

and it can be shown to be symmetry and positive definiteness preserving, provided that the initial approximation H_0 is a symmetric positive definite matrix and if s_k and y_k satisfy the so-called *curvature condition*

$$\langle s_k, y_k\rangle > 0.$$

This condition, which is clearly necessary because H_{k+1} satisfies (1.3), will not hold in general and must be enforced by imposing restrictions on the line search procedure that determines the step size η_k.

The spectral gradient method

More than one hundred and forty years after Cauchy having introduced the gradient method, Barzilai and Borwein provided new insight on gradient methods in their groundbreaking paper [8]. They discovered that *in two dimensions*, some choices of η_k (for the steepest descent method) rendered the sequence $(\nabla f(x_k))$ *R-superlinearly convergent* to zero in the case of a strictly convex quadratic f! The resulting algorithm, besides being very inexpensive and having low storage requirements, was globally convergent without any kind of line search. Even more amazing, they showed that the convergence rate improved as the ill-conditioning of the problem increased! The conclusion was at once striking and evident:

> *"It follows that the performance of the steepest descent method cannot be attributed solely to the choice of the search direction."*
> *"Clearly, the behaviour of any algorithm depends on the choice of a step size no less than it depends on the choice of a search direction."*

The belief in an efficient algorithm for large scale minimization, *based only on gradient directions*, began to flourish among the optimization community.

The idea behind the method is very straightforward: recall the secant equation (1.2); now suppose that a matrix B_{k+1} with a very simple structure is sought, namely $B_{k+1} = \sigma I$, $\sigma \in \mathbb{R}$. The equation (1.2) becomes $\sigma s_k = y_k$, which in general cannot be solved; however, if one settles for the choice that minimizes, in euclidean norm, the discrepancy between both members of this equation, one gets

$$\tilde{\sigma}_{k+1} = \arg\min_{\sigma\in\mathbb{R}} \|\sigma s_k - y_k\|_2 = \frac{\langle s_k, y_k\rangle}{\langle s_k, s_k\rangle}. \tag{1.4}$$

Observe that being $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, in the particular case of a quadratic f with a positive definite hessian $\nabla^2 f \equiv A$, one has

$$\tilde{\sigma}_{k+1} = \frac{\langle s_k, As_k\rangle}{\langle s_k, s_k\rangle} > 0$$

and thus, (1.4) is a Rayleigh quotient for A at the vector s_k; $\tilde{\sigma}_{k+1}$ is then between the minimum and maximum eigenvalues of the hessian (hence the terminology *spectral method*). Of course, if one emulates equation (1.3) instead, getting $\sigma^{-1}y_k = s_k$, another coefficient can be derived in the same way:

$$\bar{\sigma}_{k+1} = \arg\min_{\sigma\in\mathbb{R}} \|\sigma^{-1}y_k - s_k\|_2 = \frac{\langle y_k, y_k\rangle}{\langle s_k, y_k\rangle} \tag{1.5}$$

is also a Rayleigh quotient for A, but at the vector $\sqrt{A}\,s_k$. Notice that the Cauchy-Schwarz inequality yields $\tilde{\sigma}_{k+1} \leqslant \bar{\sigma}_{k+1}$. The iterates are then defined through (1.1), but using $\eta_k = \sigma_k^{-1}$ with either (1.4) or (1.5) generating, respectively, $\sigma_k = \tilde{\sigma}_k$ or $\sigma_k = \bar{\sigma}_k$ (along with x_0, an initial step size η_0 must now be given).

The justified attention raised by the Barzilai-Borwein method soon brought further knowledge to the light of day. Global convergence for strictly convex quadratics *in arbitrary dimension* was proven by Raydan [9]; albeit being not competitive with the conjugate gradient method [10, Sec. 7.6] in that case, it was shown to be far superior to classical steepest descent. That result was subsequentely extended to the (not necessarily strictly) convex quadratic case, with bounds on the variables, by the same author together with Friedlander and Martinez [11]. But the hope of obtaining superlinear convergence in arbitrary dimensions was basically discarded in view of Fletcher's work [12, Sec. 4], who argued that, in general, only R-linear convergence should be expected. Furthermore, a strange behaviour seemed to detract the prospects of a successful application to general nonlinear functions: the sequence of values $(f(x_k))$ not only did not decrease monotonically, it in fact violated monotonicity quite severely from time to time! Actually, this behaviour had already been pointed out by the authors of the method [8, Sec. 5]:

> *"Finally, note that, since the two-point algorithms are not descent algorithms, they have an advantage in that the restriction to descent algorithms often results in small step sizes for ill-conditioned problems. This may seem, however, undesirable since it is difficult to control nonmonotone algorithms."*

They even proposed a simple scheme to deal with this issue, probably unaware of the work by Grippo, Lampariello and Lucidi [13]; in their paper, the authors make the case in favor of a line search to globalize Newton's method, in which the objective function is not forced to decrease in a monotonic fashion. Their motivation arose from the fact that the pure Newton method, *i.e.* with unit step size, is sometimes nonmonotone and, despite this fact, the speed of convergence is much faster than the one observed when imposing a strict descent at each iteration. The (GLL for short) strategy proposed, that extends the popular *Armijo rule* [14], was: given $\alpha > 0$, $\beta, \gamma \in]0,1[$ and $M \in \mathbb{N}$, the step size is defined as $\eta_k = \beta^{i_k}\alpha$, where i_k is the first nonnegative integer i satisfying

$$f(x_k + \beta^i\alpha\delta_k) \leqslant \max_{0\leqslant j\leqslant m_k} f(x_{k-j}) + \gamma\beta^i\alpha\langle\nabla f(x_k), \delta_k\rangle, \qquad m_k = \min\{k, M-1\}.$$

Of course, the choice $M = 1$ ensures monotone descent; otherwise, descent is enforced only every M iterations, thus leaving room for occasional increases in function values.

The ground was layed for the implementation of the Barzilai-Borwein method for general unconstrained minimization. And it was again Raydan [15] who gave the decisive

push that put spectral gradient methods on the radar thereafter. Using a line search based on a GLL strategy, he proved a global convergence result and exhibited numerical tests which showed, somewhat surprisingly, that the method was highly competitive with up-to-date implementations of conjugate gradient methods for minimizing general functions, sometimes even outperforming them. Since then many developments ensued, as it will be timely mentioned thoughout the chapter, but Fletcher's telling words in a 2001 report [16, Sec. 4] still carry some truth in them up to this day:

> *"One thing that I think emerges from this review is just how little we understand about the BB method."*

1.2 Constrained problems

In constrained optimization, besides the *objective function* $f : \mathbb{R}^n \to \mathbb{R}$, a *constraint function* $g : \mathbb{R}^n \to \mathbb{R}^m$ is given defining certain equations (or inequations) that the vector x of unknowns must satisfy. If one is required to minimize f, the optimization problem can, for instance, be written (considering only equality constraints):

$$\min_{x \in \mathcal{C}_*} f(x), \qquad \mathcal{C}_* = \{x \in \mathbb{R}^n : \ g(x) = 0\}. \tag{1.6}$$

In coordinate notation:

$$\mathcal{C}_* = \{x \in \mathbb{R}^n : \ g^i(x) = 0, \ 1 \leqslant i \leqslant m\}.$$

The level set $\mathcal{C}_*$ is usually called the set of *feasible points*, or *feasible set*. It is assumed that, besides f, also g is at least continuously differentiable.

Constrained optimization problems arise from models in which constraints play an essential role, for instance in imposing budgetary and shape constraints in a structural design problem (see Chapters 4 and 5). They can be reformulated as unconstrained problems, if the constraints are replaced by "penalization terms" added to the objective function having the effect of discouraging constraint violations [17, Chap. 13], or by other means (*e.g.* by parametrizing the feasible set $\mathcal{C}_*$). This approach will not be followed here.

One of the most often used concepts while designing algorithms is the subject of the following definition.

1.2.1 Definition. *A point $x \in \mathbb{R}^n$ satisfying the constraint $g(x) = 0$ is said to be a regular point if the gradient vectors $\nabla g^1(x), \nabla g^2(x), \ldots, \nabla g^m(x)$ are linearly independent. In other words, the jacobian matrix $Jg(x)$ should have full rank (equal to m).*

The condition defining a regular point is often termed across the literature as *linear independence constraint qualification* (LICQ). It is the most common from a lot of "minimal hypotheses" usually called *constraint qualifications* [18, Sec. 11.3], which are assumed to hold when establishing optimality conditions for constrained problems.

The previous definition implies that $m \leqslant n$. However, since $m = n$ would yield a discrete set of feasible points, a situation which is outside the scope of the present text, it is hereafter assumed that $m < n$. At a regular point x then, the constraint function g

is a submersion, thus giving $\mathcal{C}_*$ the appropriate geometrical concept of a submanifold in $\mathbb{R}^n$ [19, Sec. 1.2]. The tangent subspace to $\mathcal{C}_*$ at x [17, Sec. 11.2] is given by

$$\mathcal{T}_x = \operatorname{Ker}(Jg(x)) = \{\tau \in \mathbb{R}^n : \ Jg(x)\tau = 0\}.$$

The following result, relating the distance to the manifold $\mathcal{C}_*$ and the norm of the constraint function defining it, will be used in Section 1.3. The statement is that, locally, those two quantities have the same order of magnitude. Part of the proof relies on the implicit function theorem, much in the same vein like a standard result about manifolds defined as inverse images [19, Secs. 1.1, 1.2].

1.2.2 Lemma. *Let $x_* \in \mathcal{C}_*$ be a regular point. Then there exist positive constants C_1, C_2 and r, such that,*

$$\text{for every } x \in B_r(x_*), \qquad C_1 \operatorname{dist}(x, \mathcal{C}_*) \leqslant \|g(x)\| \leqslant C_2 \operatorname{dist}(x, \mathcal{C}_*).$$

Proof. The second inequality is straightforward: since $U \cap \mathcal{C}_*$ is compact for every compact neighbourhood U of x_*, given x close enough to x_* it is always possible to consider $y_x \in \mathcal{C}_*$ satisfying $\|x - y_x\| = \inf_{y \in \mathcal{C}_*} \|x - y\| = \operatorname{dist}(x, \mathcal{C}_*)$; a simple Taylor expansion about y_x yields $g(x) = \mathcal{O}(\|x - y_x\|)$, that is, $\|g(x)\| \leqslant C_2 \operatorname{dist}(x, \mathcal{C}_*)$.

The first inequality is far less trivial. Being x_* a regular point, there is a nonzero $m \times m$ minor of $Jg(x_*)$; for simplicity's sake, assume that it is the one featuring the last m columns of that matrix. Then, by the implicit function theorem, there is an open neighbourhood of x_* where the last m variables $\tilde{x} = (x^{n-m+1}, \dots, x^n)$ are a function of the first $n-m$ variables $\bar{x} = (x^1, \dots, x^{n-m})$; more precisely, splitting $\mathbb{R}^n$ into $\mathbb{R}^{n-m} \times \mathbb{R}^m$, there are open neighbourhoods U of $\bar{x}_*$ and V of $\tilde{x}_*$, and a continuously differentiable function $\varphi : U \to V$, such that $(U \times V) \cap \mathcal{C}_*$ is the graph of the map $\bar{x} \mapsto g(\bar{x}, \varphi(\bar{x}))$.

Now, let $J_{\tilde{x}} g(x)$ represent the matrix formed by the last m columns of $Jg(x)$. Using the fact that $(\bar{x}, \varphi(\bar{x})) \in \mathcal{C}_*$ and the mean value theorem, one can write

$$\|g(x)\| = \|g(\bar{x}, \tilde{x}) - g(\bar{x}, \varphi(\bar{x}))\| = \|J_{\tilde{x}} g(\bar{x}, \tilde{y})(\tilde{x} - \varphi(\bar{x}))\|,$$

where $\tilde{y}$ lies in the line segment joining $\tilde{x}$ and $\varphi(\bar{x})$. Next, define $\theta : \mathbb{R}^n \to \mathbb{R}$ in the following way: $\theta(x) = \min_{\tilde{\delta}} \|J_{\tilde{x}} g(x)\tilde{\delta}\|$, where the minimum is computed over all vectors $\tilde{\delta} \in \mathbb{R}^m$ such that $\|\tilde{\delta}\| = 1$. Note that θ is well defined, because the unit sphere is compact and the map $\tilde{\delta} \mapsto \|J_{\tilde{x}} g(x)\tilde{\delta}\|$ is obviously continuous. Since $J_{\tilde{x}} g(x_*)$ is an invertible matrix, $\theta(x_*) > 0$; thus, a property like lower semicontinuity will suffice to ensure that θ is bounded below away from zero near x_*. The conclusion of Lemma 1.2.2 follows then easily, because

$$\|J_{\tilde{x}} g(\bar{x}, \tilde{y})(\tilde{x} - \varphi(\bar{x}))\| \geqslant \theta(\bar{x}, \tilde{y}) \|\tilde{x} - \varphi(\bar{x})\| \geqslant C_1 \|(\bar{x}, \tilde{x}) - \underbrace{(\bar{x}, \varphi(\bar{x}))}_{\in \mathcal{C}_*}\| \geqslant C_1 \operatorname{dist}(x, \mathcal{C}_*).$$

For proving that θ is lower semicontinuous,[1] let $x_k \to x$ and take a subsequence (x_{p_k}) such that $\liminf \theta(x_k) = \lim \theta(x_{p_k})$ and $\theta(x_{p_k}) = \|J_{\tilde{x}} g(x_{p_k})\tilde{\delta}_{p_k}\|$. Using the compactness of the unit sphere, pick a subsequence of $\tilde{\delta}_{p_k}$ converging to some $\tilde{\delta}$; then, $\liminf \theta(x_k) = \|J_{\tilde{x}} g(x)\tilde{\delta}\| \geqslant \theta(x)$. □

[1] Note that θ is obviously upper semicontinuous, since it is the lower envelope of the $\tilde{\delta}$-indexed family of continuous functions given by $x \mapsto \|J_{\tilde{x}} g(x)\tilde{\delta}\|$. So, in fact, θ is a continuous function.

The *optimality conditions* for constrained optimization problems [17, Secs. 11.3, 11.5] are, not surprisingly, more complicated than for the unconstrained case.

1.2.3 Theorem (First-order necessary conditions)**.** *Assume that $x_* \in \mathbb{R}^n$ is a solution of (1.6). If x_* is a regular point, then there exists (a unique) $\lambda_* \in \mathbb{R}^m$, called the Lagrange multiplier, such that the following conditions hold:*

$$\begin{cases} \nabla f(x_*) + \nabla g(x_*)\lambda_* = 0, \\ g(x_*) = 0. \end{cases} \tag{1.7}$$

In coordinate notation:

$$\begin{cases} f_{,j}(x_*) + \sum_{i=1}^m \lambda_*^i g_{,j}^i(x_*) = 0, \ 1 \leqslant j \leqslant n, \\ g^i(x_*) = 0, \ 1 \leqslant i \leqslant m. \end{cases}$$

These equations are often referred to as *Karush-Kuhn-Tucker conditions*, or *KKT conditions* for short, and they are the equivalent of the stationarity condition $\nabla f(x_*) = 0$ in unconstrained optimization. It should not come as a surprise then, that them alone are not sufficient to ensure optimality.

In connection with general problems of the form (1.6), it is usually convenient to introduce the *lagrangian function* $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, defined as

$$L(x, \lambda) = f(x) + \langle g(x), \lambda \rangle.$$

By splitting $\nabla L = (\nabla_x L, \nabla_\lambda L)$, the KKT conditions can be rewritten as

$$\begin{cases} \nabla_x L(x_*, \lambda_*) = 0, \\ \nabla_\lambda L(x_*, \lambda_*) = 0, \end{cases}$$

i.e. they express stationarity of the lagrangian L at (x_*, λ_*).

As in the unconstrained case, second-order derivatives are required to discern minimum points. Henceforth, whenever f and g are at least twice continuously differentiable functions, $\nabla_x^2 L(x, \lambda)$ will denote the matrix $\nabla^2 f(x) + \sum_{j=1}^m \lambda^j \nabla^2 g^j(x)$.

1.2.4 Theorem (Second-order necessary conditions)**.** *Suppose that $x_* \in \mathbb{R}^n$ is both a solution to (1.6) and a regular point. Assume that f and g are twice continuously differentiable functions in an open neighbourhood of x_*. If λ_* is the Lagrange multiplier for which (1.7) is satisfied, then:*

$$\text{for all } \tau \in \mathcal{T}_{x_*}, \qquad \langle \nabla_x^2 L(x_*, \lambda_*)\tau, \tau \rangle \geqslant 0,$$

that is, the matrix $\nabla_x^2 L(x_*, \lambda_*)$ is positive semidefinite on $\mathcal{T}_{x_*}$.

1.2.5 Theorem (Second-order sufficient conditions)**.** *Suppose there are $x_* \in \mathbb{R}^n$ and a Lagrange multiplier λ_* such that the KKT conditions (1.7) hold. Assume that f and g are twice continuously differentiable functions in an open neighbourhood of x_*. Suppose also that the matrix $\nabla_x^2 L(x_*, \lambda_*)$ is positive definite on $\mathcal{T}_{x_*}$, that is,*

$$\text{for all nonzero } \tau \in \mathcal{T}_{x_*}, \qquad \langle \nabla_x^2 L(x_*, \lambda_*)\tau, \tau \rangle > 0.$$

Then x_* is a strict local minimizer of (1.6).

1.3 A steepest descent approach

A typical case in structural design arises when engineers adjust the parameters (variables) to optimize the performance of a structure while keeping a prescribed *cost*. In such a framework, the constraint function g appearing in (1.6) is thought of as a *cost function* (see Chapter 4), a scalar function that (in a broad sense) stands for the structure's "price" (or more precisely, the difference between the cost function and a prescribed "price"). For presentation purposes, the discussion will be initially restricted to this model problem ($m = 1$) and subsequently extended to account for multiple constraints.

For the treatment of (1.6) we will try to follow, to a certain extent, some of the ideas in Section 1.1 (see pages 8–9). The question of which search direction one should consider does not have an immediate answer. There is more than one aspect to cover as the iterations progress: decreasing the function value of f while solving the equation $g = 0$. The approach proposed by Barbarosie [20] sets up a direction that targets both goals simultaneously, much in the manner of well known gradient projection methodology [17, Sec. 12.4], but with two major differences: *the iterates do not necessarily satisfy the constraints* and *no projection matrices are used*.

Given an iterate x_k, the direction δ_k used to define the next iterate, $x_{k+1} = x_k + \eta_k \delta_k$, is the sum of two components: one of them is the step $-\nabla f(x_k)$ corresponding to the steepest descent method; the other one aims at fulfilling the constraint equation $g = 0$ and has the form $-\lambda_k \nabla g(x_k)$, where $\lambda_k \in \mathbb{R}$ is a sort of Lagrange multiplier:

$$\delta_k = -\nabla f(x_k) - \lambda_k \nabla g(x_k) = -\nabla_x L(x_k, \lambda_k),$$

with the lagrangian being now simply $L(x, \lambda) = f(x) + \lambda g(x)$. The multiplier λ_k is defined in a natural way by imposing on the increment $\Delta_k = \eta_k \delta_k$ the Newton-type condition

$$\langle \nabla g(x_k), \Delta_k \rangle = -g(x_k), \tag{1.8}$$

which is immediately solvable:

$$\lambda_k = \frac{\eta_k^{-1} g(x_k) - \langle \nabla g(x_k), \nabla f(x_k) \rangle}{\|\nabla g(x_k)\|^2}. \tag{1.9}$$

With this choice of the multiplier, the whole procedure amounts to perform a tangential gradient method to minimize f, together with a unidimensional Newton method to solve the constraint equation $g = 0$.

To better understand the last assertion, consider the following reasoning. In the neighborhood of a solution x_* there are two main directions to consider from x_k: the direction $\nabla g(x_k)$, orthogonal to the level set

$$\mathcal{C}_k = \{x \in \mathbb{R}^n : \ g(x) = g(x_k)\},$$

and the subspace orthogonal to it (whose vectors are tangent to $\mathcal{C}_k$ at x_k). In this latter subspace we have to minimize f; note that, since the solution x_* should minimize f in a level set of g, namely $\mathcal{C}_*$, there is no point in decreasing f along directions other than tangent ones. In the direction $\nabla g(x_k)$ we want to solve the equation $g = 0$, moving the next iterate closer to $\mathcal{C}_*$. To clarify things further, take an orthogonal basis of $\mathbb{R}^n$ determined

by the unit vector $\nu_k = \|\nabla g(x_k)\|^{-1}\nabla g(x_k)$; then we can write $\nabla f(x_k) = \tau_k + \alpha_k\nu_k$, where $\alpha_k \in \mathbb{R}$ and $\tau_k \perp \nu_k$. With these notations we get

$$\lambda_k = \frac{\eta_k^{-1}g(x_k) - \alpha_k\|\nabla g(x_k)\|}{\|\nabla g(x_k)\|^2} = \frac{\eta_k^{-1}g(x_k)}{\|\nabla g(x_k)\|^2} - \frac{\alpha_k}{\|\nabla g(x_k)\|}$$

and therefore

$$\Delta_k = -\eta_k\nabla f(x_k) - \eta_k\lambda_k\nabla g(x_k) = -\eta_k\tau_k - \frac{g(x_k)}{\|\nabla g(x_k)\|}\nu_k.$$

The tangential component of Δ_k, equal to $-\eta_k\tau_k$, is a descent direction for f at x_k:

$$\langle\nabla f(x_k), -\eta_k\tau_k\rangle = \langle\tau_k + a_k\nu_k, -\eta_k\tau_k\rangle = -\eta_k\|\tau_k\|^2 < 0.$$

The normal component of Δ_k, equal to $-\frac{g(x_k)}{\|\nabla g(x_k)\|}\nu_k$, clearly alludes to a one-dimensional Newton method.

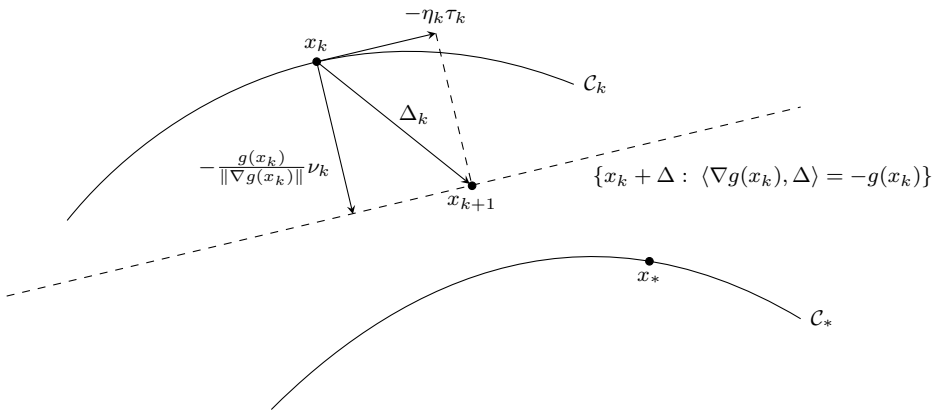

Figure 1.1: Structure of the step.

As mentioned at the beginning of this section, the algorithm generalizes naturally to vector-valued constraint functions $g : \mathbb{R}^n \to \mathbb{R}^m$ (with $m < n$). In this case $\lambda_k \in \mathbb{R}^m$, but the iterates are still defined in a similar fashion by

$$x_{k+1} = x_k + \eta_k\delta_k, \qquad \delta_k = -\nabla f(x_k) - \nabla g(x_k)\lambda_k = -\nabla_x L(x_k, \lambda_k), \tag{1.10}$$

where the Newton-type condition (1.8) now reads

$$Jg(x_k)\Delta_k = -g(x_k), \qquad \Delta_k = \eta_k\delta_k = x_{k+1} - x_k, \tag{1.11}$$

on account of which (1.9) transforms to

$$Jg(x_k)\nabla g(x_k)\lambda_k = \eta_k^{-1}g(x_k) - Jg(x_k)\nabla f(x_k). \tag{1.12}$$

In coordinate notation:

$$x_{k+1}^i = x_k^i - \eta_k f_{,i}(x_k) - \eta_k\sum_{j=1}^m g_{,i}^j(x_k)\lambda_k^j, \quad 1 \leqslant i \leqslant n,$$

where

$$\sum_{i=1}^{n}\sum_{j=1}^{m} g^l_{,i}(x_k)g^j_{,i}(x_k)\lambda^j_k = \eta_k^{-1}g^l(x_k) - \sum_{i=1}^{n} g^l_{,i}(x_k)f_{,i}(x_k), \quad 1 \leqslant l \leqslant m.$$

The linear system of equations (1.12) uniquely determines λ_k if $Jg(x_k)$ has full rank (equal to m); see Definition 1.2.1 and the comments following it. The method can still be interpreted geometrically as a steepest descent method in the directions tangent to $\mathcal{C}_k$, combined with a Newton method in the directions normal to $\mathcal{C}_k$.

1.3.1 Remark. Observe that, in view of the KKT conditions (1.7), the solution of (1.12) computed at a regular point x_*, minimizer of (1.6), is exactly the matching λ_*. Therefore, if $x_k \approx x_*$ then λ_k is roughly equal to the true Lagrange multiplier (see Corollary 1.3.9).

Note also that, despite (1.12) having been obtained in connection with the iteration (1.10), it can stand on its own as an independent formula. In fact, (1.12) can be regarded as a particular case of

$$Jg(x_k)\nabla g(x_k)\lambda_k = \xi_k g(x_k) - Jg(x_k)\nabla f(x_k), \qquad \xi_k \geqslant 0, \tag{1.13}$$

which also contains the well known *least-squares multipliers* [18, Sec. 12.3], corresponding to the choice $\xi_k = 0$. For practical purposes, while keeping $\xi_k > 0$, it is best to ensure that (ξ_k) stays bounded, because if $\xi_k g(x_k) \approx 0$ when $x_k \approx x_*$ – and if $Jg(x_k)\nabla g(x_k)$ is not ill-conditioned – one is certain of having $\lambda_k \approx \lambda_*$. ■

A sketch of the iterative procedure now follows.

1.3.2 Algorithm.
INPUT: initial guess x_0, tolerance $\varepsilon > 0$, maximum number of iterations N.
OUTPUT: approximate solution x, or message of failure.
Step 1 Set $k = 1$.
Step 2 While $k \leqslant N$ do Steps 3–8.
  Step 3 Choose a step size $\eta > 0$.
  Step 4 Solve $Jg(x_0)\nabla g(x_0)\lambda = \eta^{-1}g(x_0) - Jg(x_0)\nabla f(x_0)$.
  Step 5 Set $x = x_0 - \eta[\nabla f(x_0) + \nabla g(x_0)\lambda]$.
  Step 6 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
    STOP.
  Step 7 Set $x_0 = x$.
  Step 8 Set $k = k + 1$.
Step 9 OUTPUT('The method failed after N iterations.');
  STOP.

1.3.3 Remark. Step 3 of the previous algorithm, like the choice of a search direction and precisely for the same reasons, does not have an easy answer (except if η is fixed to begin with). The competing goals of decreasing the objective function and satisfying the constraints have to be balanced, especially when dealing with algorithms that allow iterates to leave the feasible set (as is the case). *Merit functions* and *filters* [5, Sec. 15.4] are two concepts that provide a suitable framework for achieving that balance. However, it is an approach that will not be followed here. An alternative course of action is suggested in the sequel (see Remark 1.3.10). ■

The main theorem concerning the proposed method is now stated and proved.

1.3.4 Theorem. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ ($m < n$) be twice continuously differentiable functions. Suppose there is a pair (x_*, λ_*) satisfying the KKT conditions (1.7), $x_* \in \mathcal{C}_*$ being a regular point, and that the matrix $\nabla_x^2 L(x_*, \lambda_*)$ is positive definite on $\mathcal{T}_{x_*} = \mathrm{Ker}(Jg(x_*))$. Then there exists $r > 0$ such that, given $x_0 \in B_r(x_*)$, the sequence of iterates generated by*

$$x_{k+1} = x_k - \eta[\nabla f(x_k) + \nabla g(x_k)\lambda_k], \quad k \in \mathbb{N}_0, \tag{1.14}$$

where λ_k is determined through

$$Jg(x_k)\nabla g(x_k)\lambda_k = \eta^{-1} g(x_k) - Jg(x_k)\nabla f(x_k), \tag{1.15}$$

converges Q-linearly to x_* for sufficiently small step lengths $\eta > 0$.

The reasoning follows the same pattern of the proof of Theorem 1.1.6, but a bit more care will have to be exercised in this case. First of all, an auxiliary result is established.

1.3.5 Lemma. *Let $P \not\equiv 0$ be an orthogonal projection on $\mathbb{R}^n$. If $A \not\equiv 0$ is a self-adjoint linear operator on $\mathbb{R}^n$, then $v \neq 0$ is an eigenvector of PA, associated with the eigenvalue $\mu \neq 0$, if and only if*

(i) $v \in \mathrm{Ran}(P)$,

(ii) $(A - \mu I)v \in \mathrm{Ker}(P)$.

Hence, the following estimate of the spectral radius holds: $\rho(PA) \leqslant \rho(A\big|_{\mathrm{Ran}(P)})$.

Proof. The "if" part of the assertion is trivial. The "only if" part follows basically from the fact that, P being an orthogonal projection, one has the direct sum decomposition $\mathbb{R}^n = \mathrm{Ker}(P) \oplus \mathrm{Ran}(P)$. Hence, given an eigenpair $u \neq 0$ and $\mu \neq 0$ of PA, there are unique $v \in \mathrm{Ker}(P)$ and $w \in \mathrm{Ran}(P)$ such that $Au = v + w$; but then, $PAu = \mu u$ reads $w = \mu u$. Therefore, it must be $u \in \mathrm{Ran}(P)$ and $Au - \mu u = v \in \mathrm{Ker}(P)$.

The last estimate is now obvious, since $\rho(PA) = \rho(PA\big|_{\mathrm{Ran}(P)})$ and the spectral radius of an operator is dominated by the ℓ^2 norm of that same operator (recall also that $\|P\|_2 = 1$ and that the spectral radius of a self-adjoint operator equals its ℓ^2 norm). □

1.3.6 Remark. Another useful result regarding spectral radii and matrix norms is that, for any square matrix A and $\varepsilon > 0$, there exists a natural norm with the property that $\|A\| < \rho(A) + \varepsilon$ [7, Sec. 1.4]. Then, recalling the considerations made in Corollary 1.1.5, one concludes that contractivity properties of differentiable operators $S : \mathbb{R}^n \to \mathbb{R}^n$ are essentially governed by the spectral radius of their jacobian matrices: if $\rho(JS(x)) < 1$, it always exists a vector norm for which S is locally contractive around x. ■

Proof of Theorem 1.3.4. We begin by rewriting the algorithm to display its fixed point nature. Assuming that (1.15) has a unique solution

$$\lambda_k = [Jg(x_k)\nabla g(x_k)]^{-1}[\eta^{-1} g(x_k) - Jg(x_k)\nabla f(x_k)],$$

putting this expression into (1.14) yields

$$x_{k+1} = x_k - \eta \underbrace{\left[I - \nabla g(x_k)[Jg(x_k)\nabla g(x_k)]^{-1} Jg(x_k)\right]}_{P(x_k)} \nabla f(x_k)$$
$$- \underbrace{\nabla g(x_k)[Jg(x_k)\nabla g(x_k)]^{-1}}_{K(x_k)} g(x_k);$$

so $x_{k+1} = S(x_k)$, upon defining $S(x) = x - \eta P(x)\nabla f(x) - K(x)g(x)$. Because x_* is a regular point, $Jg(x_*)$ has full rank – the same is true then for $Jg(x)$ at nearby x – and the operator S is thus well defined locally around x_*. Because of Remark 1.3.6, one is left to establish $\rho(JS(x_*)) < 1$.

$K(x)$ is clearly a right inverse of $Jg(x)$ and it is not difficult to prove that $P(x)$ is the matrix of the orthogonal projection onto the tangent subspace $\mathcal{T}_x$ to the level set

$$\mathcal{C}_x = \{y \in \mathbb{R}^n : \ g(y) = g(x)\}$$

at x. There are some trivial relations involving $P(x)$, $K(x)$ and $Jg(x)$, namely:

$$K(x)Jg(x) = I - P(x), \quad P(x)K(x) = 0 \quad \text{and} \quad P(x)\nabla g(x) = 0;$$

in view of this last equality, one can write

$$S(x) = x - \eta P(x)[\nabla f(x) + \nabla g(x)\lambda_*] - K(x)g(x),$$

and it is now easy to see, due to the KKT conditions (1.7), that the Jacobian matrix of S at x_* is given by

$$\begin{aligned} JS(x_*) &= I - \eta P(x_*)\nabla^2_x L(x_*, \lambda_*) - K(x_*)Jg(x_*) \\ &= I - \eta P(x_*)\nabla^2_x L(x_*, \lambda_*) - [I - P(x_*)] = P(x_*)[I - \eta\nabla^2_x L(x_*, \lambda_*)]. \end{aligned}$$

Since $I - \eta\nabla^2_x L(x_*, \lambda_*)$ is a symmetric matrix and $P(x_*)$ is the orthogonal projection's matrix onto $\mathcal{T}_{x_*}$, precisely the subspace where $\nabla^2_x L(x_*, \lambda_*)$ is positive definite, recalling Lemma 1.3.5 and the proof of Theorem 1.1.6, the conclusion is now at hand. □

1.3.7 Remark. The constraints converge faster than the iterates. In fact, a simple Taylor expansion about x_k yields

$$g(x_{k+1}) = \underbrace{g(x_k) + Jg(x_k)(x_{k+1} - x_k)}_{= 0, \text{ by (1.11)}} + \mathcal{O}(\|x_{k+1} - x_k\|^2);$$

thus, if $L < 1$ designates the (local) contractivity constant of the operator S (see the previous proof), it is clear that $\|x_{k+1} - x_k\|^2 = \mathcal{O}(L^{2k})$. However, notice that this does not yield Q-quadratic convergence of $(g(x_k))$, but simply an improved R-linear one. ■

1.3.8 Remark. In view of Lemma 1.2.2, one can assert that locally around x_*, $\|g(x_k)\|$ provides a reasonable estimate of $\text{dist}(x_k, \mathcal{C}_*)$. Thus, Remark 1.3.7 implies that the distance between x_k and the manifold $\mathcal{C}_*$, defined by the constraints, converges to zero faster than the distance $\|x_k - x_*\|$. ■

1.3.9 Corollary. *Under the hypotheses of Theorem 1.3.4, the sequence* (λ_k) *generated by (1.14)-(1.15) is locally R-linear convergent to* λ_**:* $\|\lambda_k - \lambda_*\| = \mathcal{O}(\|x_k - x_*\|)$.

Proof. The multipliers λ_k and λ_* are, respectively, the value at x_k and the value at x_* of the vector function given by

$$\lambda(x) = [Jg(x)\nabla g(x)]^{-1}[\eta^{-1}g(x) - Jg(x)\nabla f(x)],$$

which is well defined in a neighbourhood of x_*. Since f and g are twice continuously differentiable functions, λ is continuously differentiable and thus locally lipschitzian. □

During the proof of Theorem 1.3.4, it was seen that algorithm (1.14)-(1.15) can be rewriten as

$$x_{k+1} = x_k - \eta P_k \nabla f(x_k) - K_k g(x_k), \quad k \in \mathbb{N}_0, \tag{1.16}$$

where

$$K_k = \nabla g(x_k)[Jg(x_k)\nabla g(x_k)]^{-1} \quad \text{and} \quad P_k = I - K_k Jg(x_k), \tag{1.17}$$

being the latter the matrix of the orthogonal projection onto $\mathcal{T}_k = \mathrm{Ker}(Jg(x_k))$, the tangent subspace to the level set $\mathcal{C}_k = \{x \in \mathbb{R}^n : \ g(x) = g(x_k)\}$ at x_k.

Observe that, while under the form (1.16)-(1.17), computational costs have now doubled; the tangential and normal directions $\tau_k = -P_k\nabla f(x_k)$ and $\nu_k = -K_k g(x_k)$, respectively, involve the solution of two linear systems, whereas (1.14)-(1.15) requires solely the one corresponding to the multiplier λ_k. Notice however, besides the coefficient matrix in both of those systems being the same, that they are independent of each other and so the computation of their solutions can be parallelized, resulting in a total CPU time equal to the one needed by the simpler iteration (1.14)-(1.15).

1.3.10 Remark. Concerning the adoption of an adaptive step size strategy, there is an advantage in considering (1.16)-(1.17). The twin goals of decreasing the objective function and of satisfying the constraints, are now clearly split in the increment

$$\Delta_k = -\eta P_k \nabla f(x_k) - K_k g(x_k) = \eta\tau_k + \nu_k.$$

Instead of using a fixed step length η, one can now (for instance) use a line search method along the tangential direction τ_k to look for a suitable step size η_k to decrease f. Along ν_k the unit step is preferred, since the normal component is not directly related to minimization but with the solution of an equation through a Newton-like method.

1.4 A spectral gradient approach

The Barzilai-Borwein method presented in pages 10–11 is now extended to the general nonlinear equality constrained case.

Rather surprisingly, there is no literature on spectral gradient methods to deal in a systematic way with problems like (1.6). Besides the works of Martinez, Pilotta and Raydan [21], which is restricted to linear constraints anyway, and of Diniz-Ehrhardt *et al.* [22], where the spectral method is simply an auxiliary tool in a penalty-like approach, there is only the recent article by Gomes-Ruggiero, Martinez and Santos [23]. This latter paper addresses in fact a wider problem than (1.6): the variables x lie in a closed convex polytope

and the constraints are given under the form of inequalities[2]. However, and regarding the spectral aspect of the algorithm, the approach is a generalization of the methods by Birgin, Martinez and Raydan [24–26], which in turn merge the spectral nonmonotone ideas (see page 11) with classical projected gradient methods on convex sets.

The approach that will be introduced in the sequel *differs from all of the above in the computation of the spectral parameter*; more precisely, in the choices of vectors s_k and y_k.

1.4.1 Note. Although the established convergence theory of spectral gradient methods in the unconstrained case is the same for both Barzilai-Borwein step length choices σ_k^{-1},

$$\sigma_k = \frac{\langle s_{k-1}, y_{k-1}\rangle}{\langle s_{k-1}, s_{k-1}\rangle} \qquad \text{or} \qquad \sigma_k = \frac{\langle y_{k-1}, y_{k-1}\rangle}{\langle s_{k-1}, y_{k-1}\rangle}, \tag{1.18}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1})$, computational evidence reported throughout the literature clearly favours the former as the superior formula. For that reason it will take the "spotlight" from now on and, unless stated otherwise, σ_k will always mean the first of the above spectral coefficients.

As a side note, a remarkable property both formulas share is that they render the resulting algorithms invariant for rescalings, in both f and the independent variables x (unlike the steepest descent method). ■

Recall that in the unconstrained case and if f is a quadratic function, σ_k is a Rayleigh quotient for the actual hessian $\nabla^2 f$. In the general case of a nonlinear f, only a weaker statement can be made: since

$$y_{k-1} = \nabla f(x_k) - \nabla f(x_{k-1}) = \left(\int_0^1 \nabla^2 f(x_{k-1} + t s_{k-1})\, dt \right) s_{k-1}, \tag{1.19}$$

σ_k is just a Rayleigh quotient for an average hessian matrix. Unless f is convex, the dot product $\langle s_{k-1}, y_{k-1}\rangle$ may be negative; as such, practical implementations always safeguard the update of η_k. To allow the acceptance of the Barzilai-Borwein step size as frequently as possible, a small positive value $\eta_{\min}$ and a large one $\eta_{\max}$ are fixed and it is only when σ_k^{-1} falls outside this range that a backup formula is activated; popular choices include

$$\eta_k = \max\{\eta_{k-1}, \eta\}, \qquad \eta = \begin{cases} 1, & \|\nabla f(x_k)\| > 1, \\ \|\nabla f(x_k)\|^{-1}, & \gamma \leqslant \|\nabla f(x_k)\| \leqslant 1, \\ \gamma^{-1}, & \|\nabla f(x_k)\| < \gamma, \end{cases} \tag{1.20}$$

$(0 < \gamma \ll 1)$ or simply $\eta_k = \eta$ alone. This means that when the spectral step size is rejected (for being either too short, too long, or negative), the decision is to make essentially a steepest descent step by reusing the previous step length or by prescribing a new one.

The generalization to the constrained case of the Barzilai-Borwein idea is very simple: one takes the iteration (1.16)-(1.17) and supplies it with the spectral step size $\eta_k = \sigma_k^{-1}$ in the tangent direction $\tau_k = -P_k \nabla f(x_k)$, properly safeguarded with some formula resembling (1.20), but with appropriate choices of s_{k-1} and y_{k-1}. Clearly, in regard of Theorem 1.2.4, σ_k should now be embedded with second order information, not from the

[2]Equality constraints are accounted for upon being rewritten as a pair of inequality constraints.

objective f, but from the lagrangian L. Hence, in the first place, a multiplier estimate λ_k is needed; for that purpose, the linear system (1.13) is solved. Observe that one can get rid of (for instance) the linear system related to the direction τ_k; in fact, a simple calculation promptly gives

$$\tau_k = -\nabla f(x_k) - \nabla g(x_k)\lambda_k - \xi_k \nu_k. \tag{1.21}$$

Only the linear systems associated with the multiplier λ_k and the normal direction ν_k remain, which are independent of each other and can thus be solved in parallel.

So the spectral parameter σ_k should now be a Rayleigh quotient for an average hessian of the lagrangian, obviously *at a tangent vector*. Looking at (1.19), a choice that almost suggests itself consists of replacing s_{k-1} by $\eta_{k-1}\tau_{k-1}$ and $\nabla^2 f$ by $\nabla_x^2 L$, that is:

$$y_{k-1} = \nabla_x L(x_{k-1} + \eta_{k-1}\tau_{k-1}, \lambda_k) - \nabla_x L(x_{k-1}, \lambda_k).$$

Observe that, for every $k \in \mathbb{N}_0$, being P_k the matrix of the orthogonal projection onto $\mathcal{T}_k = \mathrm{Ker}(Jg(x_k)) = \mathrm{Ran}(P_k)$, the identity $\mathrm{Ker}(P_k) = \mathrm{Ran}(\nabla g(x_k))$ holds and therefore, for any $\lambda \in \mathbb{R}^m$,

$$\tau_k = -P_k \nabla f(x_k) = -P_k[\nabla f(x_k) + \nabla g(x_k)\lambda] = -P_k \nabla_x L(x_k, \lambda); \tag{1.22}$$

hence, because P_{k-1} is a symmetric matrix, the dot product $\langle s_{k-1}, y_{k-1}\rangle$ is the same if in y_{k-1} the term $\nabla_x L(x_{k-1}, \lambda_k)$ is replaced by $\nabla f(x_{k-1})$, *i.e.*

$$s_{k-1} = \eta_{k-1}\tau_{k-1}, \qquad y_{k-1} = \nabla_x L(x_{k-1} + s_{k-1}, \lambda_k) - \nabla f(x_{k-1}). \tag{1.23}$$

But then the resulting step length is altogether based on information from the tangent subspace at the previous iterate; this is certainly not an issue near a solution x_*, but it might degrade the performance at earlier stages. It is perhaps more sensible to gather information over the tangent subspace at the current iterate. This can be easily achieved by making the obvious adjustments to (1.23): x_k in place of x_{k-1} and $s_{k-1} \in \mathrm{Ran}(P_k)$; the most handy pick is

$$s_{k-1} = \frac{\delta}{\|\tau_k\|}\tau_k, \qquad y_{k-1} = \nabla_x L(x_k + s_{k-1}, \lambda_k) - \nabla f(x_k), \tag{1.24}$$

where δ is a small positive value in order for y_{k-1} to measure the change in $\nabla_x L$ near x_k.

Though (1.23) and (1.24) produce true Rayleigh quotients, the resulting algorithms suffer from the same shortcoming: the need to evaluate gradients at two points per iteration. In this regard they greatly differ from the original methods, which need derivative computations just at one point per iteration – a feature one would like to carry over to the new framework. This discussion motivates the following choice:

$$s_{k-1} = P_k(x_k - x_{k-1}), \qquad y_{k-1} = \nabla f(x_k) - \nabla_x L(x_{k-1}, \lambda_k). \tag{1.25}$$

Note that near a solution holds $x_k - x_{k-1} \approx s_{k-1} \in \mathrm{Ran}(P_k)$, giving rise to a spectral coefficient σ_k that is roughly a Rayleigh quotient, at a tangent vector, for an average hessian of the lagrangian (as intended). At first it might seem that this option is not computationally more efficient than the previous ones, since while not requiring gradient evaluations at additional points beyond x_{k-1} and x_k, it involves the solution of a linear system to determine s_{k-1}. However, the coefficient matrix of that system is exactly the same appearing in the systems related with λ_k and ν_k (yet again, one should bear in mind that these three linear systems are independent of one another).

1.4.2 Remark. As already mentioned, in any of the resulting algorithms corresponding to the alternatives (1.23), (1.24) and (1.25), the several linear systems involved in an iteration (two for the first versions, three in the case of the last one) all share the same symmetric positive definite coefficient matrix. Therefore, when a Cholesky factorization is affordable there may be no need to parallelize at all.

If the problem's dimension is such that iterative solvers have to be employed (given the structure of the coefficient matrix, the best option is a conjugate gradient method, eventually preconditioned), then there is no choice but to go parallel. Notice that in this case one can use simple warm-start strategies to speed up the process, like giving as initial approximation the solution computed at the previous iteration (*e.g.* λ_{k-1}, ν_{k-1} and s_{k-2} while solving for λ_k, ν_k and s_{k-1}, respectively). The observation immediately after (1.25) suggests also for that choice the suspension of the linear algebra needed to compute $s_{k-1} = P_k(x_k - x_{k-1})$, taking simply $s_{k-1} = x_k - x_{k-1}$ instead, once the iterates reach the surroundings of a stationary point (which can be recognized through the norm of $\nabla_x L(x_k, \lambda_k)$, or equivalently through the norm of τ_k). ■

A sketch of the algorithm now follows: version (1.25) is preferred and it is assumed that a Cholesky factorization of $Jg\nabla g$ is affordable (the adjustments if that is not the case are foreseeable). The first iterate is generated with Algorithm 1.3.2.

1.4.3 Algorithm.
INPUT: initial guess x_0, initial step size $\eta_0 > 0$, tolerances $0 < \varepsilon < \gamma$,
step length bounds $0 < \eta_{\min} < \eta_{\max}$, maximum number of iterations N.
OUTPUT: approximate solution x, or message of failure.
Step 1 Solve $Jg(x_0)\nabla g(x_0)\lambda_0 = \eta_0^{-1} g(x_0) - Jg(x_0)\nabla f(x_0)$.
Step 2 Set $x_1 = x_0 - \eta_0[\nabla f(x_0) + \nabla g(x_0)\lambda_0]$.
Step 3 Set $k = 2$.
Step 4 While $k \leqslant N$ do Steps 5–13.
Step 5 Compute the Cholesky factorization LL^T of $Jg(x_1)\nabla g(x_1)$.
Step 6 Solve using forward-backward substitution: $LL^T v = g(x_1)$,
$LL^T w = Jg(x_1)(x_1 - x_0)$ and $LL^T \lambda = \xi g(x_1) - Jg(x_1)\nabla f(x_1)$, $\xi \geqslant 0$.
Step 7 Set $\nu = -\nabla g(x_1)v$ and $\tau = -\nabla f(x_1) - \nabla g(x_1)\lambda - \xi\nu$.
Step 8 Set $s = x_1 - x_0 - \nabla g(x_1)w$ and $y = \nabla f(x_1) - \nabla f(x_0) - \nabla g(x_0)\lambda$.
Step 9 Set $\eta = \langle s, s\rangle / \langle s, y\rangle$.
Step 10 If $\eta \notin [\eta_{\min}, \eta_{\max}]$ then

$$\eta = \begin{cases} 1, & \|\tau\| > 1, \\ \|\tau\|^{-1}, & \gamma \leqslant \|\tau\| \leqslant 1, \\ \gamma^{-1}, & \|\tau\| < \gamma. \end{cases}$$

Step 11 Set $x = x_1 + \eta\tau + \nu$.
Step 12 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 13 Set $x_0 = x_1$, $x_1 = x$ and $k = k + 1$.
Step 14 OUTPUT('The method failed after N iterations.');
STOP.

When iterative solving is implemented (using parallel computation), Steps 5 and 6 amount to a single step.

1.4.4 Remark. Nonmonotone methods are a relatively recent development in optimization theory. They were triggered in the early eighties by means of *watchdog techniques* [27] and nonmonotone line search rules [13]. Possibly their most striking feature, when properly designed, is the ability to avoid the so-called *Maratos effect* [5, Sec. 15.5]; this is a phenomenon that plagues monotonic methods and it consists of good local steps being rejected, thus preventing the algorithm of making fast progress towards a solution. As a matter of fact, the Maratos effect alone can, for instance, deprive Newton's method of Q-quadratic convergence [18, Sec. 15.3].

Having played a crucial role in the generalization of spectral gradient methods to nonlinear problems, the GLL line search strategy (see page 11) has nevertheless been recognized to be undesirably conservative in some cases (for ill-conditioned problems the performance depends heavily on the parameter M). Accordingly, alternative nonmonotone rules have been sugested, most notably by Dai and Zhang [28], who designed an adaptive strategy independent of M, by Grippo and Sciandrone [29], who relaxed monotonicity even further by combining both nonmonotone watchdog and line search techniques, and more recently by Shi and Wang [30].

It should be stressed that, although global convergence cannot in theory be guaranteed without some kind of the aforementioned procedures, spectral gradient methods are often able to function soundly without them; in fact, the performance of the pure method (eventually safeguarded) is often better [16, 31] – this was actually the main reason behind the trend towards monotonicity relaxation over the years. So yet another interesting feature of spectral methods emerges, and one that can have a significant impact on computational costs[3]. For instance, it is very common in practical applications to have objective and/or constraint functions that depend on the parameters implicitly through a *state variable* (the temperature, the displacement, etc.) satisfying some partial differential equation (see Chapter 5); then each test performed in, say, a line search, will require a finite element solution of that equation! ■

1.4.5 Remark. Since its introduction, the Barzilai-Borwein method has been the subject of several upgrades over the years.

The need of preconditioning for ill-conditioned problems has been aknowledged and adressed in several works: for instance Luengo *et al.* [32], Bello and Raydan [33], Chehab and Raydan [34]. The latter paper is particularly interesting, since the authors develop a problem-independent preconditioning theory (this is welcome because problem-dependent preconditioners require a deep knowledge – which is not always available – of the underlying problem). Numerical tests are also performed, with encouraging results.

Alternative spectral coeficients have also been proposed. Dai, Yuan and Yuan [35] observed that the classical Barzilai-Borwein methods could also be obtained from an interpolation point of view and proposed two spectral-like step sizes which gave rise to a couple of competitive gradient methods, that on several examples superseded well established spectral gradient methods. *Alternating step size* strategies have also been investigated [29, 35]; as the designation suggests, the step length can be generated by different

[3] One more reason why watchdog, line search or trust-region methodologies are not insisted upon.

formulas at different iterations: for instance, it could alternate between any of the choices (1.18), depending on some criteria being fullfield. But maybe one of the most interesting proposals of all came from Dai *et al.* [36], who introduced a method for general nonlinear unconstrained minimization in which the Barzilai-Borwein step length is reused during $\ell \in \mathbb{N}$ consecutive iterations. A local R-linear convergence result is proved and, more interesting, numerical evidence indicates that when $\ell > n/2 \geqslant 3$, this *cyclic Barzilai-Borwein method* is locally R-superlinearly convergent! ■

1.5 Final remarks

The methods introduced in the previous section basically emulate (over the tangent space) the Barzilai-Borwein method for unconstrained minimization. Therefore, Remarks 1.4.4 and 1.4.5 abound with material likely of being extended to the constrained case.

An interesting example would be the one concerning a cyclic procedure, with a safeguard along the lines of (1.20). Recall that the increment is defined as $\Delta_k = \eta_k \tau_k + \nu_k$ and that, when the multiplier λ_k is computed through (1.13), according to (1.21) one has $\tau_k = -\nabla_x L(x_k, \lambda_k) - \xi_k \nu_k$; hence, *if* $\xi_k > 0$,

$$-\eta_k \nabla_x L(x_k, \lambda_k) = \eta_k \tau_k + \eta_k \xi_k \nu_k$$

is "almost" the increment Δ_k; one cannot take $\xi_k = \eta_k^{-1}$, because the computation of the multiplier precedes the one of the step length, but the choice $\xi_k = \eta_{k-1}^{-1}$ is possible, thus yielding exactly Δ_k in $\ell - 1$ iterations of the ℓ ones where η_k is the same. This completely exempts the computation of the increment from the linear algebra associated with ν_k, leaving a mere matrix-vector product to evaluate.

Regarding convergence theory, it is a topic that is currently motive for research. Global convergence results for spectral methods always depend on some sort of nonmonotone scheme; but it is not completely clear, for instance, how a nonmonotone (tangent) line search would go in the constrained setting, mainly because the presented algorithms only produce feasible iterates later in the optimization process. The known theoretical results [15, 21] do not seem easy to replicate even adopting a direct generalization of the classic GLL strategy (see page 11), *i.e.* given $M \in \mathbb{N}$ and $\gamma \in]0, 1[$, find $\eta_k > 0$ (starting with $\eta_k = \sigma_k^{-1}$) such that

$$f(x_k + \eta_k \tau_k) \leqslant \max_{0 \leqslant j \leqslant m_k} f(x_{k-j}) + \gamma \eta_k \langle \nabla f(x_k), \tau_k \rangle, \quad m_k = \min\{k, M-1\}.$$

Besides, the above strategy is admittedly dubious unless the previous M iterates are nearly feasible. One must also consider that, given the motivation behind the spectral coefficient σ_k in the constrained framework, possibly the function subject to tangent decrease should be, not f, but most likely the "lagrangian" $x \mapsto L(x, \lambda_k)$,[4] although when feasibility is approximately reached there may not be much of a difference. As for a local convergence result, again, what is known for unconstrained problems [37] does not appear to be readily (if at all) extendable to the constrained case.

[4] At first glance it may seem a bit awkward to decrease this function along a search direction related with f; however, recall that (1.22) is in particular valid for $\lambda = \lambda_k$.

To close the section, it is called to attention the fact that several characteristics of Algorithm 1.4.3 are in stark contrast with quasi-Newton methods for constrained minimization. The latter are somewhat heavier and they pose several technical issues – see, for instance, the book of Bonnans *et al.* [18, Sec. 16.3]. Furthermore, as it will be seen in the next chapter, the extension to inequality constrained problems in the context of active-set methods will be quite straightforward.

Chapter 2

Active-set methods for inequality constrained problems

In this chapter, a generalization of the gradient methods introduced in the previous one is proposed. The aim is to deal with inequality constraints through an active-set strategy. Again, feasibility issues are not concerned and the iterates are allowed to violate the constraints (an important exception to this rule is discussed in Section 2.3).

Section 2.1 introduces some generalities on the subject. In Section 2.2 a generalization of the steepest descent approach (see Section 1.3) is presented. Section 2.3 delineates a simple procedure to keep the iterates feasible with respect to *bounds* on the variables, a recurrent type of inequality constraint appearing in applications. The extension of the spectral gradient approach (see Section 1.4) is the subject of Section 2.4. Both methods are further generalized in Section 2.5 to handle (finite and semi-infinite) minimax problems and in Section 2.6 some numerical tests are performed.

The notations used in the previous chapter, as well as the regularity assumptions made, carry over to the present one.

2.1 Introductory notes

The problem under consideration is now

$$\min_{x\in\mathcal{C}} f(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n : \ g(x) \leqslant 0\}, \tag{2.1}$$

the inequality being understood componentwise: $\mathcal{C} = \{x \in \mathbb{R}^n : \ g^i(x) \leqslant 0,\ 1 \leqslant i \leqslant m\}$. For simplicity, only inequality constraints are taken; the inclusion of equality constraints is straightforward, as it will become clear.

2.1.1 Note. The most popular algorithms for general inequality constrained minimization fall probably under one of two categories: *active-set sequential quadratic programming methods* or *interior point methods*. The latter, as suggested by their very own designation, are especially tailored to operate within the feasibility paradigm.

Despite the several (and severe) technical challenges posed by both approaches, they have been considered the most powerful ones in nonlinear programming. However, none

of them will be pursued in the text, so the interested reader is referred, for instance, to the book of Nocedal and Wright [5, Chaps. 18,19] for details. ■

Local solutions $x_* \in \mathcal{C}$ of problem (2.1) usually lie on a level set determined by some (or eventually all) of the m constraint functions involved:

$$\mathcal{C}_* = \{x \in \mathbb{R}^n : \ g^i(x) = 0, \ i \in \mathcal{A}_*\}, \qquad \mathcal{A}_* \subset \{1, 2, \ldots, m\},$$

where $\mathcal{A}_*$ is the *set of active indices* at x_*. The constraints $g^i(x_*) \leqslant 0$ that are satisfied with an equality are called *active constraints* at x_*, as opposed to the *inactive* ones which verify the strict inequality $g^i(x_*) < 0$ (this terminology extends to any $x \in \mathcal{C}$).

The concept of an active constraint is key, as most clearly the remaining (inactive) constraints have no say with respect to the characterization of local solutions of (2.1). Obviously, if some $\mathcal{C}_*$ were known in advance, a solution could be found through an algorithm for equality constrained minimization. These considerations suggest that the problem could be regarded as having solely equality constraints, provided that the necessary adjustments are made to account for the selection of the active constraints (this is the philosophy adopted in the sequel).

Like in the algorithms of the previous chapter, the iterates are not necessarily bound to the feasible set $\mathcal{C}$; in view of this fact, a broader notion of active constraint must be employed: during the optimization process, an inequality constraint is deemed active precisely when it is violated (actually there is a bit more to it, as that inequality will be kept active as long as the associated "Lagrange multiplier" has a specific sign). The strategy will be a basic one regarding active-set methods: given an iterate $x_k \in \mathbb{R}^n$, the constraints are partitioned into those inequalities to be treated as active at x_k, which determine the so-called *working surface*

$$\mathcal{W}_k = \{x \in \mathbb{R}^n : \ g^i(x) = 0, \ i \in \mathcal{A}_k\}, \qquad \mathcal{A}_k \subset \{1, 2, \ldots, m\}, \tag{2.2}$$

and the ones to be treated as inactive (which are essentially ignored); then, the move from x_k to x_{k+1} is made via a single iteration of an algorithm applicable to $\min_{x \in \mathcal{W}_k} f(x)$.

2.1.2 Remark. The necessary optimality conditions for problem (2.1) are better expressed in terms of the active constraints at a solution $x^* \in \mathcal{C}$. The KKT conditions (see Theorem 1.2.3) can in this case be written as follows:

$$\begin{cases} \nabla f(x_*) + \sum_{i \in \mathcal{A}_*} \lambda_*^i \nabla g^i(x_*) = 0, \\ g_i(x_*) = 0, \quad i \in \mathcal{A}_*, \\ g^i(x_*) < 0, \quad i \notin \mathcal{A}_*, \\ \lambda_*^i \geqslant 0, \qquad i \in \mathcal{A}_*, \\ \lambda_*^i = 0, \qquad i \notin \mathcal{A}_*. \end{cases}$$

The first two equations are simply the optimality conditions for the equality constrained problem obtained considering only the active constraints at x_*, that is, for the problem $\min_{x \in \mathcal{C}_*} f(x)$. The third condition ensures that inactive constraints are satisfied and the last one specifies that they have null Lagrange multipliers attached. The fourth condition is most important for practical purposes: *Lagrange multipliers associated with active constraints must be nonnegative*; this fact lies at the core of informed decision-making regarding deactivation criteria along the iterations of an algorithm. ■

2.2 Extension of the steepest descent approach

As can be understood from the discussion around (2.2), to define $\mathcal{W}_k$ is equivalent to prescribe the set $\mathcal{A}_k$ of active indices at x_k. This is accomplished in two stages.

First there is a "prediction phase", where is checked if any index $i \notin \mathcal{A}_{k-1}$ verifies $g^i(x_k) > 0$; those who do, will join the previous active set $\mathcal{A}_{k-1}$ to form $\mathcal{A}_k$ (in other words, for the time being, $\mathcal{A}_k$ is just a set of eligible indices). A "correction phase" follows, where to each constraint $g^i \leqslant 0$, $i \in \mathcal{A}_k$, a multiplier λ_k^i is associated, computed analogously to (1.12); then, the constraints with attached negative multipliers are "filtered", according to some criterion, and $\mathcal{A}_k$ is updated (these steps are repeated until one is left with either no active indices at all, or with a set of indices i for which $\lambda_k^i \geqslant 0$).

The minimization algorithm

In Step 7 of the following scheme, the index associated with the smallest negative multiplier is deactivated, a strategy frequently recommended in the literature [5, Sec. 16.5]. Some further comments on this choice are made ahead, but for now one settles for it.

2.2.1 Algorithm.
INPUT: initial guess x_0, step size $\eta > 0$, tolerance $\varepsilon > 0$,
maximum number of iterations N.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$ and $\mathcal{A} = \varnothing$ (no active constraints).
Step 2 While $k \leqslant N$ do Steps 3–10.
Step 3 Set $\mathcal{I} = \{1, 2, \ldots, m\} \setminus \mathcal{A}$.
Step 4 For $i \in \mathcal{I}$ do
If $g^i(x_0) > 0$ then set $\mathcal{A} = \mathcal{A} \cup \{i\}$; (Constraint $g^i \leqslant 0$ is set active.)
Step 5 Solve $\sum_{j\in\mathcal{A}} \langle \nabla g^i(x_0), \nabla g^j(x_0)\rangle \lambda^j = \eta^{-1} g^i(x_0) - \langle \nabla g^i(x_0), \nabla f(x_0)\rangle$, $i \in \mathcal{A}$.
Step 6 Set $i = \arg\min_{j\in\mathcal{A}} \lambda^j$.
Step 7 If $\lambda^i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Constraint $g^i \leqslant 0$ is set inactive.)
GOTO Step 5.
Step 8 Set $x = x_0 - \eta \left[\nabla f(x_0) + \sum_{i\in\mathcal{A}} \lambda^i \nabla g^i(x_0)\right]$.
Step 9 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 10 Set $x_0 = x$ and $k = k + 1$.
Step 11 OUTPUT('The method failed after N iterations.');
STOP.

2.2.2 Note. Observe that, although not explicitly stated, the algorithm must be halted once the number of active constraints equals or tops the number of variables. ■

2.2.3 Remark. For large scale problems, the routine comprising Steps 5 to 7 can become unbearably heavy if many constraints are active, and thus many multipliers must be computed at each cycle (it is stressed that, upon a constraint's deactivation, the remaining multipliers are "fake" and must indeed be computed anew). In this regard, it should be pointed out that Algorithm 2.2.1 *does not depend* on the specific procedure here adopted to establish the active set. Other ways of prescribing the active constraints can be implemented without prejudice. A common one [5, Sec. 18.5], especially in sequential quadratic

programming frameworks, does it by means of an auxiliary linear programming problem, usually solved efficiently by the simplex method [5, Chap. 13] even if a very large number of variables is involved. ■

Successful active-set algorithms depend a great deal on the method chosen to solve equality constrained problems, as well as, of course, on the ability to identify the correct active constraints. In general, convergence cannot be guaranteed and *zigzagging*[1] can sometimes occur, although experience shows it to be a rare phenomenon.

On deactivation criteria[2]

Steps 6 and 7 of Algorithm 2.2.1 are quite clear if only one of the multipliers λ^i becomes negative. But if several multipliers become negative simultaneously, it seems that a somewhat arbitrary choice was made: to deactivate the constraint associated to the "most" negative multiplier. Such an option is usually supported by the sensitivity analysis of the Lagrange multipliers at a solution x_* [17, Sec. 11.7]. However, decisions about adding and dropping constraints have to be made long before a stationary point is even found on the current working surface; the possibility is thus open for alternative choices, like deactivating at once *all* the constraints with negative multipliers, or to choose the one(s) to be deactivated on the basis of some other criterion.

To put things in more precise terms, assume that only two indices, say 1 and 2, are eligible to become active. Consider the (most popular in the literature) least-squares multipliers λ^1 and λ^2, *i.e.* solution of (1.13) with $\xi_k = 0$:

$$\begin{cases} \langle \nabla g^1, \nabla g^1 \rangle \lambda^1 + \langle \nabla g^1, \nabla g^2 \rangle \lambda^2 = -\langle \nabla g^1, \nabla f \rangle, \\ \langle \nabla g^1, \nabla g^2 \rangle \lambda^1 + \langle \nabla g^2, \nabla g^2 \rangle \lambda^2 = -\langle \nabla g^2, \nabla f \rangle. \end{cases}$$

Consider also the least-squares multipliers μ^1 and μ^2,

$$\langle \nabla g^1, \nabla g^1 \rangle \mu^1 = -\langle \nabla g^1, \nabla f \rangle, \qquad \langle \nabla g^2, \nabla g^2 \rangle \mu^2 = -\langle \nabla g^2, \nabla f \rangle,$$

obtained if only $g^1 \leqslant 0$ or $g^2 \leqslant 0$ stays active, respectively; obviously they reflect the tendency exhibited by one constraint in face of the other one's deactivation. All four multipliers can be related through the linear system

$$\begin{cases} \langle \nabla g^1, \nabla g^1 \rangle \lambda^1 + \langle \nabla g^1, \nabla g^2 \rangle \lambda^2 = \langle \nabla g^1, \nabla g^1 \rangle \mu^1, \\ \langle \nabla g^1, \nabla g^2 \rangle \lambda^1 + \langle \nabla g^1, \nabla g^2 \rangle \lambda^2 = \langle \nabla g^2, \nabla g^2 \rangle \mu^2. \end{cases} \tag{2.3}$$

Now assume that λ^1 and λ^2 are negative, so that both constraints are candidates for deactivation; suppose the choice falls (for instance) on $g^2 \leqslant 0$ because, according to the "most negative multiplier criterion", $\lambda^2 < \lambda^1$. One would like this to prevent $\mu^1 < 0$ and $\mu^2 \geqslant 0$ from happening simultaneously (otherwise it means that probably a poor choice was made), but simple numerical tests show that *it does not*. However, it is not difficult to see that if $\lambda^2 \|\nabla g^2\| < \lambda^1 \|\nabla g^1\|$ is satisfied instead, the undesirable situation is avoided. Effectively, solving (2.3) for λ^1 and λ^2, that inequality is easily proven equivalent to

$$\mu^2(\langle \nabla g^1, \nabla g^2 \rangle + \|\nabla g^1\| \|\nabla g^2\|)\|\nabla g^2\| < \mu^1(\langle \nabla g^1, \nabla g^2 \rangle + \|\nabla g^1\| \|\nabla g^2\|)\|\nabla g^1\|;$$

[1] The set of active constraints changes many times.

[2] The discussion takes place at an iterate x_k which, together with any subscripts, is omitted for simplicity.

hence, by the Cauchy-Schwarz inequality, it is impossible to have $\mu^1 < 0$ and $\mu^2 \geqslant 0$.

This simple example shows that the "most negative multiplier criterion" is not "optimal" and that there is room for improvement in this aspect.[3] Unfortunately, the kind of analysis just performed for two constraints becomes excruciating and highly confusing in the general case. A more systematic approach is yet to be devised.

One might think that the situation is simpler if only one multiplier, say λ^1, is negative. In this case, can one safely deactivate the constraint $g^1 \leqslant 0$, trusting that the new multiplier μ^2 is nonnegative? The answer is no (hence the cyclic deletion procedure adopted in Algorithm 2.2.1), as can be easily verified toying with some numerical examples.

2.2.4 Remark. There may be another way to look at this whole issue. Supposing the algorithm runs "smoothly enough", that is, with sufficiently small step lengths, no dramatic changes will occur in the quantities defining the step (the gradient of f and the gradients of the g^i, with i an active index). Therefore, it is not expected that more than one constraint will become inactive at a given iteration; moreover, if this happens (more than one multiplier becomes negative) it should be taken as a warning sign: the algorithm is taking too large "strides" and the step length should be reduced. ■

2.3 Adding bounds on the variables

There is a particular case when many constraints are likely to become active, but they have the simple form $a^i - x^i \leqslant 0$ or $x^i - b^i \leqslant 0$, that is: the vector variable x is confined to a "rectangular box" $\mathcal{R} = \prod_{i=1}^{n} \left[a^i, b^i\right]$ in $\mathbb{R}^n$, the cases $a^i = -\infty$ and/or $b^i = +\infty$ being not excluded to account for variables where no lower bounds and/or no upper bounds are imposed. This particular type of constraints can be treated without passing through the (cumbersome) process of solving the corresponding set of linear equations in order to obtain the respective multipliers. It suffices to ignore these particular constraints in Step 5 of Algorithm 2.2.1, computing however the remaining multipliers (if any) as usual, then compute the new point according to Step 8 of that same algorithm, and finally "crop" each coordinate towards the range $\left[a^i, b^i\right]$. The latter operation can be viewed as a (trivial) projection onto the cartesian product $\prod_{i=1}^{n} \left[a^i, b^i\right]$. The tricky detail is that the "blocked" variables x^i should be ignored when computing the scalar products between gradients in Step 5; that is, those "blocked" variables must be treated as if they were no longer variables but mere parameters, equal to a^i or to b^i. This approach is equivalent to solving the full system of linear equations (some comments on this matter are made further ahead), with the obvious advantage of alleviating the computational burden by reducing the size of the linear system. The procedure has an additional advantage: since it is based on a (trivial) projection operation, it is able to deal with *essential* constraints of the form $x^i \geqslant a^i$ and/or $x^i \leqslant b^i$, which are very common in applications (for instance, they can mirror technological restrictions).

2.3.1 Remark. The KKT optimality conditions for problem (2.1), in the presence of bounds, can be more concisely expressed by considering the set $\mathcal{B}_* \subset \{1, 2, \ldots, n\}$ of

[3] Notice also that the criterion involving the multiplier times the gradient's norm, of the respective constraint function, has the advantage of being invariant under constraint rescalings.

indices corresponding to active bounds $a^i - x^i \leqslant 0$ or $x^i - b^i \leqslant 0$ at a solution $x_* \in \mathcal{R} \cap \mathcal{C}$:

$$\begin{cases} f_{,i}(x_*) + \sum_{j \in \mathcal{A}_*} \lambda^j g^j_{,i}(x_*) = 0, & i \notin \mathcal{B}_*, \\ f_{,i}(x_*) + \sum_{j \in \mathcal{A}_*} \lambda^j g^j_{,i}(x_*) \geqslant 0, & i \in \mathcal{B}_* : \ x^i_* = a^i, \\ f_{,i}(x_*) + \sum_{j \in \mathcal{A}_*} \lambda^j g^j_{,i}(x_*) \leqslant 0, & i \in \mathcal{B}_* : \ x^i_* = b^i, \end{cases} \qquad \begin{cases} g_j(x_*) = 0, & j \in \mathcal{A}_*, \\ g^j(x_*) < 0, & j \notin \mathcal{A}_*, \\ \lambda^j_* \geqslant 0, & j \in \mathcal{A}_*, \\ \lambda^j_* = 0, & j \notin \mathcal{A}_*. \end{cases}$$

The computations involved in the discussion following Algorithm 2.3.2 shed some light on how the first set of conditions turns up. Of course, it is assumed that all the gradients of active constraints at x_* (bounds and otherwise) are linearly independent. ■

The upgrade from Algorithm 2.2.1 is achieved by making the aforementioned changes. It is useful to consider, regarding the variables x^i, the set $\mathcal{F} \subset \{1, 2, \ldots, n\}$ of indices corresponding to "free" (or "unblocked") variables, and to incidentally introduce the notation $\langle x, y \rangle_{\mathcal{F}} = \sum_{k \in \mathcal{F}} x^k y^k$ for given $x, y \in \mathbb{R}^n$.

2.3.2 Algorithm.
INPUT: initial guess $x_0 \in \prod_{i=1}^n \left]a^i, b^i\right[$, step size $\eta > 0$, tolerance $\varepsilon > 0$,
maximum number of iterations N.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$, $\mathcal{F} = \{1, 2, \ldots, n\}$ and $\mathcal{A} = \varnothing$ (no active constraints).
Step 2 While $k \leqslant N$ do Steps 3–11.
Step 3 Set $\mathcal{I} = \{1, 2, \ldots, m\} \setminus \mathcal{A}$.
Step 4 For $i \in \mathcal{I}$ do
If $g^i(x_0) > 0$ then set $\mathcal{A} = \mathcal{A} \cup \{i\}$; (Constraint $g^i \leqslant 0$ is set active.)
Step 5 Solve $\sum_{j \in \mathcal{A}} \langle \nabla g^i(x_0), \nabla g^j(x_0) \rangle_{\mathcal{F}} \lambda^j = \eta^{-1} g^i(x_0) - \langle \nabla g^i(x_0), \nabla f(x_0) \rangle_{\mathcal{F}}$, $i \in \mathcal{A}$.
Step 6 Set $i = \arg\min_{j \in \mathcal{A}} \lambda^j$.
Step 7 If $\lambda^i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Constraint $g^i \leqslant 0$ is set inactive.)
GOTO Step 5.
Step 8 Set $x = x_0 - \eta \left[\nabla f(x_0) + \sum_{i \in \mathcal{A}} \lambda^i \nabla g^i(x_0)\right]$.
Step 9 For $i = 1, \ldots, n$ do
If $x^i < a^i$ then set $x^i = a^i$ and $\mathcal{F} = \mathcal{F} \setminus \{i\}$;
If $x^i > b^i$ then set $x^i = b^i$ and $\mathcal{F} = \mathcal{F} \setminus \{i\}$.
Step 10 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 11 Set $x_0 = x$ and $k = k + 1$.
Step 12 OUTPUT('The method failed after N iterations.');
STOP.

2.3.3 Note. A similar statement to that of Note 2.2.2 applies: the algorithm is now halted when the number of active constraints equals or tops the number of *free* variables. ■

To ilustrate the "equivalence" between Algorithms 2.2.1 and 2.3.2 in handling bounds, one takes x_0 in the boundary of $\prod_{i=1}^n \left[a^i, b^i\right]$ (with no loss of generality, the problem will be supposed to have upper bounds only). The relative complement of $\mathcal{F}$ in $\{1, 2, \ldots, n\}$, call it $\mathcal{B}$, collecting the indices of blocked variables is introduced, along with the notation $\langle x, y \rangle_{\mathcal{B}} = \sum_{k \in \mathcal{B}} x^k y^k$ for given $x, y \in \mathbb{R}^n$.

So Algorithm 2.2.1 iterates according to

$$x^i = \begin{cases} b^i - \eta\left(f_{,i} + \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j + \ell^i\right), & i \in \mathcal{B}, \\ x^i_0 - \eta\left(f_{,i} + \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j\right), & i \in \mathcal{F}, \end{cases} \tag{2.4}$$

where the multipliers λ^i (associated with active inequalities $g^i \leqslant 0$) and ℓ^i (associated with active bounds $x^i - b^i \leqslant 0$) are determined by

$$\begin{cases} \displaystyle\sum_{j\in\mathcal{A}} \langle \nabla g^i, \nabla g^j\rangle \lambda^j + \sum_{k\in\mathcal{B}} \langle \nabla g^i, e_k\rangle \ell^k = \eta^{-1} g^i - \langle \nabla g^i, \nabla f\rangle, & i \in \mathcal{A}, \\ \displaystyle\sum_{j\in\mathcal{A}} \langle e_i, \nabla g^j\rangle \lambda^j + \sum_{k\in\mathcal{B}} \langle e_i, e_k\rangle \ell^k = -\langle e_i, \nabla f\rangle, & i \in \mathcal{B}. \end{cases} \tag{2.5}$$

From the second lot of equations it follows imediately that $\ell^i = -f_{,i} - \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j$ and, consequently, $x^i = b^i$ for all $i \in \mathcal{B}$. Furthermore, since

$$\begin{aligned} \sum_{k\in\mathcal{B}} \langle \nabla g^i, e_k\rangle \ell^k &= \sum_{k\in\mathcal{B}} g^i_{,k}\ell^k = \sum_{k\in\mathcal{B}} g^i_{,k}\left(-f_{,k} - \sum_{j\in\mathcal{A}} g^j_{,k}\lambda^j\right) \\ &= -\sum_{k\in\mathcal{B}} g^i_{,k} f_{,k} - \sum_{j\in\mathcal{A}} \left(\sum_{k\in\mathcal{B}} g^i_{,k} g^j_{,k}\right)\lambda^j = -\langle \nabla g^i, \nabla f\rangle_{\mathcal{B}} - \sum_{j\in\mathcal{A}} \langle \nabla g^i, \nabla g^j\rangle_{\mathcal{B}} \lambda^j, \end{aligned}$$

the first lot of equations can be rewritten as

$$\sum_{j\in\mathcal{A}} \left(\langle \nabla g^i, \nabla g^j\rangle - \langle \nabla g^i, \nabla g^j\rangle_{\mathcal{B}}\right)\lambda^j = \eta^{-1} g^i - \langle \nabla g^i, \nabla f\rangle + \langle \nabla g^i, \nabla f\rangle_{\mathcal{B}}, \quad i \in \mathcal{A},$$

that is,

$$\sum_{j\in\mathcal{A}} \langle \nabla g^i, \nabla g^j\rangle_{\mathcal{F}} \lambda^j = \eta^{-1} g^i - \langle \nabla g^i, \nabla f\rangle_{\mathcal{F}}, \quad i \in \mathcal{A}.$$

It is precisely the prior linear system of equations that determines the multipliers λ^i in Algorithm 2.3.2, which iterates as

$$x^i = \min\{\hat{x}^i, b^i\}, \quad 1 \leqslant i \leqslant n, \qquad \hat{x}^i = \begin{cases} b^i - \eta\left(f_{,i} + \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j\right), & i \in \mathcal{B}, \\ x^i_0 - \eta\left(f_{,i} + \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j\right), & i \in \mathcal{F}. \end{cases} \tag{2.6}$$

Because the λ^i are exactly the same ones yielded by (2.5), this means that all components x^i with $i \in \mathcal{F}$ are equal in both (2.4) and (2.6) if $\hat{x}^i \leqslant b^i$ for every $i \in \mathcal{F}$. As for the other components: having the inequalities $x^i - b^i \leqslant 0$ with $i \in \mathcal{B}$ been considered active in Algorithm 2.2.1, then $\ell^i = -f_{,i} - \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j \geqslant 0$; hence, $\hat{x}^i \geqslant b^i$ for all $i \in \mathcal{B}$ and the projection will ensure that $x^i = b^i$ holds for each $i \in \mathcal{B}$, just like in (2.4).

2.3.4 Remark. The bound constraints are automatically managed by Algorithm 2.3.2 and no additional criteria is required: activation takes place during the projection phase in Step 9; deactivation occurs naturally in Step 8 once any of the increment's components corresponding to a blocked index has the appropriate sign (recall that, for $i \in \mathcal{B}$, $-f_{,i} - \sum_{j\in\mathcal{A}} g^j_{,i}\lambda^j$ is precisely the multiplier ℓ^i associated with the bound $x^i \leqslant b^i$ in Algorithm 2.2.1). Note that when $\mathcal{A} = \varnothing$ and $\mathcal{B} \neq \varnothing$, Algorithm 2.3.2 will act like a classical gradient method with projection [7, Sec. 8.6] (and like steepest descent if $\mathcal{A} = \mathcal{B} = \varnothing$). ■

To sum up, given $\eta > 0$ and $x_0 \in \prod_{i=1}^n \left]a^i, b^i\right[$, the iteration of the modified algorithm takes the following form:

$$x_{k+1}^i = \max\left\{a^i, \min\left\{\hat{x}_{k+1}^i, b^i\right\}\right\},\ 1 \leqslant i \leqslant n, \qquad \hat{x}_{k+1} = x_k - \eta\left[\nabla f(x_k) + \nabla g(x_k)\lambda_k\right],$$

the multipliers λ_k being determined through

$$J_{\mathcal{F}} g_k(x_k)\nabla_{\mathcal{F}} g_k(x_k)\lambda_k = \eta^{-1} g_k(x_k) - J_{\mathcal{F}} g_k(x_k)\nabla_{\mathcal{F}} f(x_k),$$

where g_k is the vector function with components g^i, $i \in \mathcal{A}_k$, and the subscript $\mathcal{F}$ means that the derivatives are taken with respect to the current free variables.

It is also noted that, by inserting the solution λ_k of the prior linear system in the step $\delta_k = -\nabla f(x_k) - \nabla g(x_k)\lambda_k$ and upon defining

$$\begin{cases} \hat{\tau}_k = -\nabla f(x_k) + \nabla g_k(x_k)\left[J_{\mathcal{F}} g_k(x_k)\nabla_{\mathcal{F}} g_k(x_k)\right]^{-1} J_{\mathcal{F}} g_k(x_k)\nabla_{\mathcal{F}} f(x_k), \\ \hat{\nu}_k = -\nabla g_k(x_k)\left[J_{\mathcal{F}} g_k(x_k)\nabla_{\mathcal{F}} g_k(x_k)\right]^{-1} g_k(x_k), \end{cases} \tag{2.7}$$

the increment $\Delta_k = \eta\delta_k$ can be written equivalently $\Delta_k = \eta\hat{\tau}_k + \hat{\nu}_k$.

2.4 Extension of the spectral gradient approach

The generalization of Algorithm 1.4.3, like that of Algorithm 1.3.2, is a very straightforward one. After prescribing the active set $\mathcal{A}_k$, the iteration is obtained from (1.16)-(1.17) replacing g by g_k, the vector function whose components are the g^i with $i \in \mathcal{A}_k$:

$$x_{k+1} = x_k + \eta_k\tau_k + \nu_k, \qquad \tau_k = -P_k\nabla f(x_k), \quad \nu_k = -K_k g_k(x_k), \tag{2.8}$$

where

$$K_k = \nabla g_k(x_k)\left[Jg_k(x_k)\nabla g_k(x_k)\right]^{-1}, \qquad P_k = I - K_k Jg_k(x_k), \tag{2.9}$$

and $\eta_k = \langle s_{k-1}, s_{k-1}\rangle\langle s_{k-1}, y_{k-1}\rangle^{-1}$ is an appropriate spectral step length. By introducing the "lagrangian" $L_k(x,\lambda) = f(x) + \langle g_k(x), \lambda\rangle = f(x) + \sum_{i\in\mathcal{A}_k}\lambda^i g^i(x)$, choices (1.24) and (1.25) are accordingly modified to

$$s_{k-1} = \frac{\delta}{\|\tau_k\|}\tau_k, \qquad y_{k-1} = \nabla_x L_k(x_k + s_{k-1}, \lambda_k) - \nabla f(x_k), \tag{2.10}$$

and

$$s_{k-1} = P_k(x_k - x_{k-1}), \qquad y_{k-1} = \nabla f(x_k) - \nabla_x L_k(x_{k-1}, \lambda_k), \tag{2.11}$$

where, similarly to (1.13), λ_k is computed through

$$Jg_k(x_k)\nabla g_k(x_k)\lambda_k = \xi_k g_k(x_k) - Jg_k(x_k)\nabla f(x_k). \qquad (\xi_k \geqslant 0) \tag{2.12}$$

An option resembling (1.23), which would now include information coming from a possibly outdated set of constraints, seems blatantly inadequate and is altogether discarded.

2.4.1 Remark. Common sense suggests (2.10) when $\mathcal{A}_k \neq \mathcal{A}_{k-1}$ and (2.11) if $\mathcal{A}_k = \mathcal{A}_{k-1}$. Still, in the former case, since an additional linearization is required, it may be tempting to go with (2.11) just the same, even though it is now a somewhat artificial choice. Observe, however, that an indiscriminate use of (2.11) increases the storage requirements of the algorithm: *all* the gradients $\nabla g^1(x_{k-1}), \ldots, \nabla g^m(x_{k-1})$ must be available, and not just $\nabla g^i(x_{k-1})$ with $i \in \mathcal{A}_{k-1}$, since the active constraints at x_k are not known before hand. Yet, for large scale problems, this may be preferable to the computational effort in (2.10) of evaluating $\nabla g^i(x_{k-1} + s_{k-1})$ for all $i \in \mathcal{A}_k$. ■

Bound constraints can be handled implicitly by taking the directions $\hat{\tau}_k$ and $\hat{\nu}_k$ defined in (2.7) and performing the iteration

$$x_{k+1}^i = \max\left\{a^i, \min\left\{\hat{x}_{k+1}^i, b^i\right\}\right\},\ 1 \leqslant i \leqslant n, \qquad \hat{x}_{k+1} = x_k + \eta_k \hat{\tau}_k + \hat{\nu}_k,$$

with a spectral step length η_k "based solely on free variables". More precisely: the components of s_{k-1} and y_{k-1} corresponding to blocked variables are null and those associated with free variables, depending on the adopted strategy, are given by

$$s_{k-1}^{\mathcal{F}} = \frac{\delta}{\|\hat{\tau}_k^{\mathcal{F}}\|}\hat{\tau}_k^{\mathcal{F}}, \qquad y_{k-1}^{\mathcal{F}} = \nabla_{\mathcal{F}} L_k(x_k + s_{k-1}, \lambda_k) - \nabla_{\mathcal{F}} f(x_k),$$

or by

$$s_{k-1}^{\mathcal{F}} = P_k(x_k^{\mathcal{F}} - x_{k-1}^{\mathcal{F}}), \qquad y_{k-1}^{\mathcal{F}} = \nabla_{\mathcal{F}} f(x_k) - \nabla_{\mathcal{F}} L_k(x_{k-1}, \lambda_k),$$

where $P_k = I - \nabla_{\mathcal{F}} g_k(x_k)\left[J_{\mathcal{F}} g_k(x_k) \nabla_{\mathcal{F}} g_k(x_k)\right]^{-1} J_{\mathcal{F}} g_k(x_k)$ and the superscript $\mathcal{F}$ signalizes the "subvector" whose components correspond to free variables; the multipliers vector λ_k is the solution of the linear system

$$J_{\mathcal{F}} g_k(x_k) \nabla_{\mathcal{F}} g_k(x_k) \lambda_k = \xi_k g_k(x_k) - J_{\mathcal{F}} g_k(x_k) \nabla_{\mathcal{F}} f(x_k), \qquad (\xi_k \geqslant 0)$$

in which case $\hat{\tau}_k = -\nabla f(x_k) - \nabla g_k(x_k)\lambda_k - \xi_k \hat{\nu}_k$. It is obvious that relations (2.8) to (2.12) are recovered when all the variables are free.

In the next scheme it is assumed that Choleski factorizations are affordable (if not, the linear systems would have to be solved via some iterative method, the best candidate being the method of conjugate gradients). To keep the notation concise, g will denote the vector function $(g^i)_{i \in \mathcal{A}}$ for the most recent update $\mathcal{A}$ of the active set.

2.4.2 Algorithm.
INPUT: initial guess $x_0 \in \prod_{i=1}^n \left]a^i, b^i\right[$, initial step size $\eta_0 > 0$, tolerances $0 < \varepsilon < \gamma$,
step length bounds $0 < \eta_{\min} < \eta_{\max}$, maximum number of iterations N.
OUTPUT: approximate solution x, or message of failure.
Step 1 Set $k = 1$, $\mathcal{F} = \{1, 2, \ldots, n\}$ and $\mathcal{A} = \varnothing$ (no active constraints).
Step 2 Obtain x_1 by performing one iteration of Algorithm 2.3.2.
Step 3 Set $k = 2$ and $\mathcal{A}_0 = \mathcal{A}$.
Step 4 While $k \leqslant N$ do Steps 5–14.
  Step 5 Set $\mathcal{I} = \{1, 2, \ldots, m\} \setminus \mathcal{A}$.
  Step 6 For $i \in \mathcal{I}$ do
    If $g^i(x_1) > 0$ then set $\mathcal{A} = \mathcal{A} \cup \{i\}$; (Constraint $g^i \leqslant 0$ is set active.)
  Step 7 If $\mathcal{A} = \varnothing$ then

set $\tau = -\nabla f(x_1)$ and $\nu = 0$;
set $s = x_1^{\mathcal{F}} - x_0^{\mathcal{F}}$ and $y = \nabla_{\mathcal{F}} f(x_1) - \nabla_{\mathcal{F}} f(x_0)$.

Step 8 If $\mathcal{A} \neq \varnothing$ then do Steps 8.1–8.9.

Step 8.1 Compute the Choleski factorization LL^T of $J_{\mathcal{F}} g(x_1) \nabla_{\mathcal{F}} g(x_1)$.

Step 8.2 Solve $LL^T \lambda = \xi g(x_1) - J_{\mathcal{F}} g(x_1) \nabla_{\mathcal{F}} f(x_1)$, for some $\xi \geqslant 0$.

Step 8.3 Set $i = \arg\min_{j \in \mathcal{A}} \lambda^j$.

Step 8.4 If $\lambda^i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Constraint $g^i \leqslant 0$ is set inactive.)
GOTO Step 8.1.

Step 8.5 Solve $LL^T v = g(x_1)$.

Step 8.6 Set $\nu = -\nabla g(x_1) v$ and $\tau = -\nabla f(x_1) - \nabla g(x_1) \lambda - \xi \nu$.

Step 8.7 If $\mathcal{A} = \mathcal{A}_0$ then
solve $LL^T w = J_{\mathcal{F}} g(x_1)(x_1^{\mathcal{F}} - x_0^{\mathcal{F}})$;
set $s = (x_1^{\mathcal{F}} - x_0^{\mathcal{F}}) - \nabla_{\mathcal{F}} g(x_1) w$;
set $y = \nabla_{\mathcal{F}} f(x_1) - \nabla_{\mathcal{F}} f(x_0) - \nabla_{\mathcal{F}} g(x_0) \lambda$.

Step 8.9 If $\mathcal{A} \neq \mathcal{A}_0$ then
set $s = \delta \|\tau^{\mathcal{F}}\|^{-1} \tau^{\mathcal{F}}$, for some $0 < \delta \ll 1$;
set $x_2 = x_1$ and $x_2^{\mathcal{F}} = x_1^{\mathcal{F}} + s$;
set $y = \nabla_{\mathcal{F}} f(x_2) + \nabla_{\mathcal{F}} g(x_2) \lambda - \nabla_{\mathcal{F}} f(x_1)$.

Step 9 Set $\eta = \langle s, s \rangle / \langle s, y \rangle$.

Step 10 If $\eta \notin [\eta_{\min}, \eta_{\max}]$ then

$$\eta = \begin{cases} 1, & \|\tau\| > 1, \\ \|\tau\|^{-1}, & \gamma \leqslant \|\tau\| \leqslant 1, \\ \gamma^{-1}, & \|\tau\| < \gamma. \end{cases}$$

Step 11 Set $x = x_1 + \eta\tau + \nu$.

Step 12 For $i = 1, \ldots, n$ do
If $x^i < a^i$ then set $x^i = a^i$ and $\mathcal{F} = \mathcal{F} \setminus \{i\}$;
If $x^i > b^i$ then set $x^i = b^i$ and $\mathcal{F} = \mathcal{F} \setminus \{i\}$.

Step 13 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.

Step 14 Set $x_0 = x_1$, $x_1 = x$, $\mathcal{A}_0 = \mathcal{A}$ and $k = k + 1$.

Step 15 OUTPUT('The method failed after N iterations.');
STOP.

Note that when $\mathcal{A} = \varnothing$, if $\mathcal{F} = \{1, 2, \ldots, n\}$ the algorithm acts like a classical Barzilai-Borwein method [8, Sec. 2], and if $\mathcal{F} \subsetneq \{1, 2, \ldots, n\}$ it behaves somewhat like the spectral projected gradient method (minus the line search) of Birgin *et al.* [24, Sec. 2].

2.4.3 Remark. It is again reminded that Steps 8.1 to 8.4 can be replaced by some other procedure to determine the active constraints (see Remark 2.2.3).

The numerous observations about the spectral methods defined in Section 1.4 remain equally pertinent in the present framework. For instance, the statements in Remark 1.4.2 about iterative solvers still apply, keeping in mind that warm-start strategies have now to be exercised with care. Ditto for the contents of Remarks 1.4.4 and 1.4.5. ■

2.5 Applications to worst-case optimization

In engineering design, among other fields of application, it is often convenient (or even mandatory) to express the problem in a way that, faced with several plausible scenarios, the effects of a worst-case scenario event are minimized [38,39] (think of a high tower subject to the action of the wind, whose intensity and direction can obviously assume multiple configurations). Such features can frequently be accomodated by a formulation like

$$\min_{x\in\mathbb{R}^n} F(x), \qquad F(x) = \max\{f_1(x), f_2(x), \ldots, f_m(x)\}, \tag{2.13}$$

termed *finite (or discrete) minimax* due to the finite number of variables and of objectives, or more generally by

$$\min_{x\in\mathbb{R}^n} F(x), \qquad F(x) = \max_{y\in Y} f(x,y), \quad Y \subset \mathbb{R}^p \text{ compact}, \tag{2.14}$$

termed *semi-infinite (or continuous) minimax* because of the finite number of variables but infinite number of objectives $f_y(x) = f(x,y)$, $y \in Y$.

These are examples of problems implicating functions which are not everywhere differentiable. In this case, instead of gradients, *subgradients* are considered to "emulate" usual methodology from the field of smooth optimization (worsening of technical issues notwithstanding). Currently, *bundle methods* [40] are aknowledged as the most reliable and effective globally convergent algorithms for nonsmooth optimization. They only assume some mild regularity property on the functions involved, typically (local) lipschitz continuity. However, minimax formulations are prone to alternative points of view.

Problems of the type (2.13) and (2.14) will be adressed in the sequel, where it is supposed that all the $f_i : \mathbb{R}^n \to \mathbb{R}$ and $f : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R}$ are at least continuously differentiable functions (despite this, of course, the objective function F is usually nonsmooth). Again, no constraints are considered purely for simplicity, as their handling in either case will soon became clear.

Since the multiple functions involved are smooth, it is tempting to try and bypass the nonsmoothness of F in some way. This is the goal presiding the present section.

Finite minimax problems

It is frequent in the literature, with the addition of an artificial variable $z \in \mathbb{R}$, to transform problem (2.13) into an inequality constrained one,

$$\min_{(x,z)\in\mathcal{C}} z, \qquad \mathcal{C} = \{(x,z) \in \mathbb{R}^{n+1} : \ f_i(x) - z \leqslant 0, \ 1 \leqslant i \leqslant m\}, \tag{2.15}$$

from which the following optimality conditions can be obtained:

$$\begin{cases} \sum_{i\in\mathcal{O}_*} \lambda_*^i \nabla f_i(x_*) = 0, \\ f_i(x_*) = F(x_*), \quad i \in \mathcal{O}_*, \\ f_i(x_*) < F(x_*), \quad i \notin \mathcal{O}_*, \end{cases} \qquad \begin{cases} \lambda_*^i \geqslant 0, \quad i \in \mathcal{O}_*, \\ \lambda_*^i = 0, \quad i \notin \mathcal{O}_*, \\ \sum_{i\in\mathcal{O}_*} \lambda_*^i = 1, \end{cases}$$

where $\mathcal{O}_*$ is the set of indices of *active objective functions* at x_*.

The equivalence between (2.13) and (2.15) has often been exploited to avert non-smoothness issues [41–48]. This can also be achieved by producing a smooth approximation of F. *Smoothing* techniques have also enjoyed special attention in the literature [49–57]. Both strategies have in common the goal of approaching (2.13) in such a way that usual methodology from smooth optimization can be applied. The plan introduced in the sequel trails the same train of thought, but in a more direct way.

Departing from $x_0 \in \mathbb{R}^n$, suppose it is f_2 the maximal function between all the f_i; then one starts minimizing f_2, keeping an eye on the inequalities $f_i \leqslant f_2$ for $i \neq 2$, and continues doing so while f_2 remains maximal. This can be done, for instance, with Algorithm 2.2.1, $\mathcal{A} = \varnothing$ for the time being. If at some iterate x_k a different function becomes maximal, say f_5 (meaning the inequality $f_5 \leqslant f_2$ was violated), then one proceeds with the minimization of f_5 but, and here is the key point, *not necessarily dropping* f_2. More exactly, f_5 takes the place of f_2 as the objective function and the previous inequalities are replaced by new ones, $f_i \leqslant f_5$ for $i \neq 5$, of which $f_2 \leqslant f_5$ is considered *active*; on other words, unless this latter inequality gets deactivated in the meanwhile, one is now in fact minimizing f_2 and f_5 simultaneously. Recall that, once activated, an inequality becomes inactive only when the respective multiplier is negative (and not merely when it ceases to be violated, thus preventing the procedure of becoming an utter chaotic mess). The process continues in the same fashion thereafter until (hopefully) convergence is observed.

2.5.1 Algorithm.
INPUT: initial guess x_0, step size $\eta > 0$, tolerance $\varepsilon > 0$,
maximum number of iterations N.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$ and $\mathcal{O} = \varnothing$ (no active functions).
Step 2 While $k \leqslant N$ do Steps 3–11.
Step 3 Set $l = \arg\max_{1 \leqslant i \leqslant m} f_i(x_0)$.
Step 4 Set $\mathcal{O} = \mathcal{O} \cup \{l\}$ (function f_l is set active) and $\mathcal{A} = \mathcal{O} \setminus \{l\}$.
Step 5 For $i \in \mathcal{A}$ do
$g^i(x_0) = f_i(x_0) - f_l(x_0)$ and $\nabla g^i(x_0) = \nabla f_i(x_0) - \nabla f_l(x_0)$.
Step 6 Solve $\sum_{j \in \mathcal{A}} \langle \nabla g^i(x_0), \nabla g^j(x_0) \rangle \lambda^j = \eta^{-1} g^i(x_0) - \langle \nabla g^i(x_0), \nabla f_l(x_0) \rangle$, $i \in \mathcal{A}$.
Step 7 Set $i = \arg\min_{j \in \mathcal{A}} \lambda^j$.
Step 8 If $\lambda^i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Function f_i is set inactive.)
GOTO Step 6.
Step 9 Set $x = x_0 - \eta \left[\nabla f_l(x_0) + \sum_{i \in \mathcal{A}} \lambda^i \nabla g^i(x_0) \right]$.
Step 10 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 11 Set $\mathcal{O} = \mathcal{A} \cup \{l\}$, $x_0 = x$ and $k = k + 1$.
Step 12 OUTPUT('The method failed after N iterations.');
STOP.

The aforementioned strategy is a sort of "sequential nonlinear programming" approach to minimax problems. This can be better visualized if instead of constantly changing the objective function and the constraints, one keeps them fixed as long as possible. To be more precise, suppose that $f_1(x_0) = F(x_0)$; then, one starts by considering the problem

$$\min_{x \in \mathcal{C}} f_1(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n : \ g(x) \leqslant 0\}, \tag{2.16}$$

where $g : \mathbb{R}^n \to \mathbb{R}^{m-1}$ is defined by $g^i(x) = f_i(x) - f_1(x)$ for all $i \neq 1$. Next, Algorithm 2.3.2 or 2.4.2 is applied to (2.16), *the only difference lying on the activation criterion:* $g^i \leqslant 0$ is activated when the function f_i becomes maximal. It is only when the inequality $g^j \leqslant 0$ becomes inactive, j being the index of the current maximal function, that a change occurs. For instance, assume that at some stage $\mathcal{A}_{k-1} = \{3, 5, 6\}$ and $\mathcal{A}_k = \{5, 6\}$ (which means that $g^3 \leqslant 0$ has just been deactivated), and suppose one has precisely $f_3(x_k) = F(x_k)$; then a new problem is considered:

$$\min_{x \in \mathcal{C}} f_3(x), \qquad \mathcal{C} = \{x \in \mathbb{R}^n : \ g(x) \leqslant 0\}, \tag{2.17}$$

with $g : \mathbb{R}^n \to \mathbb{R}^{m-1}$ defined by $g^i(x) = f_i(x) - f_3(x)$ for all $i \neq 3$, of which $g^5 \leqslant 0$ and $g^6 \leqslant 0$ are currently active (f_1 has been "wiped off the picture"). The algorithm is now applied to (2.17) and so on and so forth.

2.5.2 Remark. The course of action proposed is obviously extendable to other active-set methods, since the generated "sequence of problems" is open to other techniques in nonlinear optimization (*e.g.* sequential quadratic programming).

Notice also that other problems can be adressed, as several nonsmooth functions can be reformulated as max-functions, a trivial example being $F(x) = |f(x)| = \max\{-f(x), f(x)\}$. However, this may prove to be unacceptably expensive at times because the number of objective functions can grow exponentially, for instance if $F(x) = \sum_{1 \leqslant i \leqslant m} |f_i(x)|$. ■

Semi-infinite minimax problems

Besides the already mentioned *bundle methods*, several ways of adressing continuous minimax problems can be found across the literature. See, for instance, the book of Rustem and Howe [38, Chaps. 2–4]. A particularly attractive proposal comes in the form of *discretization methods* [58, Chap. 3]: an approximation of (2.14) is obtained replacing Y by a discrete subset; the resulting *finite* minimax problem is then solved via a suitable algorithm. This process is iterated using increasingly finer discretizations of Y when establishing convergence theory for such methods [58, Secs. 3.4–3.6] [59, Sec. 5], a procedure which is in general forbidding from the numerical point of view. Therefore, in practice, approximate solutions to (2.14) are computed by solving a single discretized problem. The main drawback, adressed more substantially than ever before in a recent paper by Royset and Pee [60], is well summarized in the words of the authors:

> *"The apparent simplicity of discretization algorithms hides a fundamental trade-off between the level of discretization of Y and the computational work required to approximately solve the resulting finite minimax problem. One would typically require a fine discretization of Y to guarantee that the finite minimax problem approximates (2.14), in some sense, with high accuracy. However, in that case, the finite minimax problem becomes large scale (in the number of functions to maximize over) and the computational work to solve it may be high. A coarser discretization saves in the solution time of the correspondingly smaller finite minimax problem at the expense of a poorer approximation of (2.14). It is often difficult, in practice, to construct discretizations of Y that balance this trade-off effectively."*

In fact, if the dimension p is sufficiently large, there may be no way at all to acceptably discretize Y. Discretization methods are thus applied mainly when p is small: Y is a time interval, a range of temperatures, *et cetera*. The approach proposed below, although relying likewise on a subset $Y_N \subset Y$ of finite cardinality $N \in \mathbb{N}$, is able to deal with larger values of p. The "mesh" Y_N is not used to replace (2.14) by a finite minimax formulation; its sole purpose is to provide starting points for the maximization, over Y, of the map $y \mapsto f(x, y)$.[4] The process goes as follows.

Take some inital guess $x_0 \in \mathbb{R}^n$ and find $y_0 = \arg\max_{y \in Y_N} f(x_0, y)$; it is likely that in the surroundings of y_0 a true (hopefully global) maximizer of $y \mapsto f(x_0, y)$ exists; then an algorithm is employed to maximize this function, departing from y_0, until a stationary point $y_1 = y_1(x_0) \in Y$ is computed. One has just found its first objective function: $f_1(x) = f(x, y_1)$. Next, departing from x_0, perform one (*e.g.* steepest descent) step to minimize f_1, in this way obtaining the next iterate x_1. Note that y_1 needs updating: starting from $y_1(x_0)$, maximize $y \mapsto f(x_1, y)$ to get the new $y_1 = y_1(x_1) \in Y$.

Now run through all $y \in Y_N$ to find $y_0 = \arg\max_{y \in Y_N} f(x_1, y)$; if $f(x_1, y_0) > f_1(x_1)$, a new candidate has been found and as before, starting from y_0, one maximizes $y \mapsto f(x_1, y)$ to obtain a stationary point $y_2 = y_2(x_1) \in Y$; here is the second objective function: $f_2(x) = f(x, y_2)$. Once multiple objectives arise, the minimization steps are performed via some algorithm for discrete minimax problems (*e.g.* the one described earlier).

In general, at an iterate x_k one will have objectives $f_1, f_2, \ldots, f_m$ indexed, respectively, by a supposed maximizer $y_1(x_{k-1}), y_2(x_{k-1}), \ldots, y_m(x_{k-1})$ of $y \mapsto f(x_{k-1}, y)$. The first stage consists of determining the updates $y_i(x_k)$ through maximization of $y \mapsto f(x_k, y)$, taking $y_0 = y_i(x_{k-1})$ as the initial guess (be aware that these m tasks are independent from one another and can therefore be parallelized). The second stage evaluates whether new maximizers arise by checking if $\max\{f(x_k, y) : y \in Y_N\} > \max\{f_i(x_k) : 1 \leqslant i \leqslant m\}$; if so, a new objective function $f_{m+1}(x) = f(x, y_{m+1})$ is considered, $y_{m+1} = y_{m+1}(x_k) \in Y$. In a third and last stage, a single iteration of a discrete minimax algorithm is applied to the problem $\min_{x \in \mathbb{R}^n} \max\{f_1(x), \ldots, f_{m+1}(x)\}$. It is during this phase that, eventually, some of the $y_i(x_k)$ cease to qualify as maximizers, meaning some objective functions are deactivated – see Step 8 of Algorithm 2.5.1. When this happens, one judges to be a waste of time keep tracking those past maximizers and truly *deletes* them from the lot, trusting they will be caught again in the "mesh" if justified.

2.5.3 Algorithm.
INPUT: initial guess x_0, step size $\eta > 0$, tolerance $\varepsilon > 0$,
"grid" $Y_N = \{p_1, \ldots, p_N\}$, maximum number of iterations N_{max}.
OUTPUT: approximate solution x or message of failure.
Step 1 Set $k = 1$ and $\mathcal{O} = \varnothing$ (no maximizers).
Step 2 While $k \leqslant N_{max}$ do Steps 3–14.
Step 3 For $i \in \mathcal{O}$ do
$y_i = \arg\max_{y \in Y} f(x_0, y)$; (Update the maximizers.)
$f_i(x_0) = f(x_0, y_i)$ and $\nabla f_i(x_0) = \nabla_x f(x_0, y_i)$.
Step 4 Set $f_N = \max_{1 \leqslant i \leqslant N} f(x_0, p_i)$.
Step 5 If $k = 1$ then set $f_{max} = -\infty$; else set $f_{max} = \max_{i \in \mathcal{O}} f_i(x_0)$.

[4] This maximization procedure is somewhat related to the so-called *direct search methods* [61], though here it is based on derivatives.

Step 6 If $f_N > f_{max}$ then
set $\mathcal{O} = \mathcal{O} \cup \{k\}$;
set $y_k = \arg\max_{y \in Y} f(x_0, y)$; (A new maximizer.)
set $f_k(x_0) = f(x_0, y_k)$ and $\nabla f_k(x_0) = \nabla_x f(x_0, y_k)$.
Step 7 Set $l = \arg\max_{i \in \mathcal{O}} f_i(x_0)$ and $\mathcal{A} = \mathcal{O} \setminus \{l\}$.
Step 8 For $i \in \mathcal{A}$ do
$g^i(x_0) = f_i(x_0) - f_l(x_0)$ and $\nabla g^i(x_0) = \nabla f_i(x_0) - \nabla f_l(x_0)$.
Step 9 Solve $\sum_{j \in \mathcal{A}} \langle \nabla g^i(x_0), \nabla g^j(x_0) \rangle \lambda^j = \eta^{-1} g^i(x_0) - \langle \nabla g^i(x_0), \nabla f_l(x_0) \rangle$, $i \in \mathcal{A}$.
Step 10 Set $i = \arg\min_{j \in \mathcal{A}} \lambda^j$.
Step 11 If $\lambda^i < 0$ then set $\mathcal{A} = \mathcal{A} \setminus \{i\}$; (Maximizer y_i is deleted.)
GOTO Step 9.
Step 12 Set $x = x_0 - \eta \left[\nabla f_l(x_0) + \sum_{i \in \mathcal{A}} \lambda^i \nabla g^i(x_0) \right]$.
Step 13 If $\|x - x_0\| < \varepsilon$ then OUTPUT(x);
STOP.
Step 14 Set $\mathcal{O} = \mathcal{A} \cup \{l\}$, $x_0 = x$ and $k = k + 1$.
Step 15 OUTPUT('The method failed after N iterations.');
STOP.

2.5.4 Remark. The level of discretization remains an issue. The use of a coarse "mesh" Y_N increases the possibility of missing a good starting point to find the global maximizer of $y \mapsto f(x_k, y)$ over Y, though one might argue that this side effect may be dimmed by the maximization phase itself. In this respect, the algorithm has a better chance of withstanding "unfortunate" choices of Y_N than the more common discretization methods.

On the other hand, too fine a discretization will burden the algorithm with a large number of evaluations of $y \mapsto f(x_k, y)$ at each iteration, although this is by far a less serious handicap than the one afflicting usual discretization algorithms under similar circumstances (recall the previous quote). It can also easily give rise to the *same* maximizer being caught more than once.

It is not hard to imagine the latter case, where at least two of the $y_i(x_k)$ become "equal", occurring naturally as a result of the optimization process as well (a situation depicted in Figure 2.1). A judgement on whether to supress some maximizers could be

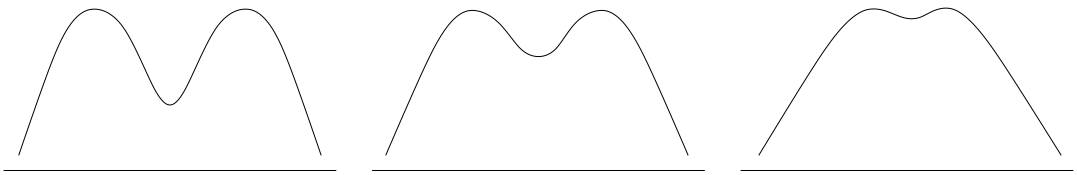

Figure 2.1: Merging maximizers.

made, for example, by testing $\|y_i(x_k) - y_j(x_k)\|$ against a prescribed tolerance based on the "diameter" of Y_N. But there is another case worth considering: while undergoing minimization, some maximizers may exhibit a tendency to "vanish" (as suggested in Figure 2.2). The handling of such situations should probably rely on a slope-based criterion.

These (and possibly other) aspects point towards a "filtering" of maximizers, or more accurately: to the adoption of additional deactivation criteria beyond those already contained in the finite minimax algorithm employed. ■

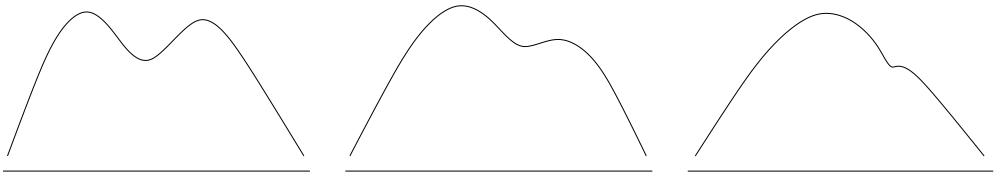

Figure 2.2: A fading maximizer.

2.6 Numerical tests

The following examples, though standard in the literature, are admittedly small scale and serve merely to corroborate the methodology developed in these first chapters. To be given a chance of competing in difficult problems with existing specialized software, the discussed algorithms will require a thoughtful implementation including (among other aspects): parallel computing, the adoption of strategies to deal with ill-conditioning issues and proper line search rules.

Smooth problems

For convenience, the variables in the following descriptions will be denoted with subscripts. The initial guess is denoted by x_0 and the objective's optimal value (or more accurately, the best known value) is denoted by f_*.

Example 1 [62, Prob. 56]

$f(x) = -x_1x_2x_3, \quad x \in \mathbb{R}^7,$
$g^1(x) = x_1 - 4.2\sin^2(x_4) = 0,$
$g^2(x) = x_2 - 4.2\sin^2(x_5) = 0,$
$g^3(x) = x_3 - 4.2\sin^2(x_6) = 0,$
$g^4(x) = x_1 + 2x_2 + 2x_3 - 7.2\sin^2(x_7) = 0,$
$x_0 = (0.4, 2.4, 2.3, 0.1, 1.5, 1.5, 0.4),$
$f_* = -3.456.$

Example 2 [62, Prob. 64]

$f(x) = 5x_1 + 50000/x_1 + 20x_2 + 72000/x_2 + 10x_3 + 144000/x_3, \quad x \in \mathbb{R}^3,$
$g^1(x) = 4/x_1 + 32/x_2 + 120/x_3 - 1 \leqslant 0,$
$x_i \geqslant 10^{-5}, \quad i = 1, 2, 3,$
$x_0 = (10, 8, 12),$
$f_* = 6299.842428.$

Example 3 [62, Prob. 71]

$f(x) = x_1x_4(x_1 + x_2 + x_3) + x_3, \quad x \in \mathbb{R}^4,$
$g^1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 - 40 = 0,$
$g^2(x) = 25 - x_1x_2x_3x_4 \leqslant 0,$
$1 \leqslant x_i \leqslant 5, \quad i = 1, 2, 3, 4,$

$x_0 = (2.4, 2.3, 2.1, 2.4)$,
$f_* = 17.0140173$.

Example 4 [62, Prob. 77]

$f(x) = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6, \quad x \in \mathbb{R}^5,$
$g^1(x) = {x_1}^2 x_4 + \sin(x_4 - x_5) - 2\sqrt{2} = 0,$
$g^2(x) = x_2 + {x_3}^4 {x_4}^2 - 8 - \sqrt{2} = 0,$
$x_0 = (2.2, 2.3, 2.1, 2.1, 2.2)$,
$f_* = 0.24150513$.

Example 5 [62, Prob. 78]

$f(x) = x_1 x_2 x_3 x_4 x_5, \quad x \in \mathbb{R}^5,$
$g^1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 + {x_5}^2 - 10 = 0,$
$g^2(x) = x_2 x_3 - 5 x_4 x_5 = 0,$
$g^3(x) = {x_1}^3 + {x_2}^3 + 1 = 0,$
$x_0 = (-4, 3, 4, -3, -4)$,
$f_* = -2.91970041$.

Example 6 [62, Prob. 81]

$f(x) = \exp(x_1 x_2 x_3 x_4 x_5) - 0.5\left({x_1}^3 + {x_2}^3 + 1\right)^2, \quad x \in \mathbb{R}^5,$
$g^1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 + {x_5}^2 - 10 = 0,$
$g^2(x) = x_2 x_3 - 5 x_4 x_5 = 0,$
$g^3(x) = {x_1}^3 + {x_2}^3 + 1 = 0,$
$-2.3 \leqslant x_i \leqslant 2.3, \quad i = 1, 2,$
$-3.2 \leqslant x_i \leqslant 3.2, \quad i = 3, 4, 5,$
$x_0 = (-0.1, 2.2, 3.1, -1.5, 2)$,
$f_* = 0.0539498478$.

Example 7 [62, Prob. 100]

$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + {x_3}^4 + 3(x_4 - 11)^2 +$
$\quad 10{x_5}^6 + 7{x_6}^2 + {x_7}^4 - 4x_6 x_7 - 10x_6 - 8x_7, \quad x \in \mathbb{R}^7,$
$g^1(x) = 2{x_1}^2 + 3{x_2}^4 + x_3 + 4{x_4}^2 + 5x_5 - 127 \leqslant 0,$
$g^2(x) = 7x_1 + 3x_2 + 10{x_3}^2 + x_4 - x_5 - 282 \leqslant 0,$
$g^3(x) = 23x_1 + {x_2}^2 + 6{x_6}^2 - 8x_7 - 196 \leqslant 0,$
$g^4(x) = 4{x_1}^2 + {x_2}^2 - 3x_1 x_2 + 2{x_3}^2 + 5x_6 - 11x_7 \leqslant 0,$
$x_0 = (1, 2, 0, 4, 0, 1, 1)$,
$f_* = 680.6300573$.

Example 8 [62, Prob. 113]

$f(x) = {x_1}^2 + {x_2}^2 + x_1 x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 +$
$\quad 2(x_6 - 1)^2 + 5{x_7}^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \quad x \in \mathbb{R}^{10},$
$g^1(x) = 4x_1 + 5x_2 - 3x_7 + 9x_8 - 105 \leqslant 0,$

$g^2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leqslant 0,$
$g^3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leqslant 0,$
$g^4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2{x_3}^2 - 7x_4 - 120 \leqslant 0,$
$g^5(x) = 5{x_1}^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leqslant 0,$
$g^6(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3{x_5}^2 - x_6 - 30 \leqslant 0,$
$g^7(x) = {x_1}^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leqslant 0,$
$g^8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leqslant 0,$
$x_0 = (12, 12, -2, 15, -9, 12, -8, 20, -3, 18),$
$f_* = 24.3062091.$

Numerical results

The previous examples were tested with both Algorithm 2.3.2 and 2.4.2, using a prescribed tolerance $\varepsilon = 10^{-5}$. In Algorithm 2.4.2 it was also set $\eta_{\min} = 10^{-10}$ and $\eta_{\max} = 10^{10}$; the value of δ taken in Step 8.9 varies, but tipically $\delta \in \{10^{-1}, 10^{-2}, 10^{-3}\}$; the backup step size $\eta = \|\tau\|^{-1}$ was preferred in Step 10, as it generally yielded better results.

Ex./Alg.	k	$\|f(x_k) - f_*\|$	$\|\nabla_{\mathcal{F}} L_k(x_k)\|_2$	$\|g_k(x_k)\|_2$
1/2.3.2	134	6.95765×10^{-9}	0.000102522	2.99615×10^{-12}
1/2.4.2	46	1.25002×10^{-11}	5.15917×10^{-7}	1.94915×10^{-11}
2/2.3.2	116	7.84785×10^{-8}	3.71591×10^{-7}	2.34405×10^{-16}
2/2.4.2	23	7.77245×10^{-8}	7.62577×10^{-7}	3.30561×10^{-13}
3/2.3.2	64	6.75526×10^{-9}	9.90508×10^{-5}	1.28359×10^{-10}
3/2.4.2	20	1.08911×10^{-8}	1.72894×10^{-6}	9.80891×10^{-11}
4/2.3.2	77	1.38292×10^{-9}	6.22216×10^{-5}	2.56196×10^{-11}
4/2.4.2	29	1.22061×10^{-9}	6.20603×10^{-7}	2.24908×10^{-10}
5/2.3.2	36	4.07426×10^{-7}	9.78667×10^{-5}	7.89035×10^{-11}
5/2.4.2	8	4.12986×10^{-7}	3.11388×10^{-8}	6.13538×10^{-9}
6/2.3.2	64	2.58336×10^{-10}	9.69671×10^{-6}	1.1081×10^{-10}
6/2.4.2	19	3.00706×10^{-11}	3.52191×10^{-9}	8.37372×10^{-12}
7/2.3.2	73	7.47049×10^{-8}	0.00020038	1.22189×10^{-10}
7/2.4.2	28	7.15255×10^{-8}	3.7884×10^{-5}	2.43876×10^{-9}
8/2.3.2	49	3.0732×10^{-8}	6.97803×10^{-5}	4.17926×10^{-12}
8/2.4.2	18	2.76816×10^{-8}	2.15806×10^{-5}	7.28042×10^{-9}

Table 2.1: Results for Algorithms 2.3.2 and 2.4.2.

The notation L_k stands for the "lagrangian" $x \mapsto f(x) + \langle g_k(x), \lambda_k \rangle$, where g_k is the vector function whose components are the g^i associated to equality constraints, plus the ones associated with active inequality constraints at x_k.

As in the unconstrained case, the spectral gradient version revealed a much better performance. The expectation is that, once properly implemented, it will be able to emulate

(in constrained problems) the behaviour of its counterpart for unconstrained optimization: a "cheaper" method than most, but still able to equal (and sometimes outperform, in terms of clock time) more renowned solvers. It remains to be seen.

2.6.1 Remark. A curious fact emerged during numerical trials: taking $\xi = 0$ or $\xi \neq 0$ in Step 8.2 can produce truly diverse scenarios. Recall that $\xi = 0$ corresponds to the popular least-squares multipliers; $\xi \neq 0$, especially $\xi = 1/\eta_0$ (η_0 designates the step length η at the previous iteration), resembles the choice in Step 5 of Algorithm 2.3.2 (note that $\xi = 1/\eta$ is not an option, since λ predates η). The latter choice proved superior among the two in the sense that, in some examples, it was the only one for which convergence was observed, and in cases where both choices work, the discrepancy in the number of iterations is negligeable. Moreover, some tests with "remote" initial guesses show that, often, a significantly lesser number of iterations is required with $\xi = 1/\eta_0$ than with $\xi = 0$.

Of course, one cannot draw definitive conclusions based on so few examples, but this may be symptomatic of a more general trend. The reason must lie on the additional term featuring the scaled values of the active constraints, which appears to "marry" particularly well with infeasible methods, such as the ones introduced, at least more than the usual least-squares multipliers. This aspect probably deserves further research, the main goal being that of identifying "optimal" choices of the parameter ξ. ■

Nonsmooth problems

The next problems are solved according to the strategy described in Section 2.5, relying on Algorithm 2.5.1 (implementation of a spectral gradient version is currently underway). In the last two problems, the maximization over Y is performed with the *spectral projected gradient method* of Birgin *et al.* [24, Alg. 2.2]. No criteria in the terms of Remark 2.5.4 were implemented. For convenience, the variables in the problems' description will be denoted with subscripts.

Example 9 [63, Prob. 2.5]

$F(x) = \max\{f_1(x), f_2(x), f_3(x), f_4(x)\}, \quad x \in \mathbb{R}^4,$
$f_1(x) = {x_1}^2 + {x_2}^2 + 2{x_3}^2 + {x_4}^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$
$f_2(x) = f_1(x) + 10({x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 + x_1 - x_2 + x_3 - x_4 - 8),$
$f_3(x) = f_1(x) + 10({x_1}^2 + 2{x_2}^2 + {x_3}^2 + 2{x_4}^2 - x_1 - x_4 - 10),$
$f_4(x) = f_1(x) + 10(2{x_1}^2 + 2{x_2}^2 + {x_3}^2 + 2x_1 - x_2 - x_4 - 5),$
$x_0 = (0, 0, 0, 0),$
$F_* = -44.$

Example 10 [63, Prob. 2.23]

$F(x) = \max\{f_1(x), f_2(x), \ldots, f_{10}(x)\}, \quad x \in \mathbb{R}^{11},$
$f_i(x) = \sum_{j=0}^{10} (i+j)^{-1} \exp\big([x_{j+1} - \sin(i - 1 + 2j)]^2\big), \quad 1 \leqslant i \leqslant 10,$
$x_0 = (1, 1, \ldots, 1),$
$F_* = 261.08258.$

Example 11 [63, Prob. 3.14]

$F(x) = x_1x_2x_3x_4x_5 + 10(|f_1(x)| + |f_2(x)| + |f_3(x)|), \quad x \in \mathbb{R}^5,$
$f_1(x) = {x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2 + {x_5}^2 - 10,$
$f_2(x) = x_2x_3 - 5x_4x_5,$
$f_3(x) = {x_1}^3 + {x_2}^3 + 1,$
$x_0 = (-2, 1.5, 2, -1, -1),$
$F_* = -2.9197004.$

Example 12 [63, Prob. 3.17]

$F(x) = \max\{f_1(x), f_2(x), f_3(x)\}, \quad x \in \mathbb{R}^{10},$
$f_1(x) = \sum_{i=1}^{10}(x_i - 1)^2 + 10^{-3}\sum_{i=1}^{10}\left({x_i}^2 - 0.25\right)^2,$
$f_2(x) = \sum_{i=2}^{30}\left[\sum_{j=2}^{10} x_j(j-1)(\frac{i-1}{29})^{j-2} - \left(\sum_{j=1}^{10} x_j(\frac{i-1}{29})^{j-1}\right)^2 - 1\right]^2 + {x_1}^2 + (x_2 - {x_1}^2 - 1)^2,$
$f_3(x) = \sum_{i=2}^{10}\left[100\left(x_i - {x_{i-1}}^2\right)^2 + (1 - x_i)^2\right],$
$x_0 = (-0.1, -0.1, \ldots, -0.1),$
$F_* = 9.7857721.$

Example 13 [64, Sec. 3 – Prob. 1]

$F(x) = \max\{{x_1}^2, {x_2}^2, \ldots, {x_n}^2\}, \quad x \in \mathbb{R}^n,$
$x_0 = (1, 2, \ldots, n/2, -n/2 - 1, -n/2 - 2, \ldots, -n),$
$F_* = 0.$

Example 14 [38, Sec. 5.4 – Prob. 12]

$F(x) = \max\{f(x, y) : \ y \in [-2, 2]^3\}, \quad x \in \mathbb{R}^4,$
$f(x, y) = {y_1}^2 + {y_2}^2 + {y_3}^2 + y_1\left({x_1}^2 - x_2 + x_3 - x_4 + 2\right) + y_2\left(-x_1 + 2{x_2}^2 - {x_3}^2 + 2x_4 - 10\right) +$
$\quad y_3\left(2x_1 - x_2 + 2x_3 - {x_4}^2 - 5\right) + 5\left({x_1}^2 + {x_2}^2 + {x_3}^2 + {x_4}^2\right),$
$x_0 = (10, 1, 1, 10),$
$F_* = 43.40816.$

Example 15 [38, Sec. 5.4 – Prob. 14]

$F(x) = \max\{f(x, y) : \ y \in [-2, 2]^4\}, \quad x \in \mathbb{R}^4,$
$f(x, y) = {y_1}^2 + {y_2}^2 + {y_3}^2 + {y_4}^2 + y_1\left({x_1}^2 - 2.2x_2 + x_3 - 10x_4 + 10\right) +$
$\quad y_2\left(-2x_1 + 2{x_2}^2 - {x_3}^2 + 3x_4 - 10\right) + y_3\left(2x_1 - x_2 + 6x_3 - {x_4}^2 - 5\right) +$
$\quad 5y_4\left({x_1}^2 + {x_2}^2\right) + 5\left({x_3}^2 + {x_4}^2\right),$
$x_0 = (0, 0, 0, 0),$
$F_* = 42.56435.$

Numerical results

The tolerance used in Algorithm 2.5.1 was $\varepsilon = 10^{-5}$. Example 13 is ran with $n = 100$. In Examples 14 and 15, the cell Y was discretized by considering the two endpoints and the middle point of the interval $[-2, 2]$.

In Table 2.2, $\mathcal{O}_k$ designates the set of indices of active functions at x_k; L_k stands for the "lagrangian" $x \mapsto f_i(x) + \langle g_k(x), \lambda_k \rangle$, being g_k the vector function with components $f_j - f_i$ $(i, j \in \mathcal{O}_k,\ i \neq j)$, where i is the index of the maximal function at x_k. In Table 2.3 the last column lists the maximizers of $y \mapsto f(x_k, y)$ found on the cell Y.

Ex.	k	$\lvert F(x_k) - F_* \rvert$	$\mathcal{O}_k$	$\lVert \nabla L_k(x_k) \rVert_2$
9	9	1.32601×10^{-10}	$1, 2, 4$	8.14538×10^{-6}
10	276	7.26219×10^{-6}	$1, 4$	9.69559×10^{-5}
11	289	1.42476×10^{-7}	$1, 3, 4, 5$	0.000955112
12	1402	0.00020687	$1, 2, 3$	0.00536328
13	652	2.41032×10^{-9}	$1, 2, \ldots, 100$	9.81901×10^{-6}

Table 2.2: Results for Algorithm 2.5.1.

Ex.	k	$\lvert F(x_k) - F_* \rvert$	y_k
14	46	3.26652×10^{-6}	$(2, -2, -2)$
15	12	5.81677×10^{-6}	$(2, -2, -2, 2); (-2, -2, -2, 2)$

Table 2.3: Results for Algorithm 2.5.3.

In Examples 14 and 15 the results are in accordance with the ones produced by the quasi-Newton algorithms tested in [38, Sec. 5.4], though in Example 15 the latter finish with two more maximizers: $(2, -2, -2, -2)$ and $(-2, -2, -2, -2)$.

2.6.2 Note. Some examples exhibit a somewhat slow convergence speed. It is expected, like in Examples 1–8, that the number of iterations will be greatly cut once the algorithm with spectral choice of step length is employed. ■

2.7 Final remarks

Encouraging results have been obtained in some preliminary tests for equality and/or inequality constrained problems. A theoretical study of the algorithms' performance (with special emphasis on deactivation criteria) is object of future work.

"Smooth algorithms" for (discrete and continuous) minimax problems were obtained, in a simple way, by facing the problem directly without resorting to artifices for circumventing the nonsmoothness of the objective function. The fact that the methods are based on an active-set strategy also plays in their favour: only the gradients of active functions are evaluated at each iteration, which is not a negligeable aspect in large scale applications.

Though several refinements are still required, particularly in the algorithm for continuous minimax problems, it is felt that the presented methodologies can produce some competitive methods. These methodologies are also very flexibile, in the sense that they can clearly integrate diverse courses of action at its different stages. The ones proposed reflect only a personal preference.

Chapter 3

Properties of cost functionals in free material optimization

This chapter focuses on the study of a certain type of integral functional arising in some fields of structural optimization, particularly the so-called *free material design*.

Section 3.1 introduces the fundamental notions and results to develop the analysis, and in Section 3.2 the mathematical framework is set. Sections 3.3 and 3.4 are the chapter's core: they deal with lower semicontinuity and its relation to convexity. The last section addresses two additional concepts, subadditivity and positive homogeneity, and their significance in practical terms is investigated, as well as their mathematical implications.

On notations

Throughout the chapter, $\mathbb{S} \subset \mathbb{R}^{n,n}$ $(n \geqslant 2)$ denotes the set of symmetric matrices and $\mathbb{S}^+ \subset \mathbb{S}$ the cone of (symmetric) positive semidefinite matrices. If $A \in \mathbb{S}$, $\lambda(A)$ denotes the n-tuple of eigenvalues of A sorted nonincreasingly: $\lambda_1(A) \geqslant \lambda_2(A) \geqslant \cdots \geqslant \lambda_n(A)$ – this is convenient because $A \mapsto \lambda(A)$ is then a well defined map from $\mathbb{S}$ into $\mathbb{R}^n$.

Given $A, B \in \mathbb{R}^{n,n}$ one writes $A \preccurlyeq B$, or $B \succcurlyeq A$, to mean that $B - A$ is positive semidefinite. For positive real numbers $\alpha < \beta$, one puts

$$S_{\alpha,\beta} = \{A \in \mathbb{S} : \ \alpha I \preccurlyeq A \preccurlyeq \beta I\},$$

and being $\Omega \subset \mathbb{R}^n$ a bounded domain, one further defines

$$\mathcal{S}_{\alpha,\beta}(\Omega) = \{A \in L^\infty(\Omega;\mathbb{S}) : \ A(x) \in S_{\alpha,\beta} \text{ for a.e. } x \in \Omega\},$$

i.e. the set of bounded measurable matrix functions $A : \Omega \to \mathbb{S}$ which, *almost everywhere*, take values in $S_{\alpha,\beta}$; the symbol "$\overset{\star}{\rightharpoonup}$" is used to denote weak $*$ convergence.

Finally, $\varepsilon = (\varepsilon_k)$ will denote a strictly positive sequence converging to zero.

3.1 Preliminaries

Composite materials (materials made up of more than one substance) play a significant role in many applied fields of science (engineering, mechanics, physics, *etc.*). The physical parameters in such materials are discontinuous, as they oscillate between the different

values characterizing each of its components. When these components are mixed at a very small "length" scale ε, the parameters oscillate rapidly and the *microscopic* structure becomes very complicated. Such an intricate mixture will surely possess very different properties from the ones of each constitutive substance, and it is not a far stretch of the imagination to think that it behaves in fact as a new material at the *macroscopic* level.[1] The way to get a good approximation of this macroscopic behaviour, is by letting the parameter ε tend to zero in the equations describing the physical phenomena in question. This limit process is described by the *theory of homogenization*.

A good model for the intended asymptotic analysis is provided by the equation

$$\begin{cases} -\operatorname{div}(A\nabla u) = f & \text{in } \Omega, \\ u = \bar{u} & \text{on } \partial\Omega, \end{cases} \tag{3.1}$$

where the matrix (or second order tensor) A can be interpreted as a conductivity tensor, meaning: it models properties like the ability to carry heat or electricity of a given physical material – reason why the terminology *material tensors* is common in such frameworks. The function u is the temperature or the electric potential, which is prescribed to be a given $\bar{u}$ over the boundary of the (bounded) domain $\Omega \subset \mathbb{R}^n$, and f is a given source term.

Since the intent is to model oscillatory behaviour, the material A is supposed to be *heterogeneous*, *i.e.* it varies pointwise (as opposed to a *homogeneous* one, which is constant); hence, the natural assumption on the coefficients of A is that they belong to $L^\infty(\Omega)$. Therefore, the functions u, $\bar{u}$ and f are to be taken in appropriate Sobolev spaces and (3.1) is to be understood in the *weak* sense. Existence and uniqueness results are standard in the literature [65, Secs. 6.4–6.5]. For now, suffice it to say that if $A \in \mathcal{S}_{\alpha,\beta}(\Omega)$, then the above problem is well-posed.

In the early 1970s, Murat and Tartar identified the appropriate type of convergence, *H-convergence* (or *convergence in the sense of homogenization*), that formalizes the limit procedure mentioned in the introductory discussion. The content for the rest of the section (and much more) can be found with more detail in their work [66–68].

3.1.1 Definition. *Let (A_ε) be a sequence of matrix functions in $\mathcal{S}_{\alpha,\beta}(\Omega)$. One says that (A_ε) H-converges to some $A_0 \in \mathcal{S}_{\alpha,\beta}(\Omega)$, denoted $A_\varepsilon \overset{H}{\rightharpoonup} A_0$, when for all $f \in H^{-1}(\Omega)$ and $\bar{u} \in H^{1/2}(\partial\Omega)$, the sequence of functions $u_\varepsilon \in H^1(\Omega)$ satisfying*

$$\begin{cases} -\operatorname{div}(A_\varepsilon \nabla u_\varepsilon) = f & \text{in } \Omega, \\ u_\varepsilon = \bar{u} & \text{on } \partial\Omega, \end{cases} \tag{3.2}$$

converges, weakly in $H^1(\Omega)$, to the function $u_0 \in H^1(\Omega)$ satisfying

$$\begin{cases} -\operatorname{div}(A_0 \nabla u_0) = f & \text{in } \Omega, \\ u_0 = \bar{u} & \text{on } \partial\Omega, \end{cases} \tag{3.3}$$

(Dirichlet conditions are considered solely for simplicity, as H-convergence is independent of the boundary conditions considered on $\partial\Omega$.)

[1] If one looks at a chessboard from sufficiently far a distance, one ceases to discern the black and white squares only to see a grey blob instead.

The matrix A_0 appearing in the "limit problem" (3.3) is a *homogenized* (or *effective*) tensor, a material whose behaviour is equivalent, from a macroscopic perspective, to the highly heterogeneous one of A_ε when $\varepsilon \approx 0$. This makes all the difference regarding a numerical solution of (3.2); a finite element approximation of u_ε has unaffordable computational costs for small ε, while (3.3) is often easy to solve numerically.

3.1.2 Remark. The set $\mathcal{S}_{\alpha,\beta}(\Omega)$ is a *compact metrizable space* when endowed with the H-topology, that is, the topology associated with the previous notion of convergence. ■

3.1.3 Remark. $\mathcal{S}_{\alpha,\beta}(\Omega)$ is also compact and metrizable with respect to the weak $*$ topology of $L^\infty(\Omega;\mathbb{S})$. However, if $A_\varepsilon \overset{\star}{\rightharpoonup} A_+$, the sequence (u_ε) of solutions verifying (3.2) converges, weakly in $H^1(\Omega)$, to some u_0 which *is not* necessarily the solution of the problem corresponding to A_+. ■

In general there are no explicit formulae for the H-limit of a given sequence of tensors. Two exceptions to this rule are: the case of materials with a periodic structure, and the case of *layered* (or *laminated*) materials which is presented next (it will play a crucial role in the results of Section 3.3).

3.1.4 Theorem. *If the coefficients A_ε^{ij} of a sequence (A_ε) in $\mathcal{S}_{\alpha,\beta}(\Omega)$ depend only on one coordinate, say x_1, then $A_\varepsilon \overset{H}{\rightharpoonup} A_0$ is equivalent to:*

(a)
$$\frac{1}{A_\varepsilon^{11}} \overset{\star}{\rightharpoonup} \frac{1}{A_0^{11}};$$

(b) *for $i \neq 1$,*
$$\frac{A_\varepsilon^{i1}}{A_\varepsilon^{11}} \overset{\star}{\rightharpoonup} \frac{A_0^{i1}}{A_0^{11}};$$

(c) *for $i \neq 1$ and $j \neq 1$,*
$$A_\varepsilon^{ij} - \frac{A_\varepsilon^{i1} A_\varepsilon^{1j}}{A_\varepsilon^{11}} \overset{\star}{\rightharpoonup} A_0^{ij} - \frac{A_0^{i1} A_0^{1j}}{A_0^{11}}.$$

The following result provides a more accurate "bound" on the H-limit of sequences belonging to the set $\mathcal{S}_{\alpha,\beta}(\Omega)$.

3.1.5 Theorem. *Consider a sequence (A_ε) in $\mathcal{S}_{\alpha,\beta}(\Omega)$ such that $A_\varepsilon \overset{H}{\rightharpoonup} A_0$, $A_\varepsilon \overset{\star}{\rightharpoonup} A_+$ and $A_\varepsilon^{-1} \overset{\star}{\rightharpoonup} A_-^{-1}$. Then $A_-(x) \preccurlyeq A_0(x) \preccurlyeq A_+(x)$ for a.e. $x \in \Omega$.*

Actually, the issue of "bounds" is one of the most important (and toughest) in homogenization theory. The goal is to characterize the matrices A_0 that can be obtained as H-limits of sequences (A_ε) belonging to a proper subset of $\mathcal{S}_{\alpha,\beta}(\Omega)$. One of the few cases where an optimal result can be established is now introduced.

For each $\theta \in [0,1]$ define the numbers

$$\begin{cases} \mu_+(\theta) = (1-\theta)\alpha + \theta\beta, \\ \mu_-(\theta) = \left(\dfrac{1-\theta}{\alpha} + \dfrac{\theta}{\beta}\right)^{-1}, \end{cases}$$

and the set $K_\theta \subset \mathbb{R}^n$ by

$$\mu \in K_\theta \iff \begin{cases} \mu_-(\theta) \leqslant \mu_i \leqslant \mu_+(\theta), \quad i = 1, \dots, n, \\ \displaystyle\sum_{i=1}^{n} \frac{1}{\mu_i - \alpha} \leqslant \frac{1}{\mu_-(\theta) - \alpha} + \frac{n-1}{\mu_+(\theta) - \alpha}, \\ \displaystyle\sum_{i=1}^{n} \frac{1}{\beta - \mu_i} \leqslant \frac{1}{\beta - \mu_-(\theta)} + \frac{n-1}{\beta - \mu_+(\theta)}; \end{cases} \tag{3.4}$$

define also the set $G_\theta \subset S_{\alpha,\beta}$ of matrices A whose n-tuple of eigenvalues belongs to K_θ:

$$G_\theta = \{A \in \mathbb{S} : \ \lambda(A) \in K_\theta\}. \tag{3.5}$$

The next theorem identifies all the attainable materials, through homogenization, by mixtures of two materials (αI and βI).

3.1.6 Theorem. *Assume that $A_\varepsilon \overset{H}{\rightharpoonup} A_0$ and that $A_\varepsilon(x) = a_\varepsilon(x) I$, where a_ε takes only values α and β. Assume also that, for some $\theta \in L^\infty(\Omega; [0,1])$,*

$$a_\varepsilon \overset{\star}{\rightharpoonup} (1-\theta)\alpha + \theta\beta. \tag{3.6}$$

Then holds:

$$A_0(x) \in G_{\theta(x)} \quad \text{for a.e. } x \in \Omega. \tag{3.7}$$

Conversely, if $A_0 \in \mathcal{S}_{\alpha,\beta}(\Omega)$ and $\theta \in L^\infty(\Omega; [0,1])$ satisfy (3.7), a sequence (a_ε) of measurable functions exists, evaluating only to α or β, such that (3.6) holds and $a_\varepsilon I \overset{H}{\rightharpoonup} A_0$.

3.1.7 Note. The set of homogenized tensors G_θ is also commonly known as G-closure; one usually writes $a_\varepsilon(x) = (1 - \chi_\varepsilon)(\alpha I) + \chi_\varepsilon(\beta I)$, where χ_ε designates the characteristic function of the subset $\omega_\varepsilon \subset \Omega$ filled with material βI, whose pointwise proportion in the effective material A_0 is given by the function θ (of course, αI then fills $\Omega \backslash \omega_\varepsilon$ and is present in A_0 with pointwise proportion $1 - \theta$). ■

3.2 Setting of the problem

A basic problem in structural engineering is to design the strongest (or more accurately, the stiffest) structure capable of withstanding a given set of external loads. By structure is here meant an *elastic body*[2]. The framework is that of *linear elasticity*, different from the one of the previous section: the algebraic entities that model material properties are no longer matrices (second order tensors), but *fourth order tensors* instead (see Section 4.1).

For symplicity's sake, the analysis is restricted to the conductivity setting (a fourth order tensor is a much more complicated object, as are several underlying concepts). Yet, one may sometimes find it easier to employ terminology from the field of elasticity, not only because that is where the meaningful and interesting case for applications lie, but also because it conveys a clearer physical meaning.

[2]Roughly, a deformable body undergoing *reversible* changes of shape in response to applied forces. For instance, a metallic spring stretches when pulled, but reverts back to its original state when let loose.

Optimization of structures is usually performed by varying size and shape parameters (*e.g.* the thickness of a plate, the boundary of a solid body, *etc.*). With the advent of composites and other advanced manufactured materials, it was a natural step to consider the material tensor itself as the main optimization parameter. This approach, which ultimately searches for the best structure among all possible ones, became commonly known as *free material design*. The concept was introduced by Ringertz [69] and subsequently developed with more detail in the paper by Bendsøe *et al.* [70], where it was suggested to represent materials as positive semidefinite tensors.

Let M be a subset of $\mathbb{S}^+$. An example is the set $S_{\alpha,\beta}$; another example is the set G, defined in Section 3.4, consisting of all homogenized materials which can be obtained by mixing the materials αI and βI. Now let Ω be a bounded domain in $\mathbb{R}^n$. Since the material tensor is supposed to vary pointwise over Ω, the functions involved in a free material optimization problem are usually defined on the set $\mathcal{M}(\Omega)$ of measurable functions $A : \Omega \to M$. Available resources in real world applications are not unlimited; to reflect such a modelling aspect, a constraint is considered on the material tensors in terms of a *cost function* $\Phi : \mathcal{M}(\Omega) \to \mathbb{R}$, defined by

$$\Phi(A) = \int_\Omega \phi(A(x))\, dx, \tag{3.8}$$

where $\phi : M \to \mathbb{R}$ is a *spectral (or isotropic) function*, that is:

$$\phi(Q^T A Q) = \phi(A) \quad \textit{for all } Q \in \mathbb{O} \textit{ and } A \in M, \tag{3.9}$$

$\mathbb{O} \subset \mathbb{R}^{n,n}$ being the set of all orthogonal matrices. Equivalently, ϕ should depend only on the eigenvalues of the matrix A; this is mandatory in free material design, as it ensures that a structure's "price" (which is what, in a broad sense, a cost function stands for) is independent of reference frame. Note that M must be a *spectral set*: given $A \in M$, then $\{Q^T A Q :\ Q \in \mathbb{O}\} \subset M$.

The properties of the functional Φ are thus of great importance in the free material approach and they depend, of course, on properties of the integrand ϕ. Besides (3.9), natural requirements include a certain degree of smoothness of ϕ (continuity at least), as well as the following monotonicity property:

$$\textit{for all } A, B \in M, \qquad B \succcurlyeq A \implies \phi(B) \geqslant \phi(A); \tag{3.10}$$

this requirement translates the practical fact that a stronger material must not be "cheaper" than a weaker one.

A free material optimization problem, in its simplest form, deals with the minimization of some objective functional $\Psi : \mathcal{M}(\Omega) \to \mathbb{R}$, which measures the performance of the structure according to some criteria, within a range of materials not exceeding some prescribed "price" (*i.e.* on a sublevel set of Φ). The lower semicontinuity of Φ is thus crucial, and it is related to the convexity of its integrand ϕ.

3.2.1 Note. Ψ depends on A through the solution $u = u(A)$ of some elliptic problem in Ω, like (3.1) for instance, with A representing the material coefficients. No particular form of that elliptic problem is chosen because, although the objective functional itself should be lower semicontinuous in order for the optimization problem to be well-posed, the study is focused on the properties of the cost functional Φ. ■

3.3 Lower semicontinuity

Some literature [71, Subsec. 5.2.1] address the issue of lower semicontinuity with respect to the weak $*$ topology. This is usually sufficient to ensure well-posedness of the underlying optimization problem, and also convenient because it is easy to characterize integrands ϕ which turn the integral functional Φ lower semicontinuous [72, Th. 4]:

3.3.1 Theorem. *The functional Φ defined by (3.8) is lower weak$*$ semicontinuous on $\mathcal{S}_{\alpha,\beta}(\Omega)$ if and only if the integrand ϕ is a convex function on $S_{\alpha,\beta}$.*

But as stated in Section 3.1, the notion that best describes the limit behaviour of oscillating material coefficients is H-convergence (see also Remark 3.1.3). Unfortunately, it is not easy to prove lower semicontinuity with respect to the H-topology. Under the monotonicity assumption, one can prove easily one implication:

3.3.2 Theorem. *Suppose that ϕ is a continuous nondecreasing function on $S_{\alpha,\beta}$, in the sense of (3.10). If the functional Φ defined on $\mathcal{S}_{\alpha,\beta}(\Omega)$ by (3.8) is lower weak$*$ semicontinuous, then Φ is also lower H-semicontinuous.*

Proof. Consider a sequence (A_ε) in $\mathcal{S}_{\alpha,\beta}(\Omega)$, H-converging to $A \in \mathcal{S}_{\alpha,\beta}(\Omega)$. Taking into account that $\mathcal{S}_{\alpha,\beta}(\Omega)$ is compact with respect to the weak $*$ topology, consider a subsequence ε' of ε such that $(A_{\varepsilon'})$ converges weakly $*$ to some $A_+ \in \mathcal{S}_{\alpha,\beta}(\Omega)$, fulfilling also the condition $\liminf \phi(A_\varepsilon) = \lim \phi(A_{\varepsilon'})$. By Theorem 3.1.5 follows that $A \leqslant A_+$, so $\phi(A) \leqslant \phi(A_+) \leqslant \liminf \phi(A_{\varepsilon'})$ and the assertion follows. □

Theorems 3.3.1 and 3.3.2 imply the following:

3.3.3 Corollary. *If ϕ is a continuous convex function on $S_{\alpha,\beta}$, nondecreasing in the sense of (3.10), then the functional Φ defined on $\mathcal{S}_{\alpha,\beta}(\Omega)$ by (3.8) is lower H-semicontinuous.*

In general, there is no simple characterization of the lower semicontinuity of Φ with respect to the H-topology of $\mathcal{S}_{\alpha,\beta}(\Omega)$. However, in the particular case when ϕ depends only on the trace of the coefficient matrix,

$$\phi(A) = \varphi(\operatorname{tr}(A)) \quad \textit{for all } A \in S_{\alpha,\beta}, \tag{3.11}$$

and assuming also the monotonicity property, one can prove that lower H-semicontinuity of Φ on $\mathcal{S}_{\alpha,\beta}(\Omega)$ is equivalent to the convexity of φ.

3.3.4 Theorem. *Let $\varphi : [n\alpha, n\beta] \to \mathbb{R}$ be a continuous nondecreasing function. Define*

$$\phi : S_{\alpha,\beta} \to \mathbb{R}, \quad \phi(A) = \varphi(\operatorname{tr}(A))$$

and

$$\Phi : \mathcal{S}_{\alpha,\beta}(\Omega) \to \mathbb{R}, \quad \Phi(A) = \int_\Omega \phi(A(x))\,dx = \int_\Omega \varphi(\operatorname{tr}(A(x)))\,dx.$$

Then, Φ is lower H-semicontinuous on $\mathcal{S}_{\alpha,\beta}(\Omega)$ if and only if φ is convex.

Proof. for $n = 2$ ($n > 2$ is similar):

The sufficiency follows immediately from Corollary 3.3.3, since $A \mapsto \mathrm{tr}(A)$ is linear and nondecreasing in the sense of (3.10).

The necessity is proven by building a specific sequence of laminates. Consider γ, x and y arbitrary numbers in $[\alpha, \beta]$. Define

$$A_\varepsilon = \chi_\varepsilon \begin{bmatrix} \gamma & 0 \\ 0 & x \end{bmatrix} + (1 - \chi_\varepsilon) \begin{bmatrix} \gamma & 0 \\ 0 & y \end{bmatrix}, \tag{3.12}$$

where $\chi_\varepsilon \overset{\star}{\rightharpoonup} \theta$ in $L^\infty(\Omega)$, θ being a constant value in $[0, 1]$. According to Theorem 3.1.4, this sequence of laminates H-converges to

$$\begin{bmatrix} \gamma & 0 \\ 0 & \theta x + (1-\theta) y \end{bmatrix},$$

and the lower semicontinuity condition reads

$$\varphi(\gamma + \theta x + (1-\theta) y) \leqslant \theta \varphi(\gamma + x) + (1-\theta)\varphi(\gamma + y).$$

Thus, φ is convex in the interval $[\gamma + \alpha, \gamma + \beta]$. Since γ is arbitrary in $[\alpha, \beta]$, we obtain the desired convexity of φ in $[2\alpha, 2\beta]$. □

It is only under the hypothesis (3.11) that one has an explicit characterization of the lower semicontinuity of Φ with respect to the H-topology of $\mathcal{S}_{\alpha,\beta}(\Omega)$.

Convexity of φ is also necessary when ϕ depends solely on the determinant of the coefficient matrix: $\phi(A) = \varphi(\det(A))$. One takes precisely the laminates (3.12) and a similar reasoning leads now to the convexity of φ on the interval $[\alpha^2, \beta^2]$. However, it is easy to prove that functions depending only on the determinant of A are not admissible costs:

3.3.5 Theorem. *Let $\varphi : [\alpha^n, \beta^n] \to \mathbb{R}$ be a differentiable convex function. Define*

$$\phi : S_{\alpha,\beta} \to \mathbb{R}, \qquad \phi(A) = \varphi(\det(A))$$

and

$$\Phi : \mathcal{S}_{\alpha,\beta}(\Omega) \to \mathbb{R}, \qquad \Phi(A) = \int_\Omega \phi(A(x))\, dx = \int_\Omega \varphi(\det(A(x)))\, dx.$$

If Φ is lower H-semicontinuous on $\mathcal{S}_{\alpha,\beta}(\Omega)$, then φ is constant.

Proof. for $n = 2$ ($n > 2$ is similar):

Consider four arbitrary points $a, b, c, d \in [\alpha, \beta]$. Building the laminates

$$A_\varepsilon = \chi_\varepsilon \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} + (1 - \chi_\varepsilon) \begin{bmatrix} c & 0 \\ 0 & d \end{bmatrix}$$

and applying the lower semicontinuity hypothesis, one obtains the inequality

$$\varphi\left(\frac{\theta b + (1-\theta) d}{\frac{\theta}{a} + \frac{1-\theta}{c}} \right) \leqslant \theta\, \varphi(ab) + (1-\theta)\, \varphi(cd)$$

with equality for $\theta = 0$.

Now, differentiating the above inequality in $\theta = 0$:

$$c^2\varphi'(cd)\Big[(b-d)\,\frac{1}{c} - d\left(\frac{1}{a}-\frac{1}{c}\right)\Big] \leqslant \varphi(ab) - \varphi(cd),$$

that is,

$$\frac{c}{a}\,\varphi'(cd)\,(ab-cd) \leq \varphi(ab) - \varphi(cd).$$

Take $\lambda \in]\alpha,\beta[$ and choose $c = d = \lambda$, $a \in [\frac{\lambda^2}{\beta}, \lambda[\cap[\alpha,\beta]$; define $b_\delta = \frac{\lambda^2+\delta}{a}$, where $\delta > 0$ is a small parameter. One checks easily that, for sufficiently small δ, $b_\delta \in [\alpha,\beta]$. Taking into account that $ab_\delta = \lambda^2 + \delta$ and $cd = \lambda^2$, the last inequality is rewriten as

$$\frac{\lambda}{a}\,\varphi'(\lambda^2) \leqslant \frac{\varphi(\lambda^2+\delta) - \varphi(\lambda^2)}{\delta}$$

to let $\delta \searrow 0$, recalling that $a < \lambda$, in order to conclude $\varphi'(\lambda^2) \leqslant 0$.

Choosing now $a \in]\lambda, \frac{\lambda^2}{\alpha}]$, the same $c = d = \lambda$ and $b_\delta = \frac{\lambda^2+\delta}{a}$, the inequality above implies $\varphi'(\lambda^2) \geqslant 0$.

It was proved that $\varphi'(\lambda^2) = 0$ for all $\lambda \in [\alpha,\beta]$; hence, φ is constant in $[\alpha^2,\beta^2]$. □

3.3.6 Remark. Theorem 3.3.5 also shows something else: reminding that $\phi(A) = \det(A)$ is a polyconvex function, and thus quasiconvex and rank-one convex as well, all these weaker types of convexity [73, Chap. 5] are excluded as sufficient conditions. In order for Φ to be lower semicontinuous on $\mathcal{S}_{\alpha,\beta}(\Omega)$ with respect to H-convergence, the integrand ϕ has to be either convex or something strictly between convex and polyconvex. In the following section it will be shown that ϕ may indeed be not convex. ■

3.3.7 Remark. Note that the proofs of Theorems 3.3.4 and 3.3.5 use sequences of laminates. This means that the conditions therein described are necessary for lower semicontinuity in a weaker convergence than the one of the H-topology: they are necessary for the lower semicontinuity of Φ with respect to the convergence of laminates [74]. ■

To close the section, some simple examples of lower semicontinuous cost functionals (with respect to the H-topology) are presented. The monotonicity property (3.10) follows easily from a well known result in matrix analysis [75, Cor. III.2.3]:

3.3.8 Theorem (Weyl's monotonicity theorem)**.** *Let A, H be $n \times n$ matrices, with A hermitian and H positive semidefinite. Then*

$$\lambda_i(A+H) \geqslant \lambda_i(A) \quad \textit{for all } i \in \{1,2,\ldots,n\}.$$

The first example is one of the most common choices in free material optimization: $\phi(A) = \mathrm{tr}(A)$. Lower semicontinuity of the corresponding cost functional Φ is a direct consequence of Theorem 3.3.4.

The other popular example in free material settings, together with the previous one, is the Frobenius norm: $\phi(A) = \|A\|_F = \sqrt{\mathrm{tr}(A^TA)}$, which equals $\sqrt{\mathrm{tr}(A^2)}$ since A is symmetric. It is spectral because the trace itself is a spectral function. Lower semicontinuity of Φ is immediate from Corollary 3.3.3 since ϕ (being a norm) is a convex function.

The last example, $\phi(A) = \max_{\|\xi\|=1}\langle A\xi,\xi\rangle$, is the spectral radius (isotropy is thus guaranteed), since only positive definite matrices are dealt with. Being the pointwise supremum of the family $\{\phi_\xi\}_{\|\xi\|=1}$, with $\phi_\xi(A) = \langle A\xi,\xi\rangle$, it is surely a convex function. Lower semicontinuity of the resulting cost functional Φ is again due to Corollary 3.3.3.

3.4 An example from homogenization theory

In the previous section, convexity of ϕ was shown to be a sufficient condition for the lower H-semicontinuity of Φ, while polyconvexity was ruled out (and therefore quasiconvexity and rank-one convexity as well); in fact, when ϕ depends solely on the trace of its argument, convexity is also seen to be a necessary condition for lower semicontinuity with respect to H-convergence (Theorem 3.3.4). In the present section, it will be shown that such an assertion cannot be expected to hold for the case of a general integrand ϕ.

Consider the sets K_θ and G_θ introduced in (3.4) and (3.5), respectively. K_θ is a convex and *symmetric set*: if $\mu \in K_\theta$, then $\mu_\sigma = (\mu_{\sigma(1)}, \dots, \mu_{\sigma(n)}) \in K_\theta$ for each permutation σ of $\{1, \dots, n\}$. Hence, the inverse image $G_\theta = \lambda^{-1}(K_\theta)$ is a spectral set. The same observation holds true for the sets

$$K = \bigcup_{0 \leqslant \theta \leqslant 1} K_\theta, \qquad G = \bigcup_{0 \leqslant \theta \leqslant 1} G_\theta = \lambda^{-1}(K); \tag{3.13}$$

furthermore, it is not difficult to prove that K is convex.

3.4.1 Note. As mentioned in Section 3.1, for a given $\theta \in [0,1]$, G_θ is the set of material tensors attainable through homogenization by mixtures of αI and βI, in proportions $1-\theta$ and θ, respectively; thus, G is the set of all such homogenized materials. ■

One now defines $\phi : G \to \mathbb{R}$ by

$$\phi(A) = \min\{\theta \in [0,1] : \ A \in G_\theta\}, \tag{3.14}$$

which gives rise to the "cheapest" mixture between αI and βI producing the material A [76]. One obtains a cost functional Φ defined not on the entire $\mathcal{S}_{\alpha,\beta}(\Omega)$, but only on the subset of those matrix functions taking values in G:

$$\mathcal{G}(\Omega) = \{A \in L^\infty(\Omega; \mathbb{S}) : \ A(x) \in G \text{ for a.e. } x \in \Omega\}.$$

It is a simple exercise to prove that such a set is compact for the H-topology.

3.4.2 Remark. Of course, $\phi(A) = \min\{\theta \in [0,1] : \ \lambda(A) \in K_\theta\}$; an explicit expression, in terms of eigenvalues, is found by taking the equality in the third relation of (3.4) and solving for θ:

$$\phi(A) = \varphi(\lambda(A)), \qquad \varphi(\mu) = 1 - \frac{n\beta(\beta-\alpha)^{-1} - 1}{\beta \sum_{i=1}^n (\beta - \mu_i)^{-1} - 1} \tag{3.15}$$

with $\varphi(\beta, \beta, \dots, \beta) = 1$, where $\varphi : K \to \mathbb{R}$ is obviously a *symmetric function*, that is, $\varphi(\mu) = \varphi(\mu_\sigma)$ for every $\mu \in K$ and each permutation σ of $\{1, \dots, n\}$. ■

In view of (3.15), the function ϕ is isotropic and continuous on G. Monotonicity is once again easy to establish thanks to Theorem 3.3.8. Furthermore:

3.4.3 Theorem. *Take $\Phi : \mathcal{G}(\Omega) \to \mathbb{R}$ defined by $\Phi(A) = \int_\Omega \phi(A(x))\, dx$, with $\phi : G \to \mathbb{R}$ given in (3.14). Then Φ is lower H-semicontinuous.*

Proof. Let $A_\varepsilon \overset{H}{\rightharpoonup} A$ in $\mathcal{G}(\Omega)$; the "local costs" $\theta_\varepsilon(x) = \phi(A_\varepsilon(x))$ can be supposed to converge weakly $*$, say $\theta_\varepsilon \overset{\star}{\rightharpoonup} \theta_0$, because $L^\infty(\Omega;[0,1])$ is weak $*$ compact. From Theorem 3.1.6 it is known that each A_ε is a H-limit of a sequence $\chi_\eta^\varepsilon(\beta I) + (1-\chi_\eta^\varepsilon)(\alpha I)$, with $\chi_\eta^\varepsilon \overset{\star}{\rightharpoonup} \theta_\varepsilon$. Since $L^\infty(\Omega;[0,1])\times\mathcal{G}(\Omega)$ is a metrizable space, a diagonal sequence $(\chi_{\eta_\varepsilon}^\varepsilon)$ can be extracted such that

$$\chi_{\eta_\varepsilon}^\varepsilon \overset{\star}{\rightharpoonup} \theta_0 \quad \text{and} \quad \chi_{\eta_\varepsilon}^\varepsilon(\beta I) + (1-\chi_{\eta_\varepsilon}^\varepsilon)(\alpha I) \overset{H}{\rightharpoonup} A;$$

again from Theorem 3.1.6, this means precisely that $A(x) \in G_{\theta_0(x)}$ *for a.e.* $x \in \Omega$. By definition of $\theta(x) = \phi(A(x))$, it is then obvious that

$$\int_\Omega \theta(x)\,dx \leqslant \int_\Omega \theta_0(x)\,dx = \liminf \int_\Omega \theta_\varepsilon(x)\,dx,$$

thus proving the assertion. □

However, ϕ is not a convex function! To show this it suffices to use the following result (see the paper of Daniilidis *et al.* [77, Sec. 1] and the references therein):

3.4.4 Theorem. *Let $K \subset \mathbb{R}^n$ be convex and symmetric and suppose $\varphi : K \to \mathbb{R}$ is a symmetric function. Then the set $\lambda^{-1}(K)$ is convex and the spectral function $\phi = \varphi \circ \lambda$ is convex if and only if φ is convex.*

To prove that ϕ defined by (3.14) is not convex, one just has to show that φ given in (3.15) is itself a nonconvex function. To see this, observe that the set K defined in (3.13) contains the line segment with endpoints $(\alpha,\alpha,\dots,\alpha)$ and $(\beta,\beta,\dots,\beta)$; therefore, if φ were convex on K, the function defined by

$$\psi(t) = \varphi(t,t,\dots,t) = \begin{cases} 1 - \dfrac{n\beta(\beta-\alpha)^{-1}-1}{n\beta(\beta-t)^{-1}-1}, & t \neq \beta, \\ 1, & t = \beta, \end{cases}$$

would be convex on $[\alpha,\beta]$; but for every $t \in \,]\alpha,\beta[$

$$\psi''(t) = -\frac{2n\beta[n\beta(\beta-\alpha)^{-1}-1]}{[(n-1)\beta+t]^3} < 0,$$

being ψ in fact concave.

3.4.5 Remark. The actual property that completely characterizes the lower semicontinuous functionals – with respect to the H-topology of $\mathcal{S}_{\alpha,\beta}(\Omega)$ – defined by (3.8), with integrand satisfying (3.9) and (3.10), still remains an open question, but it is now cornered strictly between convexity and polyconvexity. ■

3.4.6 Remark. The functional Φ defined by (3.8) with integrand ϕ given by (3.14), or equivalently by (3.15), is an interesting example of an integral functional that is lower H-semicontinuous, but not lower weak $*$ semicontinuous (the integrand is not convex). ■

3.5 Subbaditivity and positive homogeneity

The most frequent cost functionals, used by engineers in free material optimization, feature an integrand ϕ that is either the trace or the Frobenius norm. On top of verifying properties (3.9) and (3.10), they also satisfy two additional ones: *subadditivity*,

$$\text{for all } A, B \in M, \qquad A + B \in M \implies \phi(A + B) \leqslant \phi(A) + \phi(B), \tag{3.16}$$

and *positive homogeneity* (of degree one),

$$\text{for all } t > 0 \text{ and } A \in M, \qquad tA \in M \implies \phi(tA) = t\,\phi(A), \tag{3.17}$$

where M is a subset of $\mathbb{S}^+$ (actually, the trace is a linear function). These properties can be given an intuitive mechanical motivation, showing them to be acceptable requisites for a cost functional. The first one basically states that the price of a material built by superimposing two base materials should not be higher than their combined prices. The second one implies, for instance, that upon halving the strength of some material, its price is also halved.

We now retrieve the example of the previous section, but with $\alpha = 0$, that is, mixtures between material and void; hence, (3.14) is simply the smallest proportion of material which produces A and (3.15) reduces to

$$\phi(A) = \varphi(\lambda(A)), \qquad \varphi(\mu) = 1 - \frac{n-1}{\beta \sum_{i=1}^{n} (\beta - \mu_i)^{-1} - 1}. \tag{3.18}$$

Note that, in this case, the set K_θ (respectively, G_θ) increases with θ and ultimately $K_1 = [0, \beta]^n = K$ (respectively, $G_1 = \{A \in \mathbb{S}^+ : A \preccurlyeq \beta I\} = G$).

3.5.1 Remark. The passage from $\alpha > 0$ to $\alpha = 0$ is mathematically delicate [78,79]. The case $\alpha = 0$ is nevertheless highly relevant, since in most industrial applications it is much easier to build a perforated material than a mixture of different materials. Also, this case (involving only one material and void) is conceptually simpler. ■

It is expected of (3.18) to define a subadditive function; otherwise, it would imply that for the manufacture of a given material $A \in G$, a proportion of material smaller than $\phi(A)$ could be spent by producing first some $A_1, A_2 \in G$ and then superimposing them!

3.5.2 Remark. This operation of superimposing two (previously manufactured) materials is quite natural in two dimensions ($n = 2$): it suffices to glue two sheets of material one on top of the other. It is less intuitive in three dimensions. However, if the volume fractions are low, one can imagine two foam-like materials interpenetrating each other. ■

The subadditivity property of the function ϕ, defined in (3.18), has been investigated using both analytic and numerical tools. The conclusion was that it is fulfilled in two dimensions, but not in the three (or higher) dimensional case.

3.5.3 Theorem. *Let $n = 2$, $K = [0, \beta]^2$, $G = \lambda^{-1}(K) = \{A \in \mathbb{S}^+ : A \preccurlyeq \beta I\}$ and $\varphi : K \to \mathbb{R}$, $\phi : G \to \mathbb{R}$ be defined by (3.18), with $\varphi(\mu_1, \beta) = \varphi(\beta, \mu_2) = 1$ for every $\mu_1, \mu_2 \in [0, \beta]$. Then ϕ is subadditive.*

The proof makes use of the well known Weyl inequalities for eigenvalues [75, Th. III.2.1] (recall that $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n$):

3.5.4 Theorem. *Let A, B be $n \times n$ hermitian matrices. Then,*

$$\lambda_j(A+B) \leqslant \lambda_i(A) + \lambda_{j-i+1}(B) \quad \textit{for } i \leqslant j,$$
$$\lambda_j(A+B) \geqslant \lambda_i(A) + \lambda_{j-i+n}(B) \quad \textit{for } i \geqslant j.$$

Proof of Theorem 3.5.3. The subadditivity condition (3.16) reads: for $A, B \in G$ such that $A + B \in G$,

$$1 + \frac{n-1}{\sum_{i=1}^n \frac{\beta}{\beta-\lambda_i(A+B)} - 1} - \frac{n-1}{\sum_{i=1}^n \frac{\beta}{\beta-\lambda_i(A)} - 1} - \frac{n-1}{\sum_{i=1}^n \frac{\beta}{\beta-\lambda_i(B)} - 1} \geqslant 0; \tag{3.19}$$

we begin by rewriting this condition in a more convenient way by introducing the quantities $r = \sum_{i=1}^n r_i$, $s = \sum_{i=1}^n s_i$ and $t = \sum_{i=1}^n t_i$, where for every $i \in \{1, 2, \ldots, n\}$

$$r_i = \frac{\beta}{\beta - \lambda_i(A)} - 1 = \frac{\lambda_i(A)}{\beta - \lambda_i(A)}, \quad s_i = \frac{\lambda_i(B)}{\beta - \lambda_i(B)}, \quad \text{and} \quad t_i = \frac{\lambda_i(A+B)}{\beta - \lambda_i(A+B)}.$$

In terms of the (nonnegative) scalars r, s and t, (3.19) writes

$$\frac{rst + 2(n-1)rs + (n-1)^2(r+s-t)}{(r+n-1)(s+n-1)(t+n-1)} \geqslant 0; \tag{3.20}$$

since the denominator on the left is obviously nonnegative, one must show that the numerator is also nonnegative. That is the case in two dimensions.

Inserting $n = 2$ in (3.20), one has to prove that $rst + 2rs + r + s - t \geqslant 0$, for which purpose it suffices to see that

$$\sum_{i=1}^{2} [r_i s_i t_i + 2r_i s_i + r_i + s_i - t_i] \geqslant 0 \tag{3.21}$$

(several terms from rst and rs are missing, *but they are all nonnegative*). Simple calculations show that

$$r_i s_i t_i + 2r_i s_i + r_i + s_i - t_i = \frac{\lambda_i(A) + \lambda_i(B) - \lambda_i(A+B)}{[\beta - \lambda_i(A)][\beta - \lambda_i(B)][\beta - \lambda_i(A+B)]}$$

and therefore (3.21) is equivalent to

$$\frac{\lambda_1(A) + \lambda_1(B) - \lambda_1(A+B)}{[\beta - \lambda_1(A)][\beta - \lambda_1(B)][\beta - \lambda_1(A+B)]} \geqslant \frac{\lambda_2(A+B) - \lambda_2(A) - \lambda_2(B)}{[\beta - \lambda_2(A)][\beta - \lambda_2(B)][\beta - \lambda_2(A+B)]};$$

note that both numerators are positive in view of Theorem 3.5.4. The above relation holds, thus proving the subadditivity of ϕ, because both members have the same numerator, since

$$\sum_{i=1}^{2} [\lambda_i(A) + \lambda_i(B) - \lambda_i(A+B)] = \operatorname{tr}(A) + \operatorname{tr}(B) - \operatorname{tr}(A+B) = 0,$$

and because the denominator on the left hand side is obviously smaller than the one on the right hand side – recall that $\lambda_1(A) \geqslant \lambda_2(A)$ for any $A \in \mathbb{S}$. □

3.5.5 Remark. Quite unexpectedly, a similar statement to Theorem 3.5.3 fails to hold in general, including for $n = 3$ (the other physically relevant case)! For instance, if $A = \operatorname{diag}(0.4, 0.6, 0.3)$ and $B = \operatorname{diag}(0.1, 0.3, 0.1)$, one has $\phi(A + B) = \varphi(0.5, 0.9, 0.4) \approx 0.84$; but $\phi(A) + \phi(B) = \varphi(0.4, 0.6, 0.3) + \varphi(0.1, 0.3, 0.1) \approx 0.81$. This means that "cheaper" ways exist of producing a mixture than directly through homogenization!

Moreover, numerical experiments suggest that this violation of subadditivity happens even at low volume fractions (eigenvalues close to zero). As an example, take

$$\lambda(A) = (0.0002815, 0.00601019, 0.0305253), \quad \lambda(B) = (0.00544244, 0.00136686, 0.112491);$$

then $\phi(A) + \phi(B) \approx 0.081169239$, while $\phi(A + B) \approx 0.08259915$. See Remark 3.5.2 for the relevance of low volume fractions. ■

3.5.6 Remark. It is also puzzling that, regardless of dimension, ϕ is not positively homogeneous! This is obvious by looking at (3.18) – note that positive homogeneity in terms of a matrix variable is equivalent to that in terms of eigenvalues; for instance, one would expect $2A$ to consume twice the amount of material used for A, but it does not! ■

The function ϕ defined in (3.18), despite appearing naturally in homogenization theory, has some important drawbacks. One of them is the upper limit on the admissible materials, $A \preccurlyeq \beta I$, which prohibits concentrations of material. Remarks 3.5.5 and 3.5.6 show two additional limitations of this functional. These considerations imply that a seemingly natural cost functional may actually be not appropriate for practical applications.

There are some classic results relating subadditivity with concavity [80, Chap. 7], but these apply only to real variable functions. A more important result can be found in the book of Rockafellar [81, Th. 4.7] and promptly yields:

3.5.7 Theorem. *Let $\phi : \mathbb{S}^+ \to \mathbb{R}$ be positively homogeneous. Then ϕ is subadditive if and only if it is convex.*

To better underline the mathematical implications of the previous result, recall that lower H-semicontinuous cost functionals Φ whose integrand ϕ verifies only properties (3.9) and (3.10) are yet to be completely identified (see Remark 3.4.5). However, when both subadditivity and positive homogeneity are added to the picture, Theorem 3.5.7 basically determines all such cost functionals (since according to Corollary 3.3.3, convexity of ϕ is sufficient to ensure lower H-semicontinuity of Φ).

3.6 Final remarks

As mentioned earlier, the conductivity framework was preferred mostly for simplification purposes. This entails no severe loss of generality in the chapter's results if *elasticity* is considered instead; it just would mean that cumbersome calculations had to be performed, as the algebraic entities involved get more complex.

The one exception are the results of Sections 3.4 and 3.5; more precisely, every assertion depending on the explicit knowledge of G_θ. The reason is that a counterpart of Theorem 3.1.6 does not exist in elasticity theory: the characterization of the G-closure of mixtures between two homogeneous *isotropic materials* (see Section 4.1) is a famous open problem in homogenization theory.

Chapter 4

A generalized notion of compliance

It is common in structural optimization to look for stiff structures (*i.e.* structures that do not deform much): beams that do not bend much, bridges that can withstand heavy loads, and so on [71]. This is usually achieved by minimizing a quantity commonly termed in engineering by *compliance* (it is to be understood as the opposite of *stiffness*); the quantity measuring compliance, however, depends on the conditions imposed on the structure and situations arise where it is not clear how to define such a measure. It is the purpose of the present chapter to address this issue.

Section 4.1 sets the framework. Section 4.2 recovers a couple of well known cases and provides the suitable motivation to the subsequent introduction of the *generalized compliance*. In Section 4.3 the computation of derivatives with respect to structural parameters is performed, as they are needed in the numerical simulations which close the chapter.

On notations

Unlike the previous sections, a tensor's components will be here denoted with subscripts. The notation for inner products will also change: $u \cdot v$ for given vectors $u, v \in \mathbb{R}^n$. This is to be suggestive of the tensorial order involved (one dot for vectors, which are first order tensors), as the *Frobenius inner product* for square matrices (second order tensors) is written with two dots, $A : B = \mathrm{tr}(A^T B) = \sum_{i,j=1}^{n} A_{ij} B_{ij}$.

Like in the first chapter, the *comma notation* for derivatives is kept. If $A : \Omega \to \mathbb{R}^{n,n}$ is a matrix-valued function defined on an open set $\Omega \subset \mathbb{R}^n$, $\mathrm{div}(A)$ stands for the vector field with components $\mathrm{div}(A)_i = \sum_{j=1}^{n} A_{ij,j}$.

4.1 Preliminaries

The equations modelling the physical processes of a deformable body undergoing loading conditions, unlike the ones considered in the previous chapter, are of a vector nature; for instance, an applied load is characterized not only by its magnitude, but also by the direction (a vector) along which it is enforced. Changes of shape being involved, it is not surprising that meaningful quantities describing physical aspects of such a body depend on the vector field u of *displacements* (meaning, for each particle, the difference between its final and initial positions).

Owing to deformation, internal forces appear within the body. In general, these *stresses* are not uniformly distributed and may differ pointwise; they are discribed by a second order symmetric tensor, the *Cauchy stress tensor* σ. Also, a *constitutive law* must be considered, that is, an equation describing how the stress relates with the *strain* (measure of relative deformation, represented also by a second order symmetric tensor).

A frequent scenario in applications is that of *linear elasticity.*[1] The hypothesis of infinitesimal strains ("small" deformations) is one of the basic principles in this framework; the *strain tensor* is taken to be the symmetric part $\varepsilon(u)$ of the displacement's jacobian matrix: $\varepsilon_{ij}(u) = (u_{i,j} + u_{j,i})/2$, for all $i, j \in \{1, \dots, n\}$. The other basic principle is that stresses and strains are linearly related, or in simpler terms: stress is proportional to strain; the constitutive law is then written

$$\sigma_{ij} = \sum_{k,l=1}^{n} E_{ijkl}\varepsilon_{kl} \quad \textit{for all } i, j, k, l \in \{1, \dots, n\}. \tag{4.1}$$

The *elastic coefficients* E_{ijkl} are the components of a *fourth order tensor* (in this case playing the role of a linear application on the space of second order symmetric tensors) and they represent material properties of the substance(s) from which the solid body is made of; for physical reasons, they must satisfy some symmetry relations:

$$E_{ijkl} = E_{ijlk} = E_{jikl} = E_{klij}.$$

A tensor $E \in \mathbb{R}^{n,n,n,n}$ whose coefficients verify the above set of equalities will be termed an *elastic tensor*, and one writes $E \in \mathbb{E}$. An important example of elastic tensor is the one representing a homogeneous *isotropic material*:

$$E_{ijkl} = \lambda\delta_{ik}\delta_{jl} + \mu(\delta_{ij}\delta_{kl} + \delta_{il}\delta_{kj}) \quad \textit{for all } i, j, k, l \in \{1, \dots, n\}, \tag{4.2}$$

where δ_{ij} is the *Kronecker delta* ($\delta_{ij} = 1$ if $i = j$, zero otherwise) and the constants λ, μ are the so-called *Lamé coefficients*. In this case, (4.1) reads

$$\sigma = \lambda \operatorname{tr}(\varepsilon) I + 2\mu\varepsilon. \tag{4.3}$$

4.1.1 Note. A material is *isotropic* when, roughly, it has the same properties in all directions (otherwise, it is called *anisotropic*). This is easy to express in conductivity: the material tensor is just a scalar multiple of the identity, $A(x) = a(x)I$; in elasticity theory one needs the more complicated relation (4.2), where eventually $\lambda = \lambda(x)$ and $\mu = \mu(x)$ in the case of a heterogeneous body. ■

Given an open and bounded *design domain* $\Omega \subset \mathbb{R}^n$ (which may be subject to body loads), whose boundary $\partial\Omega$ is partitioned into disjoint parts Γ_N (where some surface loads may be applied) and Γ_D (where some displacements are prescribed), the general linear elasticity problem can be mathematically formulated as:

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = f & \textit{in } \Omega, \\ u = \bar{u} & \textit{on } \Gamma_D, \\ E\varepsilon(u)\nu = g & \textit{on } \Gamma_N, \end{cases} \tag{4.4}$$

[1] Recall that *elasticity* is the field of mechanics that studies the properties of deformable bodies able to return to their "rest" shape when external stimuli are withdrawn.

with ν denoting the outward unit normal to $\partial\Omega$. Like in the previous chapter, this problem is to be understood in the *weak* sense; more exactly, given $f \in L^2(\Omega;\mathbb{R}^n)$, $g \in L^2(\Gamma_N;\mathbb{R}^n)$ and $u \in H^{1/2}(\Gamma_D;\mathbb{R}^n)$, by solution of (4.4) one means a solution of the *variational problem*

$$\begin{cases} \text{find } u \in \mathcal{U} \text{ such that} \\ \displaystyle\int_\Omega E\varepsilon(u) : \varepsilon(v)\,dx = \int_\Omega f \cdot v\,dx + \int_{\Gamma_N} g \cdot v\,ds \quad \text{for all } v \in V, \end{cases} \tag{4.5}$$

where $\mathcal{U}$ is the affine space of functions in $H^1(\Omega)$ whose trace on Γ_D is equal to $\bar{u}$, and V is the linear space of functions in $H^1(\Omega)$ whose trace on Γ_D is zero.

Existence and uniqueness theory for this kind of problems is very similar to that of scalar ones in conductivity. The same is true regarding *homogenization theory*; in fact, all the concepts and results in Section 3.1 (with the exception of Theorem 3.1.6) have a foreseeable counterpart in the current setting. All these aspects and several others can be found in the book of Oleinik *et al.* [82].

4.1.2 Note. Even when $u \in H^2(\Omega;\mathbb{R}^n)$, since one only has $E \in L^\infty(\Omega;\mathbb{E})$, the stresses $Ee(u)$ do not have a well defined trace on $\partial\Omega$; so, even if Ω has a lipschitzian boundary, in which case $\nu \in L^\infty(\partial\Omega;\mathbb{R}^n)$, the third equation in (4.4) is just a formal way of stating the Neumann condition implicitly contained in (4.5).

Such regularity issues are bypassed, as they are not essential to the goal pursued in the chapter; therefore, some of the integrals appearing in the text (and other ones implied when integration by parts is called for) should be regarded merely in a formal sense. ■

4.2 Measuring compliance

The two more familiar situations are described first, as they will serve to motivate the generalized measure introduced in the bigger picture.

Zero Dirichlet boundary conditions

Start by considering a linearly elastic structure *clamped* on Γ_D:

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_D, \\ E\varepsilon(u)\nu = g & \text{on } \Gamma_N. \end{cases} \tag{4.6}$$

In this formulation, only zero Dirichlet boundary conditions are allowed (or no Dirichlet boundary conditions appear at all, when $\Gamma_D = \varnothing$).

One is interested in optimizing for stiffness a structure defined by the domain Ω and by the elasticity tensor E. Note that high stiffness is equivalent to low compliance. It does not matter whether one is talking about shape optimization (when the shape of Ω varies), free material optimization (when the elastic tensor E is the optimization parameter), or any other kind of optimization.

The most common way to evaluate the compliance of an elastic structure is through the *work done by the applied loads*

$$\mathcal{W} = \int_\Omega f \cdot u\,dx + \int_{\Gamma_N} g \cdot u\,ds$$

It is not difficult to see that the smaller the work $\mathcal{W}$, the stiffer the structure. This is so because f and g are fixed (they are data of the problem); thus, the work done by them is small if the displacement u is small, that is, if the structure does not deform much.

It is a simple exercise to check that $\mathcal{W} = 2\mathcal{E}$, where $\mathcal{E}$ is the *elastic energy* stored in the body:

$$\mathcal{E} = \frac{1}{2}\int_\Omega E\varepsilon(u) : \varepsilon(u)\,dx. \tag{4.7}$$

It suffices to take u as a test function in the variational formulation of (4.6):

$$\begin{cases} \textit{find } u \in V \textit{ such that} \\ \displaystyle\int_\Omega E\varepsilon(u) : \varepsilon(v)\,dx = \int_\Omega f \cdot v\,dx + \int_{\Gamma_N} g \cdot v\,ds \quad \textit{for all } v \in V. \end{cases}$$

Structures subject to a prescribed displacement

The extreme opposite case is when no loads are applied at all, the deformation being caused by nonzero Dirichlet boundary conditions:

$$\begin{cases} -\operatorname{div}[E\varepsilon(u)] = 0 & \textit{in } \Omega, \\ u = \bar{u} & \textit{on } \Gamma_D, \\ E\varepsilon(u)\nu = 0 & \textit{on } \Gamma_N. \end{cases} \tag{4.8}$$

The description of a stiff structure as one that "does not deform much" is no longer valid: the displacement is prescribed on Γ_D. A structure described by problem (4.8) should be called "stiff" if the effort it takes to impose the displacement $\bar{u}$ on Γ_D is large; that is, the work done by the force $g = E\varepsilon(u)\nu$,

$$\int_{\Gamma_D} E\varepsilon(u)\nu \cdot \bar{u}\,ds, \tag{4.9}$$

should be *large*. This is quite different from the situation with problem (4.6), where for a structure to be stiff the work done by the applied loads had to be *small*. Note that g cannot be called "applied load" now, since it is not a datum of problem (4.8).

Again, it is easy to check that the quantity (4.9) is equal to twice the elastic energy $\mathcal{E}$ stored in the body. It suffices to integrate by parts in (4.7) and use the boundary conditions in (4.8).

In this case then, one should *maximize* the stored elastic energy in order to obtain a stiff structure, which is the opposite of the first situation described in this section. This phenomenon has already been observed in the literature [83, Rem. 9] [84, Rem. 11] [85].

Loads and prescribed displacement

Consider now the general case (4.4), when loads are applied (f in Ω, g on Γ_N) and a nonzero displacement $\bar{u}$ is prescribed on Γ_D.

How can one define the notion of stiff structure (or its reverse, the compliance) in this context? Should one minimize or maximize the stored elastic energy $\mathcal{E}$?

Claim. *It is proposed the difference*

$$\mathcal{C} = \int_\Omega f \cdot u \, dx + \int_{\Gamma_N} g \cdot u \, ds - \frac{1}{2} \int_\Omega E\varepsilon(u) : \varepsilon(u) \, dx$$

as a measure of compliance for structures subject to general boundary conditions, described by problem (4.4).

The previous quantity will be called *generalized compliance*. It is clear that $\mathcal{C}$, $\mathcal{W}/2$ and $\mathcal{E}$ are equal for structures governed by (4.6). On the other hand, for structures governed by (4.8), $\mathcal{C}$ and $\mathcal{E}$ are equal in modulus and have opposite signs. So, for any boundary conditions, one should minimize $\mathcal{C}$ in order to obtain a stiff structure. It should be noted that an independent work by Niu, Xu and Cheng [86] points to the same conclusion.

4.3 Computation of sensitivities

In Section 4.4 a comparison will be made (by means of numerical tests) between the objective functionals $\mathcal{W}$, $\mathcal{E}$ and $\mathcal{C}$, in terms of how adequately they measure the performance of a structure governed by (4.4). Since a gradient algorithm will be used, the derivative of the objective functional with respect to some structural parameter – which, for mere convenience, will be assumed to be the material tensor itself – has to be determined.

Differentiation of functionals depending on a "control" parameter (the elastic tensor E in this case) through the solution of some differential equation, depends obviously on the differentiability of that solution itself with respect to the "control". This kind of results are well known from *optimal control theory* and can, for instance, be found in the work of Chenais [89, Th. 4.2]. The map $E \mapsto u$, which associates to a given tensor $E \in L^\infty(\Omega; \mathbb{E})$ the solution u of (4.5), is henceforth assumed to be differentiable.

Because f and g are data of problem (4.4), hence they do not depend on E, the derivatives of $\mathcal{W}$, $\mathcal{E}$ and $\mathcal{C}$ at E in a direction δE are given, respectively, by

$$\delta\mathcal{W} = \int_\Omega f \cdot \delta u \, dx + \int_{\Gamma_N} g \cdot \delta u \, ds, \tag{4.10}$$

$$\delta\mathcal{E} = \frac{1}{2} \int_\Omega \delta E\varepsilon(u) : \varepsilon(u) \, dx + \int_\Omega E\varepsilon(\delta u) : \varepsilon(u) \, dx \tag{4.11}$$

and

$$\delta\mathcal{C} = \int_\Omega f \cdot \delta u \, dx + \int_{\Gamma_N} g \cdot \delta u \, ds - \frac{1}{2} \int_\Omega \delta E\varepsilon(u) : \varepsilon(u) \, dx - \int_\Omega E\varepsilon(\delta u) : \varepsilon(u) \, dx, \tag{4.12}$$

where δu stands for the derivative of u at E in the direction δE; note that, because $\bar{u}$ is fixed, it vanishes upon differentiation, thus yielding $\delta u = 0$ on Γ_D. Of course, to obtain a final formula in terms of δE only, it is essential to eliminate δu from these expressions since, being u the solution of (4.5), it depends in a highly implicit manner on E. Amazingly, in the case of the generalized compliance $\mathcal{C}$, the terms involving δu disappear altogether from (4.12): multiplication of the state equation in (4.4) by δu and integration by parts gives

$$\int_\Omega E\varepsilon(u) : \varepsilon(\delta u) \, dx = \int_\Omega f \cdot \delta u \, dx + \int_{\Gamma_N} g \cdot \delta u \, ds;$$

therefore, (4.12) evaluates simply to

$$\delta\mathcal{C} = -\frac{1}{2}\int_\Omega \delta E\varepsilon(u) : \varepsilon(u)\,dx. \tag{4.13}$$

However, the same reasoning does not prove to be effective either for the work $\mathcal{W}$ of the applied loads or for the stored energy $\mathcal{E}$; with those objective functionals one has to make use of the *adjoint method*, a method whose technical aspects will be kept aside for symplicity's sake; they can be found throughout the literature, for instance in the books of Allaire [87, Sec. 4.3] and Céa [88, Chap. 5], or in a paper by Chenais [89, Th. 4.1].

The procedures that follow amount basically to the adjoint method, performed in a somewhat informal (but perhaps more telling) fashion. First observe that by differentiating (4.5), with respect to E and in a direction δE, one ends up with the variational problem

$$\begin{cases} \textit{find } \delta u \in V \textit{ such that} \\ \displaystyle\int_\Omega E\varepsilon(\delta u) : \varepsilon(v)\,dx = -\int_\Omega \delta E\varepsilon(u) : \varepsilon(v)\,dx \quad \textit{for all } v \in V. \end{cases} \tag{4.14}$$

This problem alone will not suffice to get rid of δu in the derivatives (4.10) and (4.11); to complete that task, the so-called *adjoint problem* has to be introduced. The idea is the following: the terms involving δu can be thought of as the value, at δu, of a linear form defined over V; they are thus prone to be rewritten in terms of an auxiliary state variable $p \in V$, termed the *adjoint state*. The key aspect is that both p and δu qualify as test functions; more precisely, δu in the adjoint problem and p in (4.14).

When the work $\mathcal{W}$ is considered, the adjoint state p is defined as the solution of:

$$\begin{cases} \textit{find } p \in V \textit{ such that} \\ \displaystyle\int_\Omega E\varepsilon(p) : \varepsilon(v)\,dx = \int_\Omega f \cdot v\,dx + \int_{\Gamma_N} g \cdot v\,ds \quad \textit{for all } v \in V, \end{cases} \tag{4.15}$$

which is nothing more than the variational formulation of

$$\begin{cases} -\operatorname{div}[E\varepsilon(p)] = f & \textit{in } \Omega, \\ p = 0 & \textit{on } \Gamma_D, \\ E\varepsilon(p)\nu = g & \textit{on } \Gamma_N. \end{cases} \tag{4.16}$$

Since δu and p both belong to V, one concludes, first by (4.15) and then by (4.14), that

$$\int_\Omega f \cdot \delta u\,dx + \int_{\Gamma_N} g \cdot \delta u\,ds = \int_\Omega E\varepsilon(p) : \varepsilon(\delta u)\,dx = -\int_\Omega \delta E\varepsilon(u) : \varepsilon(p)\,dx,$$

that is, the derivative (4.10) writes as

$$\delta\mathcal{W} = -\int_\Omega \delta E\varepsilon(u) : \varepsilon(p)\,dx. \tag{4.17}$$

When the stored energy $\mathcal{E}$ is to be differentiated, it gives rise to the same adjoint state p, solution of (4.16). Note that, since u satisfies (4.5), the variational formulation (4.15) is equivalent to

$$\begin{cases} \textit{find } p \in V \textit{ such that} \\ \displaystyle\int_\Omega E\varepsilon(p) : \varepsilon(v)\,dx = \int_\Omega E\varepsilon(u) : \varepsilon(v)\,dx \quad \textit{for all } v \in V. \end{cases} \tag{4.18}$$

Similarly to the previous case, using (4.18) and then (4.14), and because δu and p are in the space V of test functions, it follows that

$$\int_\Omega E\varepsilon(\delta u) : \varepsilon(u)\, dx = \int_\Omega E\varepsilon(p) : \varepsilon(\delta u)\, dx = -\int_\Omega \delta E\varepsilon(u) : \varepsilon(p)\, dx;$$

hence, the derivative (4.11) can be reduced to

$$\delta\mathcal{E} = \int_\Omega \delta E\varepsilon(u) : \varepsilon\left(\frac{u}{2} - p\right) dx. \tag{4.19}$$

4.3.1 Remark. Regarding the derivative of $\mathcal{C}$, with u solution of (4.4), the adjoint method can be likewise followed, yielding in this case a null adjoint state $p = 0$. ■

4.3.2 Remark. The only difference between problems (4.4) and (4.16) is the space where the solutions are to be found ($\mathcal{U}$ and V, respectively); in other words, the only difference lies in the Dirichlet boundary condition. When no displacements are prescribed on the structure (see Section 4.2), the two problems are identical, which means that the minimization of either $\mathcal{W}$ or $\mathcal{E}$ is a self-adjoint problem. ■

4.3.3 Remark. It may seem strange that two different quantities, $\mathcal{W}$ and $\mathcal{E}$, give rise to the same adjoint state p, solution of problem (4.16). This is especially queer for structures with no prescribed displacements, where $\mathcal{W} = 2\mathcal{E}$ (see Section 4.2). Recall, however, that

$$\int_\Omega f \cdot u\, dx + \int_{\Gamma_N} g \cdot u\, ds = \int_\Omega E\varepsilon(u) : \varepsilon(u)\, dx$$

only when u satisfies problem (4.6). Otherwise, if u is some arbitrary function in V, there is no reason for the above relation to take place. ■

4.4 Numerical tests

The numerical simulations in the paper by Niu, Xu and Cheng [86, Sec. 4.2], demonstrate that minimizing the work of the applied loads (for a structure subject to loading conditions only) is equivalent to maximizing the stored elastic energy (for a structure subject to an equivalent imposed displacement); they also show clearly that both of those formulations can be recovered from the generalized compliance formulation. The main concern in the present section is to see what happens when mixed nonhomogeneous boundary conditions are present, and how each of the three measures $\mathcal{W}$, $\mathcal{E}$ and $\mathcal{C}$ describe the structure's behaviour in that case.

In order to make a comparison in the previous terms, we consider the problem depicted in Figure 4.1: a rectangular design domain $\Omega = [0, 6] \times [0, 3]$, clamped at the lower corners $A = [0, 0.3] \times \{0\}$ and $D = [5.7, 6] \times \{0\}$. A load $g = (0, -0.7)$ is uniformly distributed on the segment $B = [1.95, 2.05] \times \{0\}$, which is roughly equivalent to a concentrated force of $(0, -0.07)$, and a displacement $\bar{u} = (0, -1.47)$ is prescribed on $C = [3.95, 4.05] \times \{0\}$.

The values of g and $\bar{u}$ have been chosen to obtain a similar displacement on the segments B and C; this is, of course, aimed at mimicking a problem with equal applied loads on those segments, a case in which one has a clear picture on how a solution looks like.

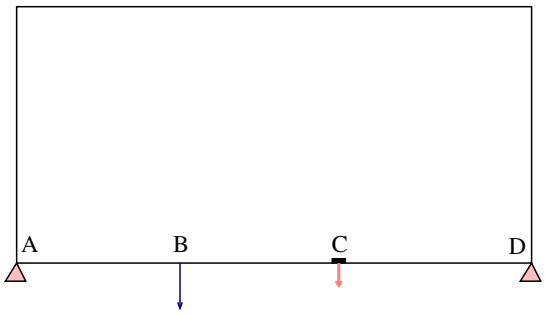

Figure 4.1: Design domain and boundary conditions of the problem.

As we shall be dealing with optimization via distribuition of isotropic material, the material tensor E will take the form $E = \rho E_0$, where ρ is a material density continuously varying within the interval $[0, 1]$ and E_0 is a fixed isotropic linearly elastic tensor, whose Lamé coefficients are $\lambda = 0.2$ and $\mu = 0.3$; according to (4.3),

$$E_0\varepsilon = 0.2\,\mathrm{tr}(\varepsilon)I + 0.6\varepsilon.$$

In this framework, the objective functional $\mathcal{W}$, $\mathcal{E}$, or $\mathcal{C}$, is dependent on the structural parameter ρ. Formulas (4.13), (4.17) and (4.19), reduce to

$$\delta\mathcal{C} = \int_\Omega \delta\rho \left[-\frac{1}{2}E_0\varepsilon(u) : \varepsilon(u)\right] dx,$$

$$\delta\mathcal{W} = \int_\Omega \delta\rho \left[-E_0\varepsilon(u) : \varepsilon(p)\right] dx$$

and

$$\delta\mathcal{E} = \int_\Omega \delta\rho \left[E_0\varepsilon(u) : \varepsilon\Big(\frac{u}{2} - p\Big)\right] dx,$$

respectively. The scalar function between square brackets will then be interpreted as the gradient of the objective functional with respect to ρ. Also, there is a constraint on the "volume" of material: it should fill 25% of the design domain, that is,

$$\int_\Omega \rho\, dx = 4.5;$$

obviously, the gradient of this constraint with respect to ρ is the constant function 1. Another constraint is that the pointwise "density" ρ should stay in [0.01,1]. Having identified the gradients of both the objective and the constraint, a descent algorithm for constrained optimization is applied, namely Algorithm 2.3.2.

Note that intermediate densities are not penalized: the algorithm is just set to run several iterations, until a reasonably well defined picture is obtained; one should then expect not a black-and-white design, but a greyscale density instead.

The implementation is made in the finite element object oriented language `FreeFem++` [90], with a fixed mesh of 16560 triangular elements; $\rho_0 = 0.25$. Visualization of results is made with the software `xd3d` [91].

The numerical results are presented in Figures 4.2–4.4, where the "density" ρ is represented through different levels of grey (1 is black, 0.01 is white). In Figure 4.2, the stored energy $\mathcal{E}$ is minimized; the obtained value is $\mathcal{E} = 0.03417205$. In Figure 4.3, the work $\mathcal{W}$ is minimized; the final value is $\mathcal{W} = 0.0617249$.

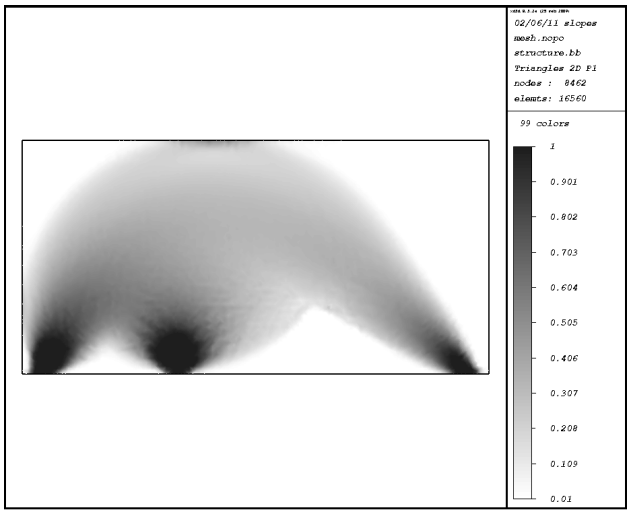

Figure 4.2: Density ρ obtained by minimization of the stored energy.

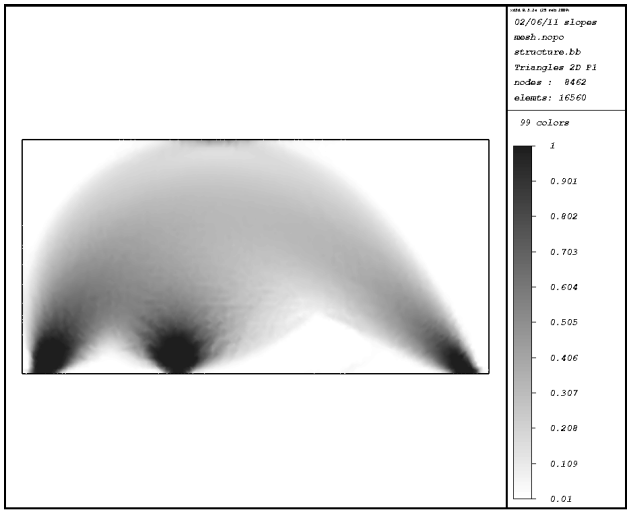

Figure 4.3: Density ρ obtained by minimization of the work done by the applied loads.

The distinctive feature to retain about both examples concerns the "blindness" of the functionals $\mathcal{W}$ and $\mathcal{E}$ towards the presence of the nonzero prescribed displacement! It is the reason why these measures are completely inadequate to represent compliance in the case of mixed *nonhomogeneous* boundary conditions.

Now in Figure 4.4, the generalized compliance $\mathcal{C}$ is minimized; its value for the optimized structure is $\mathcal{C} = -0.0123051$.

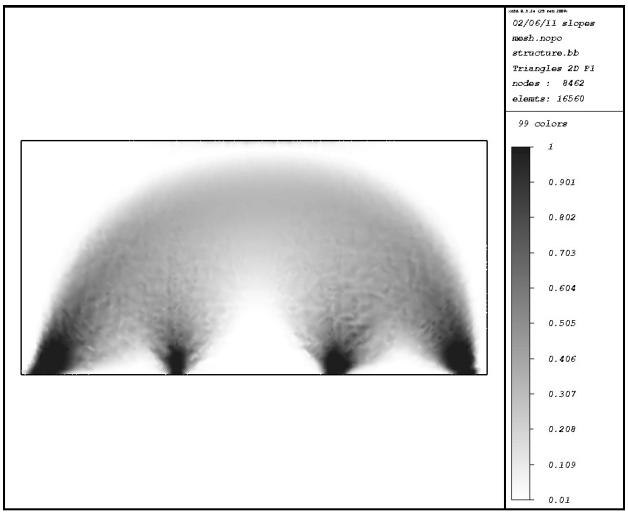

Figure 4.4: Density ρ obtained by minimization of the generalized compliance.

Unlike the other two functionals, the generalized compliance $\mathcal{C}$ is the only one that "takes advantage" of the prescribed displacement and gives rise to the expected solution.

4.5 Final remarks

A physical quantity was proposed for measuring the compliance of a structure in face of general boundary conditions, that is, allowing for nonzero applied loads together with nonzero prescribed displacements. Furthermore, that measure (the *generalized compliance*) does not require the computation of any adjoint state regarding its differentiation with respect to structural parameters (see Remark 4.3.1); this is an interesting feature from the computational point of view.

Though the "virtues" of the generalized compliance are made clear through a simple academic example, it would be quite interesting to obtain a design based on its minimization for a proper engineering problem and to test the resulting structure.

Bibliography

[1] S. Lopes, *Optimização estrutural com escolha livre de materiais*, Tese de Mestrado, Universidade de Lisboa, 2006.

[2] C. Barbarosie, S. Lopes, *Study of the cost functional for free material design problems*, Numerical Functional Analysis and Optimization, **29**, 115–125, 2008.

[3] C. Barbarosie, S. Lopes, *Properties of the cost functional in free material design*, to appear in Advances in Mathematical Sciences and Applications.

[4] C. Barbarosie, S. Lopes, *A generalized notion of compliance*, Comptes Rendus Mécanique, **339**, 641–648, 2011.

[5] J. Nocedal, S. Wright, *Numerical optimization (second edition)*, Springer, 2006.

[6] A. Cauchy, *Métode générale pour la résolution des systèmes d'équations simultanées*, Comptes Rendus de l'Académie des Sciences, **25**, 536–538, 1847.

[7] P.G. Ciarlet, *Introduction à l'analyse numérique matricielle et à l'optimisation*, Masson, 1990.

[8] J. Barzilai, J. Borwein, *Two point step size gradient methods*, IMA Journal of Numerical Analysis, **8**, 141–148, 1988.

[9] M. Raydan *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA Journal of Numerical Analysis, **13**, 321–326, 1993.

[10] R.L. Burden, J.D. Faires, *Numerical Analysis (ninth edition)*, Brooks/Cole, 2010.

[11] A. Friedlander, J.M. Martinez, M. Raydan, *A new method for large-scale box constrained convex quadratic minimization problems*, Optimization Methods and Software, **5**, 57–74, 1995.

[12] R. Fletcher, *Low storage methods for unconstrained optimization*, Lectures in Applied Mathematics (AMS), **26**, 165–179, 1990.

[13] L. Grippo, F. Lampariello, S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM Journal on Numerical Analysis, **23**, 707–716, 1986.

[14] L. Armijo, *Minimization of functions having Lipschitz continuous fisrt partial derivatives*, Pacific Journal of Mathematics, **16**, 1–3, 1966.

[15] M. Raydan, *The Barzilai and Borwein gradient method for the large-scale unconstrained minimization problem*, SIAM Journal on Optimization, **7**, 26–33, 1997.

[16] R. Fletcher, *On the Barzilai-Borwein method*, Numerical Analysis Report NA/207, University of Dundee, 2001.

[17] D. Luenberger, Y. Ye, *Linear and nonlinear programming (third edition)*, Springer, 2008.

[18] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, C.A. Sagastizábal, *Numerical optimization – Theoretical and practical aspects*, Springer, 2003.

[19] L. Nicolaescu, *Lectures on the Geometry of Manifolds*, World Scientific, 1996.

[20] C. Barbarosie, *Shape optimization of periodic structures*, Computational Mechanics, **30**, 235–246, 2003.

[21] J.M. Martinez, E.A. Pilotta, M. Raydan, *Spectral gradient methods for linearly constrained optimization*, Journal of Optimization Theory and Applications, **125**, 629–651, 2005.

[22] M.A. Diniz-Ehrhardt, M.A. Gomes-Ruggiero, J.M. Martinez, S.A. Santos, *Augmented lagrangian algorithms based on the spectral projected gradient for solving nonlinear programming problems*, Journal of Optimization Theory and Applications, **123**, 497–517, 2004.

[23] M.A. Gomes-Ruggiero, J.M. Martinez, S.A. Santos, *Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints*, SIAM Journal on Scientific Computing, **31**(3), 1628–1652, 2009.

[24] E.G. Birgin, J.M. Martinez, M. Raydan, *Nonmonotone spectral projected gradient methods on convex sets*, SIAM Journal on Optimization, **10**, 1196–1211, 2000.

[25] E.G. Birgin, J.M. Martinez, M. Raydan, *Algorithm 813: SPG - Software for convex constrained optimization*, ACM Transactions on Mathematical Software, **27**, 340–349, 2001.

[26] E.G. Birgin, J.M. Martinez, M. Raydan, *Inexact spectral projected gradient methods on convex sets*, IMA Journal of Numerical Analysis, **23**, 539–559, 2003.

[27] R.M. Chamberlain, M.J.D. Powell, C. Lemarechal, H.C. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Mathematical Programming Studies, **16**, 1–17, 1982.

[28] Y.H. Dai, H.C. Zhang, *Adaptive two-point stepsize gradient algorithm*, Numerical Algorithms, **27**, 377–385, 2001.

[29] L. Grippo, M. Sciandrone, *Nonmonotone globalization techniques for the Barzilai-Borwein gradient method*, Computational Optimization and Applications, **23**, 143–169, 2002.

[30] Z. Shi, S. Wang, *Modified nonmonotone Armijo line search for descent method*, Numerical Algorithms, **57**, 1–25, 2011.

[31] Y.H. Dai, R. Fletcher, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numerische Mathematik, **100**, 21–47, 2005.

[32] F. Luengo, M. Raydan, W. Glunt, T.L. Hayden, *Preconditioned spectral gradient method*, Numerical Algorithms, **30**, 241–258, 2002.

[33] L. Bello, M. Raydan, *Preconditioned spectral projected gradient method in convex sets*, Journal of Computational Mathematics, **23**, 225–232, 2005.

[34] J.P. Chehab, M. Raydan, *Implicit and adaptive inverse preconditioned gradient methods for nonlinear problems*, Applied Numerical Mathematics, **55**, 32–47, 2005.

[35] Y.H. Dai, J.Y. Yuan, Y.X. Yuan, *Modified two-point stepsize gradient methods for unconstrained optimization*, Computational Optimization and Applications, **22**, 103–109, 2002.

[36] Y.H. Dai, W.W. Hager, K. Schittkowski, H.C. Zhang, *The cyclic Barzilai-Borwein method for unconstrained minimization*, IMA Journal of Numerical Analysis, **26**, 604–627, 2006.

[37] Y.H. Dai, L.Z. Liao, *R-linear convergence of the Barzilai and Borwein gradient method*, IMA Journal of Numerical Analysis, **22**, 1–10, 2002.

[38] B. Rustem, M. Howe, *Algorithms for worst-case design and applications to risk management*, Princeton University Press, 2002.

[39] A. Ben-Tal, L. El Ghaoui, A. Nemirovski, *Robust Optimization*, Princeton University Press, 2009.

[40] M.M. Mäkelä, *Survey of bundle methods for nonsmooth optimization*, Optimization Methods and Software, **17**, 1–29, 2002.

[41] C. Charalambous, A.R. Conn, *An efficient method to solve the minimax problem directly*, SIAM Journal on Numerical Analysis, **17**, 162–187, 1978.

[42] S.P. Hahn, *Variable metric methods for minimizing a class of nondifferentiable functions*, Mathematical Programming, **20**, 1–13, 1981.

[43] R.S. Womersley, R. Fletcher, *An algorithm for composite nonsmooth optimization problems*, Journal of Optimization Theory and Applications, **48**, 493–523, 1986.

[44] A. Vardi *New minimax algorithm*, Journal of Optimization Theory and Applications, **75**, 613–634, 1992.

[45] G. Di Pillo, G. Grippo, L. Lucidi, *A smooth method for the finite minimax problem*, Mathematical Programming, **60**, 187–214, 1993.

[46] J.F. Sturm, S. Zhang, *A dual and interior-point approach to solve convex min-max problems*, in Minimax and Applications (D.Z. Du and P.M. Pardalos, eds.), Kluwer Academic, 69–78, 1995.

[47] Z. Zhu, X. Cai, J. Jian, *An improved SQP algorithm for solving minimax problems*, Applied Mathematical Letters, **22**, 464–469, 2009.

[48] E. Obasanjo, G. Tzallas-Regas, B. Rustem, *An interior-point algorithm for nonlinear minimax problems*, Journal of Optimization Theory and Applications, **144**, 291–318, 2010.

[49] I. Zang, *A smoothing technique for min-max optimization*, Mathematical Programming, **19**, 61–77, 1980.

[50] D.Q. Mayne, E. Polak, *Nondifferentiable optimization via adaptive smoothing*, Journal of Optimization Theory and Applications, **43**, 601–614, 1984.

[51] R.A. Polyak, *Smooth optimization methods for minimax problems*, SIAM Journal on Control and Optimization, **26**, 1274–1286, 1988.

[52] C. Gigola, S. Gomez, *A regularization method for solving finite convex min-max problems*, SIAM Journal on Numerical Analysis, **27**, 1621–1634, 1990.

[53] X. Li, *An entropy-based aggregate method for minimax optimization*, Engineering Optimization, **18**, 277–285, 1992.

[54] S. Xu, *Smoothing method for minimax problems*, Computational Optimization and Applications, **20**, 267–279, 2001.

[55] E. Polak, J.O. Royset, R.S. Womersley, *Algorithms with adaptive smoothing for finite minimax problems*, Journal of Optimization Theory and Applications, **119**, 459–484, 2003.

[56] F. Ye, H. Liu, S. Zhou, S. Liu, *A smoothing trust-region Newton-CG method for minimax problems*, Applied Mathematics and Computation, **199**, 581–589, 2008.

[57] E.Y. Pee, J.O. Royset, *On solving large-scale finite minimax problems using exponential smoothing*, Journal of Optimization Theory and Applications, **148**, 390–491, 2011.

[58] E. Polak, *Optimization: algorithms and consistent approximations*, Springer, 1997.

[59] E. Polak, R.S. Womersley, H.X. Yin, *An algorithm based on active sets and smoothing for discretized semi-infinite minimax problems*, Journal of Optimization Theory and Applications, **138**, 311–328, 2008.

[60] J.O. Royset, E.Y. Pee, *Rate of convergence analysis of discretization and smoothing algorithms for semiinfinite minimax problems*, to appear in Journal of Optimization Theory and Applications.

[61] M.J.D. Powell, *Direct search algorithms for optimization calculations*, Acta Numerica, **7**, 287–336, 1998.

[62] W. Hock, K. Schittkowski, *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, **187**, Springer-Verlag, 1981.

[63] L. Lukšan, J. Vlček, *Test problems for nonsmooth unconstrained and linearly constrained optimization*, Technical Report No. 798, Institute of Computer Science (Academy of Sciences of the Czech Republic), 2000.

[64] M. Haarala, K. Miettinen, M.M. Mäkelä, *New limited memory bundle method for large-scale nonsmooth optimization*, Optimization Methods and Software, **19**, 673–692, 2004.

[65] H. Attouch, G. Buttazzo, G. Michaille, *Variational analysis in Sobolev and BV spaces: applications to PDEs and optimization*, MPS-SIAM Series on Optimization, **6**, MPS-SIAM, 2006.

[66] L. Tartar, *Estimations of homogenized coefficients*, in Progress in Nonlinear Differential Equations and Their Applications, **31**: Topics in the mathematical modelling of composite materials (A. Cherkaev and R. Kohn, eds.), Birkhäuser, 9–20, 1997.

[67] F. Murat, L. Tartar, *H-convergence*, in Progress in Nonlinear Differential Equations and Their Applications, **31**: Topics in the mathematical modelling of composite materials (A. Cherkaev and R. Kohn, eds.), Birkhäuser, 21–43, 1997.

[68] F. Murat, L. Tartar, *Calculus of variations and homogenization*, in Progress in Nonlinear Differential Equations and Their Applications, **31**: Topics in the mathematical modelling of composite materials (A. Cherkaev and R. Kohn, eds.), Birkhäuser, 139–173, 1997.

[69] U. Ringertz, *On finding the optimal distribution of material properties*, Structural Optimization, **5**, 265–267, 1993.

[70] M.P. Bendsøe, J.M. Guedes, R. Haber, P. Pedersen, J.E. Taylor, *An analytical model to predict optimal material properties in the context of optimal structural design*, Journal of Applied Mechanics, **61**, 930–937, 1994.

[71] M.P. Bendsoe, O. Sigmund, *Topology optimization – theory, methods and applications*, Springer, 2003.

[72] L. Tartar, *Compensated compactness and applications to partial differential equations*, in Research Notes in Mathematics, **39**: nonlinear analysis and mechanics – Heriot Watt Symposium, Vol. IV (R.J. Knops, ed.), Pitman, 136–212, 1979.

[73] B. Dacorogna, *Direct methods in the calculus of variations (second edition)*, Applied Mathematival Sciences, **78**, Springer, 2010.

[74] G. Francfort, G. Milton, *Sets of conductivity and elasticity tensors stable under lamination*, Communications on Pure and Applied Mathematics, **47**, 257–279, 1994.

[75] R. Bhatia, *Matrix analysis*, Springer, 1997.

[76] G. Allaire, E. Bonnetier, G. Francfort, F. Jouve, *Shape optimization by the homogenization method*, Nümerische Mathematik, **76**, 27–68, 1997.

[77] A. Daniilidis, A. Lewis, J. Malick, H. Sendov, *Prox-regularity of spectral functions and spectral sets*, Journal of Convex Analysis, **15**, 547–560, 2008.

[78] M. Briane, A. Damlamian, P. Donato, *H-convergence for perforated domains*, in Research Notes in Mathematics, **391**: Nonlinear partial differential equations and their applications – Collège de France Seminar, Vol. XIII (D. Cioranescu and J.L. Lions, eds.), Pitman, 62–100, 1998.

[79] D. Cioranescu, A. Damlamian, P. Donato, L. Mascarenhas, *H^0-convergence as a limit-case of H-convergence*, Advances in Mathematical Sciences and Applications, **9**, 319–331, 1999.

[80] E. Hille, R.S. Philips, *Functional analysis and semigroups*, American Mathematical Society Colloquium Publications, **31**, 1957.

[81] R. Rockafellar, *Convex analysis*, Princeton University Press, 1970.

[82] O.O. Oleinik, A.S. Samaev, G.A. Yosifian, *Mathematical problems in elasticity and homogenization*, Studies in Mathematics and its Applications, **26** (J.L. Lions, G. Papanicolaou, H. Fujita and H.B. Keller, eds.), North-Holland, 1992.

[83] A.M. Toader, *The topological derivative for homogenized elastic coefficients of periodic microstructures*, SIAM Journal on Control and Optimization, **49**, 1607–1628, 2011.

[84] C. Barbarosie, A.M. Toader, *Shape and topology optimization for periodic problems, Part I: the shape and the topological derivative*, Structural and Multidisciplinary Optimization, **40**, 381–391, 2010.

[85] P. Pedersen, N.L. Pedersen, *Design objectives with non-zero prescribed support displacements*, Structural and Multidisciplinary Optimization, **43**, 205–214, 2011.

[86] F. Niu, S. Xu, G. Cheng, *A general formulation of structural topology for maximizing structural stiffness*, Structural and Multidisciplinary Optimization, **43**, 561–572, 2011.

[87] G. Allaire, *Conception optimale de structures*, Springer, 2007.

[88] J. Céa, *Optimisation, théorie et algorithmes*, Dunod, 1971.

[89] D. Chenais, *Optimal design of midsurface of shells: differentiability proofs and sensitivity computation*, Applied Mathematics and Optimization, **16**, 93–133, 1987.

[90] O. Pironneau, F. Hecht, A. Le Hyaric, *FreeFem++ home page*, URL: `http://www.freefem.org/ff++/index.htm`.

[91] F. Jouve, *xd3d home page*, URL: `http://www.cmap.polytechnique.fr/~jouve/xd3d/`.