Universidade Federal de Santa Catarina
Centro de Ciências Físicas e Matemáticas
Programa de Pós-Graduação em Matemática Pura e Aplicada

Marina Geremia

Relative-error inexact versions of Douglas-Rachford and ADMM splitting algorithms

Florianópolis
2020

Marina Geremia

Relative-error inexact versions of Douglas-Rachford and ADMM splitting algorithms

Tese submetida ao Programa de Pós-Graduação em Matemática Pura e Aplicada da Universidade Federal de Santa Catarina para a obtenção do título de Doutora em Matemática Pura e Aplicada, com área de concentração em Matemática Aplicada.
Orientador: Prof. Maicon Marques Alves, Dr.

Florianópolis
2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Geremia, Marina
Relative-error inexact versions of Douglas-Rachaford and ADMM splitting algorithms / Marina Geremia ; orientador, Maicon Marques Alves , 2021.
90 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Programa de Pós-Graduação em Matemática Pura e Aplicada, Florianópolis, 2021.

Inclui referências.

1. Matemática Pura e Aplicada. 2. Douglas-Rachford. 3. ADMM. 4. Inércia. 5. Relaxação. I. , Maicon Marques Alves. II. Universidade Federal de Santa Catarina. Programa de Pós Graduação em Matemática Pura e Aplicada. III. Título.

Marina Geremia

Relative-error inexact versions of Douglas-Rachford and ADMM splitting algorithms

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. José Mario Martínez Pérez, Dr.
Universidade Estadual de Campinas - UNICAMP

Prof. Alexandre Madureira, Dr.
Laboratório Nacional de Computação Científica - LNCC

Prof. Antonio Carlos Gardel Leitão, Dr.
Universidade Federal de Santa Catarina - UFSC

Prof. Vinicius Albani, Dr.
Universidade Federal de Santa Catarina - UFSC

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Doutora em Matemática Pura e Aplicada, com área de concentração em Matemática Aplicada.

ICP-Edu
Documento assinado digitalmente
Daniel Goncalves
Data: 09/02/2021 13:44:06-0300
CPF: 017.022.459-77

Prof. Daniel Gonçalves, Dr.
Coordenador do Programa de Pós-Graduação

ICP-Edu
Documento assinado digitalmente
Maicon Marques Alves
Data: 10/02/2021 16:33:13-0300
CPF: 967.631.910-49

Prof. Maicon Marques Alves, Dr.
Orientador

Florianópolis, 2020.

For my husband Cristian.

Agradecimentos

Muitas foram as pessoas envolvidas nesta caminhada, a quem agradeço com profundo respeito e admiração. Início agradecendo, minha querida família pelo apoio e carinho. Em especial, meu amado esposo Cristan, por todo o amor, incentivo e apoio.

Ao orientador e amigo Prof. Dr. Maicon Marques Alves meus sinceros agradecimentos por sua imensa competência, serenidade, paciência, amizade, confiança, incentivo, conhecimentos e experiências compartilhadas e, especialmente, pela constante e incansável solicitude ao longo de todo o trabalho.

Aos amigos Aline Becher, Eduardo Pandini e Daniela Losso pela amizade, apoio, conhecimentos, motivação, troca de experiências, bons momentos de convivência e partilha de angústias e de doces.

Ao amigo Raul Tintaya Marcavilacca pela amizade, discussões, colaborações e bons momentos de convivência.

Ao amigo Gabriel Mathias Ferrari pela amizade, confiança, incentivo e apoio.

À amiga Daiane Farias pela amizade, carinho, pelos muitos momentos de descontração e sua inspiradora motivação e bondade.

Aos fofinhos Benjamin e Emanuel pela alegria, carinho, pureza, momentos partilhados e por terem tornado os finais de aula na sexta-feira belos e alegres.

Aos professores membros da banca por aceitarem o convite e pelas contribuições ao trabalho.

Aos professores e funcionários do Programa de PósGraduação em Matemática Pura e Aplicada da UFSC pela dedicação e apoio.

Ao Instituto Federal de Santa Catarina (IFSC) pelo suporte financeiro.

Resumo

Nesta tese, propomos e analisamos novas versões do método Douglas-Rachford splitting (DRS) para operadores monótonos maximais e do alternating direction method of multipliers (ADMM) para otimização convexa. Inicialmente, apresentamos um método Douglas-Rachford splitting (DRS) inexato e um método Douglas-Rachford-Tseng forward-backward (F-B) splitting para resolver inclusões monótonas de dois e quatro operadores, respectivamente. Provamos complexidade computacional em iteração, tanto no sentido pontual quanto no sentido ergódico, mostrando que ambos os algoritmos admitem duas iterações diferentes: uma que pode ser incorporada ao hybrid proximal extragradient (HPE) method de Solodov e Svaiter, para a qual a complexidade em iteração é conhecida desde o trabalho de Monteiro e Svaiter, e outra que exige uma análise em separado. Em seguida, estudamos o comportamento assintótico de novas variantes dos algoritmos DRS e ADMM, ambos com efeito de relaxação e inércia, e com critério de erro relativo para os subproblemas. Por fim, com objetivo de demonstrar a aplicabilidade dos métodos propostos neste trabalho, realizamos experimentos numéricos aplicando nosso método ADMM (relaxado e com inércia) aos problemas LASSO e regressão logística.

Palavras-chave: ADMM. Algoritmos de decomposição. Algoritmo de ponto proximal. Complexidade. Critério de erro relativo. Método de Douglas-Rachford splitting inexato. Método HPE. Métodos inerciais. Método do tipo Tseng forward-backward. Operadores monótonos. Relaxação.

Resumo Expandido

Introdução

Seja $\mathcal{H}$ um espaço de Hilbert com produto interno $\langle\cdot,\cdot\rangle$ e induzido pela norma $\|\cdot\| := \sqrt{\langle\cdot,\cdot\rangle}$.

Um operator $T : \mathcal{H} \rightrightarrows \mathcal{H}$ é dito monótono se

$$\langle z - z', v - v'\rangle \geq 0, \quad \forall v \in T(z), \quad \forall v' \in T(z'). \tag{1}$$

Por outro lado, T é um operador monótono maximal se T é monótono e maximal na família dos operadores monótonos em $\mathcal{H}$. A teoria de operadores monótonos e monótonos maximais desempenha um papel central na análise não linear e, como consequência, possui aplicações em diversas áreas como, por exemplo, análise funcional, engenharia, física matemática, matemática aplicada e otimização.

Um problema de inclusão monótona (MIP) consiste em encontrar $z \in \mathcal{H}$ tal que

$$0 \in T(z) \tag{2}$$

onde $T : \mathcal{H} \rightrightarrows \mathcal{H}$ é um operador monótono maximal para o qual o conjunto solução de (2) é não vazio. Uma grande variedade de problemas de diferentes campos da matemática aplicada e da otimização podem ser descritos por (2).

Um dos algoritmos mais populares para encontrar soluções aproximadas do MIP (2) é o método do ponto proximal (PP), que desenvolvido inicialmente por Martinet [60] para resolver desigualdades variacionais monótonas (com operadores ponto-a-ponto) e posteriormente estudado e desenvolvido por Rockafellar para um contexto mais geral de operadores monótonos maximais. É bem conhecido que a aplicabilidade prática de esquemas numéricos que estão baseados no cálculo exato de resolventes de operadores monótonos depende fortemente de estratégias que permitem cálculos inexatos. Este é o caso do algoritmo de ponto proximal. Em seu trabalho pioneiro [74], Rockafellar provou que soluções de (2) podem ser obtidas utilizando o critério de erro somável. Este resultado encontrou aplicações importantes no projeto e análise de muitos algoritmos práticos para resolver problemas desafiadores em otimização e campos relacionados.

Muitas versões modernas inexatas do método ponto proximal usam tolerâncias de erro relativas, em oposição ao critério de erro somável, para resolver os subproblemas associados. Os primeiros métodos deste tipo foram propostos por Solodov e Svaiter em [77, 76]. Entre esses novos métodos, o *hybrid proximal extragradient* (HPE) *method* [76], proposto por Solodov e Svaiter em 2000, tem se destacado como uma estrutura eficaz para o projeto e análise de muitos algoritmos concretos (por exemplo, [14, 23, 38, 51, 52, 57, 64, 65, 68, 76, 78, 79, 63]). Em 2010, a complexidade de iteração do método HPE foi provada por Monteiro e Svaiter em [67].

Neste trabalho, além do MIP (2), consideramos o problema de inclusão monótona de operadores que consiste em encontrar $z \in \mathcal{H}$ tal que

$$0 \in A(z) + B(z) \tag{3}$$

bem como o MIP de quatro operadores que consiste em encontrar $z \in \mathcal{H}$ tal que

$$0 \in A(z) + C(z) + F_1(z) + F_2(z) \tag{4}$$

onde A, B e C são operadores (ponto-conjunto) monótonos maximais em $\mathcal{H}$, $F_1 : D(F_1) \to \mathcal{H}$ é Lipschitz contínuo e $F_2 : \mathcal{H} \to \mathcal{H}$ é cocoersivo. Os problemas (3) e (4) aparecem em diferentes campos da matemática aplicada e otimização, incluindo otimização convexa, processamento de sinais, PDEs, problemas inversos, entre outros [11, 48]. Note que, impondo condições suaves aos operadores C, F_1 e F_2, o problema (4) se torna um caso particular de (3) com $B := C + F_1 + F_2$.

Um dos algoritmos mais populares na atualidade para obter soluções aproximadas do MIP (3) é o *Douglas-Rachford splitting* (DRS) *method*. Proposto originalmente por Douglas e Rachford (1954), em [35], para resolução de problemas com operadores lineares, o método DRS foi generalizado por Lions e Mercier (1979), em [56], para operadores monótonos maximais não lineares gerais. O método DRS consiste em um método iterativo no qual, a cada iteração, os resolventes dos operadores A e B são empregados separadamente ao invés do resolvente do operador completo, $A+B$, que pode ser caro para calcular numericamente. Deste modo, resolvemos sequencialmente dois subproblemas regularizados (proximais) em substituição à resolução do problema completo.

Em 1992, Eckstein e Bertsekas provaram que o método DRS é um caso especial do algoritmo de ponto proximal, em [37]. Devido a este fato, grande parte da teoria do DRS, e sua instância especial chamada *alternating direction multiplier method*(ADMM) [45, 47], podem ser analisados dentro da teoria do ponto proximal. Além disso, a equivalência entre o método DRS e o método ponto proximal de Rockafellar esclarece a natureza proximal do método DRS, explica o fato de que o método DRS tem propriedades de convergência mais gerais do que outros algoritmos de separação proximal [36, 37] e permite a derivação de novas versões inexatas e relaxadas dos métodos DRS e ADMM.

Nos últimos anos, a área de *machine learning* tem atraído a atenção de diversos grupos de pesquisa, com considerável impacto nas mais diversas áreas aplicadas. Nesse contexto, o uso de algoritmos de *machine learning* em grandes conjuntos de dados estatísticos tem gerado um impacto significativo em muitas áreas, como, por exemplo, inteligência artificial, internet, biologia computacional, medicina, marketing, publicidade, análise de rede, logística, detecção de fraude, opinião mineração e economia, entre outros [17]. Esta vasta gama de problemas podem ser reescritos no quadro geral de otimização convexa, dado por

$$\min_{x \in \mathbb{R}^n} \{f(x) + g(x)\} \tag{5}$$

onde f, g são funções convexas em $\mathbb{R}^n$.

O método de multiplicadores de direção alternada (ADMM) [45, 47] é um algoritmo de primeira ordem simples, porém poderoso, para resolver (5), que ganhou popularidade durante a última década, em grande parte devido à seu amplo leque de aplicações em ciência de dados. Embora ideias semelhantes tenham aparecido ainda no início da década de 1950 (ver, por exemplo, [17]), o ADMM foi apresentado pela primeira vez em meados da década de 1970 por Gabay, Mercier, Glowinski e Marroco [45, 47].

Em [46], Gabay apresentou, pela primeira vez, o ADMM como uma aplicação do método DRS para resolução do seguinte problema de inclusão monótona

$$0 \in \partial f(x) + \partial g(x) \tag{6}$$

que é, em particular, um caso especial de (3) com $A = \partial g$ e $B = \partial f$ (ou *vice-versa*). Portanto, sob condições de qualificação padrão, o problema (6) é equivalente a (5). Essa percepção foi crucial para a obtenção de alguns dos resultados apresentados por Eckstein em [36, 37] e, como conseqüência, para obter os algoritmos derivados em [40] e os algoritmos propostos por esta tese. Ao longo deste trabalho, assumimos que (6) admite pelo menos uma solução.

A aceleração de método proximais tem sido objeto de intensa pesquisa nos últimos anos. Uma das principais características dos algoritmos inerciais proximais é que a iteração atual é definida a partir das duas últimas iterações. Os estudos que objetivam acelerar a convergência dos métodos proximais concentram-se, especialmente, na adição de informações de segunda ordem para atingir uma convergência ainda mais rápida. Uma das principais características do algoritmo inercial proximal é que a iteração atual é definida usando as informações das duas últimas iterações. Neste sentido, os algoritmos inerciais para otimização convexa e inclusões monótonas [2] aparecem em conexão com dinâmica contínua - ver, por exemplo, [2, 9, 10] - algoritmos de primeira e segunda ordem acelerados e métodos de divisão de operador - ver, por exemplo, [7, 8, 16, 26, 27, 30, 58] - com boas melhorias de desempenho teórico e prático em relação aos métodos anteriores.

Objetivos

- Propor e estudar a complexidade de iteração de um *inexact Douglas-Rachford splitting method* e um *Douglas-Rachford-Tseng's forward-backward (F-B) splitting method* para resolver (3) e (4), respectivamente.

- Desenvolver uma sequência de três algoritmos inexatos, inercias e relaxados, cada um baseado no anterior. O primeiro algoritmo é uma nova variante do algoritmo do ponto proximal para (2). Nosso método proposto é uma nova variante inercial do método de projeção proximal híbrida relaxada (HPP) de Solodov e Svaiter. Usando este primeiro algoritmo, desenvolvemos uma nova variante inexata do método *Douglas-Rachford* para resolver (3). Por fim, com base neste último método, derivamos uma nova variante inexata do algoritmo do método de multiplicadores de direção alternada (ADMM) para resolver problemas da forma (5).

- Ilustrar aplicabilidade dos métodos propostos neste trabalho a partir de experimentos numéricos, que consistem na aplicação do método ADMM (relaxado e com inércia), proposto nesta tese (Algoritmo 8), aos problemas LASSO e regressão logística.

Metodologia

Os limites de complexidade de iteração tanto do método *Douglas-Rachford splitting* inexato e quanto do método *Douglas-Rachford-Tseng's forward-backward (F-B) splitting* são obtidos no sentido pontual (não ergódico), bem como no sentido ergódico, mostrando que eles admitem duas iterações diferentes: uma que pode ser incorporada ao método HPE, para o qual a complexidade da iteração é conhecida desde o trabalho de Monteiro e Svaiter, e outra que exige uma análise separada. Por outro lado, o estudo do comportamento assintótico das novas variantes dos métodos *Douglas-Rachford splitting* e *ADMM splitting*, ambos sob efeitos de relaxação e inércia, que fazem uso do critério inexato (erro relativo) para resolver os subproblemas associados, está baseada essencialmente em uma nova versão inexata do algoritmo do ponto proximal, que também foi proposta nesta tese (Algoritmo 6), sob efeito inclui tanto de um passo inercial quanto de uma relaxação.

Resultados, Discussão e Considerações Finais

Nesta tese, propomos e analisamos algumas variantes do método Douglas-Rachford para resolução de inclusões monótonas e do método de direção alternada de multiplicadores para otimização convexa. Inicialmente, propomos e estudamos a complexidade de iteração de um *inexact Douglas-Rachford splitting method* e um *Douglas-RachfordTseng's forward-backward splitting method* para resolver inclusões monótonas de dois e quatro operadores, respectivamente. O primeiro método, Algoritmo 3, (embora baseado em um mecanismo ligeiramente diferente de iteração) foi motivado pelo trabalho recente de J. Eckstein e W. Yao, no qual um método DRS inexato é derivado de uma instância especial do *hybrid proximal extragradient* (HPE) *method* de Solodov e Svaiter, enquanto o segundo, Algoritmo 5, combina o método DRS inexato que propomos (Algoritmo 3) (usado como uma iteração externa) com um método do tipo *Tseng's forward-backward splitting* (usado como uma iteração interna) para resolver os subproblemas correspondentes. Na sequência, estudamos o comportamento assintótico de novas variantes dos métodos *Douglas-Rachford splitting* e *ADMM splitting*, ambos sob efeitos de relaxação e inércia, usando critério inexato (erro relativo) para resolver os subproblemas associados. Por fim, com objetivo de demonstrar a aplicabilidade dos métodos propostos neste trabalho, realizamos experimentos numéricos aplicando nosso método ADMM (relaxado e com inércia) aos problemas LASSO e regressão logística. Cabe ressaltar que obtivemos um desempenho computacional melhor do que os métodos ADMM inexatos apresentados anteriormente por [39, 40]. Além disso, nossos resultados numéricos indicam que as versões inexatas, propostas neste trabalho, são uma ferramenta útil para resolver de aplicações reais que podem ser descritas pelo quadro geral de otimização convexa.

Palavras-chave: ADMM. Algoritmos de decomposição. Algoritmo de ponto proximal. Complexidade. Critério de erro relativo. Método de Douglas-Rachford splitting inexato. Método HPE. Métodos inerciais. Método Tseng forward-backward. Operadores monótonos. Relaxação.

Abstract

In this thesis, we propose and analyze new versions of the Douglas-Rachford splitting (DRS) method for maximal monotone operators and the alternating direction method of multipliers (ADMM) for convex optimization. Firstly, we present an inexact Douglas-Rachford splitting (DRS) method and a Douglas-Rachford-Tseng's forward-backward (F-B) splitting method for solving two-operator and four-operator monotone inclusions, respectively. We prove iteration-complexity bounds for both algorithms in the pointwise (non-ergodic) as well as in the ergodic sense by showing that they admit two different iterations: one that can be embedded into the Solodov and Svaiter's hybrid proximal extragradient (HPE) method, for which the iteration-complexity is known since the work of Monteiro and Svaiter, and another one that demands a separate analysis. We also study the asymptotic behavior of new variants of the DRS and ADMM splitting methods, both under relaxation and inertial effects, and with inexact (relative-error) criterion for subproblems. To demonstrate the applicability of the proposed methods, we performed numerical experiments applying the ADMM (relaxed and inertial) on LASSO and logistic regression problems.

Keywords: ADMM. Complexity. HPE method. Inertia. Inexact Douglas-Rachford method. Monotone operators. Operator splitting. Proximal point algorithm. Relative error criterion. Relaxation. Splitting. Tseng's forward-backward method.

Contents

Introduction

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\|\cdot\| := \sqrt{\langle \cdot, \cdot \rangle}$. A set-valued map $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is a *monotone operator* if

$$\langle z - z', v - v' \rangle \geq 0, \quad \forall v \in T(z), \quad \forall v' \in T(z'). \tag{7}$$

On the other hand, T is a *maximal monotone operator* if T is monotone and $T = S$ whenever S is monotone on $\mathcal{H}$ and $T \subseteq S$.

According to [36], the definition of maximal monotone operator first appeared in [53]. The theory of monotone and maximal monotone operators plays a central role in nonlinear analysis and consequently has numerous applications in functional analysis, engineering, mathematical physics, applied mathematics, and optimization. An example of a monotone operator is the subdifferential operator ∂f, which is an effective tool in the analysis of nondifferentiable convex functions and algorithms in convex programming [11, 28, 36].

The pioneering work of Rockafellar [74, 75] clarified the role of monotone operators in the area of mathematical programming and established conditions for the maximal monotonicity of subdifferential mappings, the maximality of sum of two operators, among other relevant results. In 1976, he studied and developed the *proximal point* (PP) *algorithm* for solving the monotone inclusion problem (MIP), i.e., the problem of finding $z \in \mathcal{H}$ such that

$$0 \in T(z) \tag{8}$$

where $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is maximal monotone. A wide variety of problems of different fields of applied mathematics and optimization can be placed in the structure of (8). Moreover, this formulation is at the core of the modeling of inverse problems for solving diverse real-world problems; as for example, in phase retrieval, in sensor networks, in comprised tomography, and data compression [41]. An example of this is when $T = \partial f$, with f a convex function, which reduces problem (8) to the problem of minimizing f.

The term "proximal point" was originally coined in French by Moreau [70] at the beginning of the 1960s. The PP method was firstly proposed for solving variational inequalities, at the beginning of the 1970s, by Martinet [60] and later on popularized by the work of Rockafellar in [74, 75]. In its exact formulation, an iteration of the PP method can be described as

$$z^k = (\lambda_k T + I)^{-1} z^{k-1} \qquad \forall k \geq 1, \tag{9}$$

where $\lambda_k > 0$ is a stepsize parameter and z^{k-1} is the current iterate. In [74], Rockafellar proved that if, at each iteration $k \geq 1$, z^k is computed satisfying

$$\|z^k - (\lambda_k T + I)^{-1} z^{k-1}\| \leq e_k, \quad \sum_{k=1}^{\infty} e_k < \infty, \tag{10}$$

and $\{\lambda_k\}$ is bounded away from zero, then $\{z^k\}$ converges (weakly, in infinite dimensions) to a solution of (8). This result has found important applications in the design and analysis of many practical algorithms for solving challenging problems in optimization and related fields [36].

In this thesis, we consider the *two-operator* MIP of finding z such that

$$0 \in A(z) + B(z) \tag{11}$$

as well as the *four-operator* MIP, i.e., of finding z such that

$$0 \in A(z) + C(z) + F_1(z) + F_2(z) \tag{12}$$

where A, B, and C are (set-valued) maximal monotone operators on $\mathcal{H}$, $F_1 : D(F_1) \to \mathcal{H}$ is (point-to-point) *Lipschitz continuous* and $F_2 : \mathcal{H} \to \mathcal{H}$ is (point-to-point) *cocoercive* (see Section 2.2 for the precise statement). Problems (11) and (12) appear in different fields of applied mathematics and optimization, including convex optimization, signal processing, PDEs, inverse problems, among others [11, 48]. Under mild conditions on the operators C, F_1, and F_2, problem (12) becomes a special instance of (11) with $B := C + F_1 + F_2$.

One of the most popular algorithms for finding approximate solutions of (11) is the *Douglas-Rachford splitting* (DRS) *method*. It consists of an iterative method in which, at each iteration, the solutions of two regularized (proximal) subproblems are computed sequentially. In other words, at each iteration, the resolvents $J_{\gamma A} = (\gamma A + I)^{-1}$ and $J_{\gamma B} = (\gamma B + I)^{-1}$ of A and B, respectively, are employed separately instead of the resolvent $J_{\gamma(A+B)} = (\gamma(A+B)+I)^{-1}$ of the full operator $A+B$, which may be expensive to compute numerically.

An iteration of the method can be described by

$$z^k = J_{\gamma A}(2J_{\gamma B}(z^{k-1}) - z^{k-1}) + z^{k-1} - J_{\gamma B}(z^{k-1}) \qquad \forall k \geq 1, \tag{13}$$

where $\gamma > 0$ is a scaling parameter and z^{k-1} is the current iterate. The DRS method was originally proposed by Douglas and Rachford (1954), in [35], for the power-series analysis of a discretization of the heat equation and generalized by Lions and Mercier (1979), in [56], for general nonlinear maximal monotone operators, where the formulation (13) was first obtained.

In [37], Eckstein and Bertsekas proved that the DRS method (13) is a special case of the PP method (9) with $\lambda_k \equiv 1$ and $T := S_{\gamma, A, B}$, where $S_{\gamma, A, B}$ is a maximal monotone operator on $\mathcal{H}$ whose graph is

$$S_{\gamma, A, B} = \{(y + \gamma b, \gamma a + \gamma b) \in \mathcal{H} \times \mathcal{H} \mid b \in B(x),\, a \in A(y),\, \gamma a + y = x - \gamma b\}\,. \tag{14}$$

The splitting operator $S_{\gamma, A, B}$ is fundamental for proving that the DRS method is, actually, an application of the proximal point algorithm. Due to this fact, much of the theory of the DRS, and its special instance so-called *alternating direction method of multipliers* (ADMM) [45, 47], can be analyzed within the proximal point theory.

The Eckstein and Bertsekas' result also clarifies the proximal nature of the DRS method and gives some intuition to the fact that the DRS method has more general convergence properties than other proximal splitting algorithms [36, 37]. Moreover, the equivalence between (13) and the PP method of Rockafellar allowed [37] to derive new inexact and relaxed versions of the DRS and ADMM methods, after describing (13) according to the following procedure:

$$\text{compute } (x^k, b^k) \text{ such that } b^k \in B(x^k) \text{ and } \gamma b^k + x^k = z^{k-1}, \tag{15}$$

$$\text{compute } (y^k, a^k) \text{ such that } a^k \in A(y^k) \text{ and } \gamma a^k + y^k = x^k - \gamma b^k,$$

$$\text{set } z^k = y^k + \gamma b^k. \tag{16}$$

As an alternative to the summable error criterion (10), many modern inexact versions of the PP method use *relative error tolerances* for solving the associated proximal subproblems. The first methods of this type were proposed by Solodov and Svaiter in [76, 77] and subsequently studied in [66, 67, 68, 78, 79]. The key idea consists in decoupling (9) as an inclusion-equation system:

$$v \in T(z^+), \quad \lambda v + z^+ - z = 0, \tag{17}$$

where $(z, z^+, \lambda) := (z^{k-1}, z^k, \lambda_k)$, and relaxing (17) within relative error tolerance criteria. Among these new methods, the *hybrid proximal extragradient* (HPE) method [76] has been very effective as a framework for the design and analysis of many concrete algorithms (see, e.g., [14, 23, 38, 51, 52, 57, 63, 64, 65, 68, 76, 78, 79]).

In this thesis, we propose and study the iteration-complexity of an inexact Douglas-Rachford splitting method (Algorithm 3) and of a Douglas-Rachford-Tseng's forward-backward (F-B) four-operator splitting method (Algorithm 5) for solving (11) and (12), respectively. The former method is inspired and motivated (although based on a slightly different mechanism of iteration) by the recent work of J. Eckstein and W. Yao [40], while the latter one, which, in particular, will be shown to be a special instance of the former, is motivated by some variants of the Tseng's F-B splitting method [84] recently proposed in the current literature [6, 20, 66]. For more detailed information about the contributions of this thesis in light of reference [40], we refer the reader to the first remark after Algorithm 3. Moreover, we mention that, contrary to the majority of proximal splitting algorithms for solving problems with more than two blocks, Algorithm 5 is a purely primal splitting method for solving the *four-operator* MIP (12).

In the last years, the area of machine learning has attracted the attention of many research groups, with considerable impact in applied fields. In this context, the use of machine learning algorithms on large datasets in statistics has generated a significant impact in many areas, such as artificial intelligence, the internet, computational biology, medicine, marketing, advertising, network analysis, logistics, fraud detection, opinion mining, and economics, among others [17]. These problems can be posed in the general framework of convex optimization

$$\min_{x \in \mathbb{R}^n} \{f(x) + g(x)\} \tag{18}$$

where f, g are convex functions in $\mathbb{R}^n$.

The ADMM [45, 47] is a simple and yet powerful first-order algorithm for solving (18). It became popular over the last decade largely due to its wide range of applications in data science. It was first presented in the mid-1970s by Gabay, Mercier, Glowinski, and Marroco [45, 47], although similar ideas appeared still early in the 1950s (see, *e.g.*, [17]).

When applied to (18), one iteration of the method can be described as

$$x^{k+1} \in \operatorname{argmin}_{x\in\mathbb{R}^n} \left\{ f(x) + \langle p^k, x\rangle + \frac{c}{2}\|x - z^k\|^2 \right\}, \tag{19}$$

$$z^{k+1} \in \operatorname{argmin}_{z\in\mathbb{R}^n} \left\{ g(z) - \langle p^k, z\rangle + \frac{c}{2}\|x^{k+1} - z\|^2 \right\}, \tag{20}$$

$$p^{k+1} = p^k + c(x^{k+1} - z^{k+1}), \tag{21}$$

where $c > 0$ is a penalization parameter. The ADMM blends the advantages of dual decomposition and augmented Lagrangian for constrained optimization, and can be described as a coordinated decomposition procedure, in which the solutions of local subproblems (minors) are coordinated to find the solution of a global (larger) problem (*see, e.g.* [17]). However, unlike the augmented Lagrangian, in the ADMM the variables x and z are updated in an alternated (or sequential) fashion (see (19) and (20)), which accounts for the term alternating direction.

Moreover, in many applications, while one of the subproblems has a closed-form solution (*e.g.*, when $g(\cdot) = \|\cdot\|_1$), the other one can be expensive to solve numerically. Motivated by this, many inexact versions of the ADMM method (19)–(20) have been proposed (see, e.g., Algorithm 8 in Chapter 3).

In [46], Gabay presented, for the first time, the ADMM as an application of the DRS method for solving

$$0 \in \partial f(x) + \partial g(x) \tag{22}$$

which is, in particular, a special case of (11) with $A = \partial g$ and $B = \partial f$ (or *vice versa*). Problem (22) is, under standard qualification conditions, equivalent to (18). This perception was crucial to obtain some of the results presented by Eckstein in [36, 37] and, as a consequence, to obtain the algorithms proposed in [40] as well as the ones in this thesis.

Inertial algorithms for convex optimization and monotone inclusions [2] has been a subject of intense research in recent years. They appear in connection with continuous dynamics — see, *e.g.*, [2, 9, 10] — accelerated first- and second-order algorithms, and operator splitting methods — see *e.g.*, [7, 8, 16, 26, 27, 30, 58] — with good theoretical and practical performance improvements over prior methods.

This thesis develops a sequence of three inertial algorithms, each building on the previous one. The first algorithm (Algorithm 6) is a new variant of the PP algorithm [74] for solving the problem (8). It is, in particular, a new inertial variant of the relaxed hybrid proximal projection (HPP) method introduced in [79]; see also [77]. It lacks the full generality of [79], but introduces a new "inertial" step modification. From the first algorithm, we then develop a new inexact variant of the DRS method for solving monotone inclusion problems of the form (11). For such, we follow a similar derivation to [37], but use our Algorithm 6 in place of the HPE method of [76]. The resulting algorithm is an inertial-relaxed (inexact) relative-error DRS and is presented in Section 3.2 as Algorithm 7.

Finally, based on this latter method, we derive a new inexact variant of the ADMM algorithm for solving convex optimization problems of the form of (18). The resulting algorithm is presented in Section 3.3 as Algorithm 8. Using the well-known LASSO and logistic regression problems as examples, we perform some computational tests in Section 3.4 to show the applicability of the methods proposed in this work. Our numerical results showed better computational performance in terms of the number of iterations and execution time, when compared to earlier proposed inexact ADMM methods in [39, 40].

The main contributions of this thesis

The path for developing inexact DRS and ADMM methods was pioneered in [37] and is also taken in the more recent paper by Eckstein and Yao [40]. In both cases, one takes an approximate form of PP method [74] and uses it to obtain an approximate form of DRS, which can then be used to obtain new variants of the ADMM.

In this thesis, motivated by [40], we derive new inexact variants of the DRS method for maximal monotone operators and the ADMM for convex optimization. Firstly, we develop in Section 2.1 an inexact version of the DRS method (Algorithm 3) for solving (11) in which inexact computations are allowed *in both the inclusion and the equation* in (15). At each iteration, instead of a point in the graph of B, Algorithm 3 computes a point in the graph of the ε-enlargement B^{ε} of B (it has the property that $B^{\varepsilon}(z) \supset B(z)$). Moreover, contrary to the reference [40], we study the *iteration-complexity* of the proposed method for solving (11). We show that Algorithm 3 admits two types of iterations, one that can be embedded into the HPE method and another one that demands a separate analysis. We emphasize again that, although motivated by the latter reference, the Douglas-Rachford type method proposed in Chapter 2 is based on a slightly different mechanism of iteration, specially designed to allow its iteration-complexity analysis (see Theorems 2.1.5 and 2.1.6).

Secondly, in Section 2.2, we consider the four-operator MIP (12), for which we propose and study the iteration-complexity a Douglas-Rachford-Tseng's F-B splitting type method (Algorithm 5), which combines Algorithm 3 (as an outer iteration) and a Tseng's F-B splitting type method (Algorithm 4) (as an inner iteration) for solving the corresponding subproblems. The resulting algorithm, namely Algorithm 5, has a fully splitting nature and solves (12) without introducing additional variables.

Finally, in Chapter 3, we derive new inertial inexact variants of the PP, DRS and ADMM methods. One of the main differences between Chapter 3 and the development of "admm_primDR" in [40] is in the underlying variant of the PP algorithm. The "admm_primDR" analysis used the HPE method [76] due to Solodov and Svaiter, whereas here we use the new inexact HPP developed in Section 3.1. Moreover, contrary to [40], we introduce inertial and relaxation effects, which substantially improves the numerical performance of our algorithms. Our general approach resembles that of [40] that it using a primal derivation and the "coupling matrix" between f and g in the optimization formulation must be the identity, whereas [37], drawing on early work in [46], uses a dual derivation, and allows for more general coupling matrices. Our analysis is also much closer to [40] than that of [39], which uses a primal-dual "Lagrangian splitting" analysis patterned after [43]. Moreover, the inertial methods proposed in this thesis have the novel property of simultaneously combining inexact iterations, inertia, and relaxation, with the maximum inertial step α and maximum relaxation factor $\bar{\rho}$ that are subject to a mutual constraint; see (3.1.20) and (3.1.21). Thus, we may choose the inertia and relaxation parameters independently of the relative-error tolerances.

Most related works

In [18], the relaxed forward-Douglas-Rachford splitting (rFDRS) method was proposed and studied to solve *three-operator MIPs* consisting of (12) with $C = N_V$, V closed vector subspace, and $F_1 = 0$. Subsequently, among other results, the iteration-complexity of the latter method (specialized to variational problems) was analyzed in [31]. Problem (12) with $F_1 = 0$ was also considered in [32], where a three-operator splitting (TOS) method was proposed and its iteration-complexity studied. On the other hand, problem (12) with $C = N_V$ and $F_2 = 0$ was studied in [19], where the forward-partial inverse-forward splitting method was proposed and analyzed. In [20], a Tseng's F-B splitting type method was proposed and analyzed to solve the special instance of (12) in which $C = 0$.

The iteration-complexity of a relaxed Peaceman-Rachford splitting method for solving (11) was recently studied in [69]. The method of [69] was shown to be a special instance of a non-Euclidean HPE framework, for which the iteration-complexity was also analyzed in the latter reference (see also [49]). Two inexact versions of the ADMM were presented in [39]. The first method uses an absolutely summable error criterion and the second method uses a relative error criterion. Moreover, as we mentioned earlier, an inexact version of the DRS method for solving (11) was proposed and studied in [40]. Both in [39] and [40], various approximate forms of ADMM were tested computationally on different classes of problems. A different inexact of Douglas-Rachford method, in which both proximal subproblems are solved within a relative error tolerance, was recently proposed and studied in [82], but without computational testings. The sequences generated by this method converge weakly to the solution of the underlying inclusion problem, if any.

Presentation of chapters

This thesis is divided into three chapters, as described below.

Chapter 1: This chapter summarizes some definitions and preliminary results that contribute to structure, analyze, and clarify the nature of the methods proposed in this thesis. For more details, the material presented here can be readily found (often in more general form) in [21, 36, 67]. We start, in Section 1.1, by presenting the general notation and some basic concepts about convexity, maximal monotone operators, ε-enlargements, and related facts. Section 1.2 is dedicated to a concise review of the DRS, Rockafellar's PP and Solodov-Svaiter's HPE methods (Algorithm 1), as well as of a special version of the HPE method (Algorithm 2) of Marques Alves, Monteiro, and Svaiter for solving strongly monotone inclusions. Some results on the convergence analysis of the latter two methods are also presented in this section. These results are fundamental for studying the iteration-complexity of methods developed in Chapter 2.

Chapter 2: This chapter is devoted to developing and studying the iteration-complexity of an inexact DRS method and a Douglas-Rachford-Tseng's forward-backward (F-B) splitting method for solving two-operator and four-operator monotone inclusions, respectively. It is divided into two sections. In Section 2.1, we present the first method (Algorithm 3) which, although based on a slightly different mechanism of iteration, is motivated by the recent work of J. Eckstein and W. Yao [40], in which an inexact DRS method is derived from a special instance of the HPE method of Solodov and Svaiter. The second method (Algorithm 5) proposed in this chapter is introduced in Section 2.2, it combines the proposed inexact DRS method (Algorithm 3, used as an outer iteration) with a Tseng's F-B splitting type method (used as an inner iteration) for solving the corresponding subproblems. In this context, we prove iteration-complexity bounds for the two algorithms, in both sections, in the pointwise (non-ergodic) (Theorems 2.1.5 and 2.2.3) as well as in the ergodic sense (Theorems 2.1.6 and 2.2.4) by showing that they admit two different iterations: one that can be embedded into the HPE method, for which the iteration-complexity is known since the paper of Monteiro and Svaiter [67], and another one which demands a separate analysis.

The results of this chapter were published in [4].

Chapter 3: This chapter derives new inexact variants of the DRS method for maximal monotone operators and ADMM for convex optimization. The analysis of these two algorithms is based on our inertial-relaxed HPP method (Algorithm 6) presented in Section 3.1. It consists of a new inertial variant of the relaxed HPP method introduced in [79] (see also [77]), which includes both an inertial step and an overrelaxation. In this sense, it is important to emphasize that our method does not have the full generality of [79]. Nonetheless, it introduces a new "inertial" step modification. In Theorems 3.1.4 and 3.1.5 we prove the convergence analysis of Algorithm 6. Using our first algorithm (HPP method), Section 3.2 develops an inexact inertial-relaxed DRS method (Algorithm 7), for two-operator monotone inclusion problems of the form (11), for which convergence is established in Theorem 3.2.3. Section 3.3 then uses inertial-relaxed DR method to derive a partially inexact relative-error ADMM method (Algorithm 8) for solving convex optimization problems of the form (18). The main result of this section is Theorem 3.3.4. Finally, we perform some computational tests of this last algorithm in Section 3.4 using the well-known LASSO and logistic regression problems as examples, finding better practical performance than earlier proposed inexact ADMM methods [39, 40].

The results of this chapter were published in [3].

Chapter 1

Preliminaries and background materials

In this chapter, we present some definitions and preliminary results that contribute to structure, analyze, and clarify the nature of the methods developed in the other chapters. We start by recalling notations, definitions, and some preliminary materials of convexity, maximal monotone operators, and ε-enlargements, that shall be used throughout this thesis. In Section 1.2, we briefly review the following methods: DRS, Rockafellar's PP, HPE of Solodov and Svaiter (Algorithm 1), and a special version of the HPE method (Algorithm 2) proposed by Marques Alves, Monteiro, and Svaiter to solve strongly monotone inclusions. To clarify the analysis of the methods proposed in Chapter 2, in Subsections 1.2.4 and 1.2.5 we recall the pointwise and ergodic iteration-complexity bounds of Algorithms 1 and 2, respectively.

For further details regarding the results presented in this chapter, we refer the reader to [21, 36, 67].

1.1 General notation and ε-enlargements

Throughout this thesis $\mathcal{H}$ is a real Hilbert space with the inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \| := \sqrt{\langle \cdot, \cdot \rangle}$ and $\mathcal{H} \times \mathcal{H}$ denotes the Cartesian product endowed with the usual inner product and norm.

As we mentioned in the Introduction, the theory of (set-valued) maximal monotone operators plays a relevant role in different fields of applied mathematics and optimization including convex optimization, signal processing, PDEs, inverse problems, among others [11, 48]. Next, we review some basic definitions, facts, and notations from (set-valued) maximal monotone operators that will be used throughout this thesis.

A set-valued map $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is monotone if $\langle z - z', v - v' \rangle \geq 0, \quad \forall v \in T(z), \quad \forall v' \in T(z')$. On the other hand, T is a *maximal monotone operator* if T is monotone and $T = S$ whenever S is monotone on $\mathcal{H}$ and $T \subseteq S$. Here, we identify any monotone operator T with its graph, i.e., we set: $T = \{(z, v) \in \mathcal{H} \times \mathcal{H} \,|\, v \in T(z)\}$. The *sum* $T + S$ of two set-valued maps T, S is defined via the usual Minkowski sum and for $\lambda \geq 0$ the operator λT is defined by $(\lambda T)(z) = \lambda T(z) := \{\lambda v \,|\, v \in T(z)\}$. The *inverse* of $T : \mathcal{H} \rightrightarrows \mathcal{H}$, denoted by $T^{-1} : \mathcal{H} \rightrightarrows \mathcal{H}$, is defined by $v \in T^{-1}(z)$ if and only if $z \in T(v)$. Moreover, the set of zeros of $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is given by $\mathrm{zer}(T) := T^{-1}(0) = \{z \in \mathcal{H} \,|\, 0 \in T(z)\}$. The *resolvent* of a maximal monotone operator T is $J_T := (T + I)^{-1}$, where I denotes the identity map on $\mathcal{H}$, and, in particular, the following holds: $x = J_{\lambda T}(z)$ if and only if $\lambda^{-1}(z - x) \in T(x)$ if and only if $0 \in \lambda T(x) + x - z$. According to Minty in [62], if T is a maximal monotone operator, its resolvent is single-value operator defined everywhere on $\mathcal{H}$.

We denote by $\partial_\varepsilon f$ the usual ε-subdifferential of a proper closed convex function $f : \mathcal{H} \to (-\infty, +\infty]$, defined by $\partial_\varepsilon f(z) := \{v \in \mathcal{H} : f(z') \geq f(z) + \langle v, z' - z\rangle - \varepsilon \quad \forall z' \in \mathcal{H}\}$, and by $\partial f := \partial f_0$ the Fenchel-subdifferencial of f as well. The *normal cone* of a closed convex set X will be denoted by N_X and by P_X the orthogonal projection onto X.

For $T : \mathcal{H} \rightrightarrows \mathcal{H}$ maximal monotone and $\varepsilon \geq 0$, the ε-enlargement [21] of T is the operator $T^\varepsilon : \mathcal{H} \rightrightarrows \mathcal{H}$ defined by

$$T^\varepsilon(z) := \{v \in \mathcal{H} \mid \langle z - z', v - v'\rangle \geq -\varepsilon \;\; \forall (z', v') \in T\} \quad \forall z \in \mathcal{H}. \tag{1.1.1}$$

Note that $T(z) \subset T^\varepsilon(z)$ for all $z \in \mathcal{H}$.

The following proposition summarizes some useful properties of T^ε which will be useful in this thesis (see [67, Proposition 2.1]).

Proposition 1.1.1. *Let $T, S : \mathcal{H} \rightrightarrows \mathcal{H}$ be set-valued maps. Then,*

(a) *if $\varepsilon \leq \varepsilon'$, then $T^\varepsilon(x) \subseteq T^{\varepsilon'}(x)$ for every $x \in \mathcal{H}$;*

(b) *$T^\varepsilon(x) + S^{\varepsilon'}(x) \subseteq (T + S)^{\varepsilon+\varepsilon'}(x)$ for every $x \in \mathcal{H}$ and $\varepsilon, \varepsilon' \geq 0$;*

(c) *T is monotone if, and only if, $T \subseteq T^0$;*

(d) *T is maximal monotone if, and only if, $T = T^0$;*

Next we present the transportation formula for ε-enlargements.

Theorem 1.1.2. ([22, Theorem 2.3]) *Suppose $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is maximal monotone and let $z_\ell, v_\ell \in \mathcal{H}$, $\varepsilon_\ell, \alpha_\ell \in \mathbb{R}_+$, for $\ell = 1, \ldots, j$, be such that*

$$v_\ell \in T^{\varepsilon_\ell}(z_\ell), \quad \ell = 1, \ldots, j, \quad \sum_{\ell=1}^{j} \alpha_\ell = 1,$$

and define

$$\overline{z}_j := \sum_{\ell=1}^{j} \alpha_\ell \, z_\ell \,, \quad \overline{v}_j := \sum_{\ell=1}^{j} \alpha_\ell \, v_\ell \,, \quad \overline{\varepsilon}_j := \sum_{\ell=1}^{j} \alpha_\ell \left[\varepsilon_\ell + \langle z_\ell - \overline{z}_j, v_\ell - \overline{v}_j\rangle\right].$$

Then, the following hold:

(a) *$\overline{\varepsilon}_j \geq 0$ and $\overline{v}_j \in T^{\overline{\varepsilon}_j}(\overline{z}_j)$.*

(b) *If, in addition, $T = \partial f$ for some proper, convex and closed function f and $v_\ell \in \partial_{\varepsilon_\ell} f(z_\ell)$ for $\ell = 1, \ldots, j$, then $\overline{v}_j \in \partial_{\overline{\varepsilon}_j} f(\overline{z}_j)$.*

1.2 Proximal point and operator splitting methods

1.2.1 The Douglas-Rachford splitting (DRS) method

One of the most popular algorithms for finding approximate solutions of (11) is the *Douglas-Rachford splitting* (DRS) *method*. It consists of an iterative procedure in which at each iteration the resolvents $J_{\gamma A} = (\gamma A + I)^{-1}$ and $J_{\gamma B} = (\gamma B + I)^{-1}$ of A and B, respectively, are employed separately instead of the resolvent $J_{\gamma(A+B)}$ of the full operator $A + B$, which may be expensive to compute numerically. An iteration of the method can be described as in (13), i.e.,

$$z_k = J_{\gamma A}(2J_{\gamma B}(z_{k-1}) - z_{k-1}) + z_{k-1} - J_{\gamma B}(z_{k-1}) \qquad \forall k \geq 1, \tag{1.2.1}$$

where $\gamma > 0$ is a scaling parameter and z_{k-1} is the current iterate. Originally proposed in [35] for solving problems with linear operators, the DRS method was generalized in [56] for general nonlinear maximal monotone operators, where the formulation (1.2.1) was first obtained. It was proved in [56] that $\{z_k\}$ converges (weakly, in infinite dimensional Hilbert spaces) to some z^* such that $x^* := J_{\gamma B}(z^*)$ is a solution of (11). Recently, [80] solved the long standing open question of proving the weak convergence of the sequence $J_{\gamma B}(z_k)$ to a solution of (11).

1.2.2 The Rockafellar's proximal point (PP) method

The *proximal point* (PP) *method* is an iterative method for seeking approximate solutions of the MIP (8), i.e.,

$$0 \in T(z) \tag{1.2.2}$$

where T is a maximal monotone operator on $\mathcal{H}$ for which the solution set of (1.2.2) is nonempty. It was first proposed by Martinet [60] for solving monotone variational inequalities (with point-to-point operators) and further studied and developed by Rockafellar. In its exact formulation, an iteration of the PP method can be described as in (9), i.e.,

$$z_k = (\lambda_k T + I)^{-1} z_{k-1} \qquad \forall k \geq 1, \tag{1.2.3}$$

where $\lambda_k > 0$ is a stepsize parameter and z_{k-1} is the current iterate. It is well-known that the practical applicability of numerical schemes based on the exact computation of resolvents of monotone operators strongly depends on strategies that allow for inexact computations. This is the case of the PP method (1.2.3). In his pioneering work [74], Rockafellar proved that if, at each iteration $k \geq 1$, z_k is computed satisfying (10), i.e.,

$$\|z_k - (\lambda_k T + I)^{-1} z_{k-1}\| \leq e_k, \quad \sum_{k=1}^{\infty} e_k < \infty, \tag{1.2.4}$$

and $\{\lambda_k\}$ is bounded away from zero, then $\{z_k\}$ converges (weakly, in infinite dimensions) to a solution of (1.2.2). This result has found important applications in the design and analysis of many practical algorithms for solving challenging problems in optimization and related fields.

1.2.3 The DRS method is an instance of the PP method (Eckstein and Bertsekas)

In [37], the DRS method (1.2.1) was shown to be a special instance of the PP method (1.2.3) with $\lambda_k \equiv 1$. More precisely, it was observed in [37] (among other results) that the sequence $\{z_k\}$ in (1.2.1) satisfies

$$z_k = (S_{\gamma,\,A,\,B} + I)^{-1} z_{k-1} \qquad \forall k \geq 1, \tag{1.2.5}$$

where $S_{\gamma,\,A,\,B}$ is the maximal monotone operator on $\mathcal{H}$ whose graph is given by (14), i.e.,

$$S_{\gamma,\,A,\,B} = \{(y + \gamma b, \gamma a + \gamma b) \in \mathcal{H} \times \mathcal{H} \mid b \in B(x),\, a \in A(y),\, \gamma a + y = x - \gamma b\}\,. \tag{1.2.6}$$

It can be easily checked that z^* is a solution of (11) if and only if $z^* = J_{\gamma B}(x^*)$ for some x^* such that $0 \in S_{\gamma,\,A\,B}(x^*)$. The fact that (1.2.1) is equivalent to (1.2.5) clarifies the proximal nature of the DRS method and as a consequence, much of the theory of PP method can be transported to the DRS context and its special cases, included the ADMM. Moreover, allowed that [37] to obtain inexact and relaxed versions of the DRS and ADMM methods by alternatively describing (1.2.5) according to the procedure (15)-(16), given by:

$$\begin{aligned}
&\text{compute } \ (x_k, b_k) \ \text{ such that } \ b_k \in B(x_k) \ \text{ and } \ \gamma b_k + x_k = z_{k-1}; && (1.2.7)\\
&\text{compute } \ (y_k, a_k) \ \text{ such that } \ a_k \in A(y_k) \ \text{ and } \ \gamma a_k + y_k = x_k - \gamma b_k;\\
&\text{set } \ z_k = y_k + \gamma b_k. && (1.2.8)
\end{aligned}$$

1.2.4 The hybrid proximal extragradient (HPE) method of Solodov and Svaiter

Consider the *monotone inclusion problem* (MIP) (8), i.e.,

$$0 \in T(z) \tag{1.2.9}$$

where $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is a maximal monotone operator for which the solution set $T^{-1}(0)$ of (1.2.9) is nonempty.

Many applications recent interest in different fields of applied mathematics and optimization can be presented in the framework of *monotone inclusion problem* (8). As we mentioned earlier, the proximal point (PP) method of Rockafellar [74] is one of the most popular algorithms for finding approximate solutions of (1.2.9). Among the modern inexact versions of the PP method, the *hybrid proximal extragradient* (HPE) method of [76], which we present in what follows, has been shown to be very effective how a framework for the design and analysis of many concrete algorithms (see e.g. [14, 23, 38, 51, 52, 57, 64, 65, 68, 76, 78, 79, 63]).

Algorithm 1. Hybrid proximal extragradient (HPE) method for (1.2.9)

(0) Let $z_0 \in \mathcal{H}$ and $\sigma \in [0,1)$ be given and set $j \leftarrow 1$.

(1) Compute $(\widetilde{z}_j, v_j, \varepsilon_j) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ and $\lambda_j > 0$ such that

$$v_j \in T^{\varepsilon_j}(\widetilde{z}_j), \quad \|\lambda_j v_j + \widetilde{z}_j - z_{j-1}\|^2 + 2\lambda_j \varepsilon_j \leq \sigma^2 \|\widetilde{z}_j - z_{j-1}\|^2. \tag{1.2.10}$$

(2) Define

$$z_j = z_{j-1} - \lambda_j v_j, \tag{1.2.11}$$

set $j \leftarrow j+1$ and go to step 1.

Remarks.

1. If $\sigma = 0$ in (1.2.10), then it follows from Proposition 1.1.1(d) and (1.2.11) that $(z_+, v) := (z_j, v_j)$ and $\lambda := \lambda_j > 0$ satisfy (17), which means that the HPE method generalizes the exact Rockafellar's PP method.

2. Condition (1.2.10) clearly relaxes both the inclusion and the equation in (17) within a relative error criterion. Recall that $T^{\varepsilon}(\cdot)$ denotes the ε-enlargement of T and has the property that $T^{\varepsilon}(z) \supset T(z)$ (see Subsection 1.1 for details). Moreover, in (1.2.11) an extragradient step from the current iterate z_{j-1} gives the next iterate z_j.

3. We emphasize that specific strategies for computing the triple $(\widetilde{z}_j, v_j, \varepsilon_j)$ as well as the stepsize $\lambda_j > 0$ satisfying (1.2.10) will depend on the particular instance of the problem (1.2.9) under consideration. On the other hand, as mentioned before, the HPE method can also be used as a framework for the design and analysis of concrete algorithms for solving specific instances of (1.2.9) (see, e.g., [38, 64, 65, 66, 67, 68]). We also refer the reader to Sections 2.1 and 2.2, in this work, for applications of the HPE method in the context of decomposition/splitting algorithms for monotone inclusions.

Since the appearance of the paper [67], we have seen an increasing interest in studding the *iteration-complexity* of the HPE method and its special instances (e.g., Tseng's forward-backward splitting method, Korpelevich extragradient method and ADMM [66, 67, 68]). This depends on the following termination criterion [67]: given tolerances $\rho, \epsilon > 0$, find $z, v \in \mathcal{H}$ and $\varepsilon > 0$ such that

$$v \in T^{\varepsilon}(z), \quad \|v\| \leq \rho, \quad \varepsilon \leq \epsilon. \tag{1.2.12}$$

Note that, by Proposition 1.1.1(d), if $\rho = \epsilon = 0$ in (1.2.12) then $0 \in T(z)$, i.e., $z \in T^{-1}(0)$.

We now summarize the main results on *pointwise (non ergodic)* and *ergodic* iteration-complexity [67] of the HPE method that will be used in this thesis. The *aggregate stepsize sequence* $\{\Lambda_j\}$ and the *ergodic sequences* $\{\overline{\widetilde{z}}_j\}$, $\{\overline{v}_j\}$, $\{\overline{\varepsilon}_j\}$ associated to $\{\lambda_j\}$ and $\{\widetilde{z}_j\}$,

$\{v_j\}$, and $\{\varepsilon_j\}$ are, respectively,

$$\Lambda_j := \sum_{\ell=1}^{j} \lambda_\ell \,, \tag{1.2.13}$$

$$\bar{\tilde{z}}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \, \tilde{z}_\ell, \quad \bar{v}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \, v_\ell, \tag{1.2.14}$$

$$\bar{\varepsilon}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \left[\varepsilon_\ell + \langle \tilde{z}_\ell - \bar{\tilde{z}}_j, v_\ell - \bar{v}_j\rangle\right] = \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \left[\varepsilon_\ell + \langle \tilde{z}_\ell - \bar{\tilde{z}}_j, v_\ell\rangle\right]. \tag{1.2.15}$$

Theorem 1.2.1 ([67, Theorem 4.4(a) and 4.7])**.** *Let* $\{\tilde{z}_j\}$, $\{v_j\}$, *etc, be generated by the* HPE method (Algorithm 1) *and let* $\{\bar{\tilde{z}}_j\}$, $\{\bar{v}_j\}$, *etc, be given in* (1.2.13)–(1.2.15)*. Let also* d_0 *denote the distance from* z_0 *to* $T^{-1}(0) \neq \emptyset$ *and assume that* $\lambda_j \geq \underline{\lambda} > 0$ *for all* $j \geq 1$*. Then, the following hold:*

(a) *For any* $j \geq 1$, *there exists* $i \in \{1, \ldots, j\}$ *such that*

$$v_i \in T^{\varepsilon_i}(\tilde{z}_i), \quad \|v_i\| \leq \frac{d_0}{\underline{\lambda}\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}}, \quad \varepsilon_i \leq \frac{\sigma^2 d_0^2}{2(1-\sigma^2)\underline{\lambda}\, j}\,.$$

(b) *For any* $j \geq 1$,

$$\bar{v}_j \in T^{\bar{\varepsilon}_j}(\bar{\tilde{z}}_j), \quad \|\bar{v}_j\| \leq \frac{2d_0}{\underline{\lambda}\, j}, \quad \bar{\varepsilon}_j \leq \frac{2(1+\sigma/\sqrt{1-\sigma^2})d_0^2}{\underline{\lambda}\, j}\,.$$

Remark.

The (*pointwise* and *ergodic*) bounds given in (a) and (b) of Theorem 1.2.1 guarantee, respectively, that for given tolerances $\rho, \epsilon > 0$, the termination criterion (1.2.12) is satisfied in at most

$$\mathcal{O}\left(\max\left\{\frac{d_0^2}{\underline{\lambda}^2\rho^2}, \frac{d_0^2}{\underline{\lambda}\epsilon}\right\}\right) \quad \text{and} \quad \mathcal{O}\left(\max\left\{\frac{d_0}{\underline{\lambda}\rho}, \frac{d_0^2}{\underline{\lambda}\epsilon}\right\}\right)$$

iterations, respectively. We refer the reader to [67] for a complete study of the iteration-complexity of the HPE method and its special instances.

The proposition below will be useful in the next sections.

Proposition 1.2.2 ([67, Lemma 4.2 and Eq. (34)])**.** *Let* $\{z_j\}$ *be generated by the* HPE method (Algorithm 1)*. Then, for any* $z^* \in T^{-1}(0)$, *the sequence* $\{\|z^* - z_j\|\}$ *is nonincreasing. As a consequence, for every* $j \geq 1$, *we have*

$$\|z_j - z_0\| \leq 2d_0,$$

where d_0 *denotes the distance of* z_0 *to* $T^{-1}(0)$.

1.2.5 A HPE variant for strongly monotone sums

We now consider the MIP

$$0 \in S(z) + B(z) =: T(z) \tag{1.2.16}$$

where the following is assumed to hold:

(C1) S and B are maximal monotone operators on $\mathcal{H}$;

(C2) S is (additionally) μ–strongly monotone for some $\mu > 0$, i.e., there exists $\mu > 0$ such that

$$\langle z - z', v - v' \rangle \geq \mu \|z - z'\|^2 \qquad \forall v \in S(z), v' \in S(z');$$

(C3) the solution set $(S + B)^{-1}(0)$ of (1.2.16) is nonempty.

The main motivation to consider the above setting is Subsection 2.2.1, in which the monotone inclusion (2.2.5) is clearly a special instance of (1.2.16) with $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$, which is obviously $(1/\gamma)$-strongly maximal monotone on $\mathcal{H}$.

The algorithm below was proposed and studied, however using a different notation, in [6, Algorithm 1].

Algorithm 2. A specialized HPE method for solving strongly monotone inclusions

(0) Let $z_0 \in \mathcal{H}$ and $\sigma \in [0, 1)$ be given and set $j \leftarrow 1$.

(1) Compute $(\widetilde{z}_j, v_j, \varepsilon_j) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ and $\lambda_j > 0$ such that

$$v_j \in S(\widetilde{z}_j) + B^{\varepsilon_j}(\widetilde{z}_j), \quad \|\lambda_j v_j + \widetilde{z}_j - z_{j-1}\|^2 + 2\lambda_j \varepsilon_j \leq \sigma^2 \|\widetilde{z}_j - z_{j-1}\|^2. \tag{1.2.17}$$

(2) Define

$$z_j = z_{j-1} - \lambda_j v_j, \tag{1.2.18}$$

set $j \leftarrow j + 1$ and go to step 1.

Next proposition will be useful in Subsection 2.2.1.

Proposition 1.2.3 ([6, Proposition 2.2]). *Let $\{\widetilde{z}_j\}$, $\{v_j\}$ and $\{\varepsilon_j\}$ be generated by* Algorithm 2, *let $z^* := (S + B)^{-1}(0)$ and $d_0 := \|z_0 - z^*\|$. Assume that $\lambda_j \geq \underline{\lambda} > 0$ for all $j \geq 1$ and define*

$$\alpha := \left(\frac{1}{2\underline{\lambda}\mu} + \frac{1}{1 - \sigma^2} \right)^{-1} \in (0, 1).$$

Then, for all $j \geq 1$,

$$
\begin{aligned}
& v_j \in S(\widetilde{z}_j) + B^{\varepsilon_j}(\widetilde{z}_j), \\
& \|v_j\| \leq \sqrt{\frac{1+\sigma}{1-\sigma}} \left(\frac{(1-\alpha)^{(j-1)/2}}{\underline{\lambda}} \right) d_0, \\
& \varepsilon_j \leq \frac{\sigma^2}{2(1-\sigma^2)} \left(\frac{(1-\alpha)^{j-1}}{\underline{\lambda}} \right) d_0^2.
\end{aligned}
$$

Chapter 2

Iteration-complexity of an inexact Douglas-Rachford method and of a Douglas-Rachford-Tseng's F-B four-operator splitting method for solving monotone inclusions

In this chapter, we propose and study the iteration-complexity of an inexact Douglas-Rachford splitting method and a Douglas-Rachford-Tseng's forward-backward splitting method for solving two-operator and four-operator monotone inclusions, respectively. We prove iteration-complexity bounds for both algorithms in the pointwise and the ergodic sense by showing that they admit two different iterations: one that can be embedded into the HPE method, for which the iteration-complexity is known since the work of Monteiro and Svaiter, and another one which demands a separate analysis.

This chapter is organized as follows. In Section 2.1, we present the first method (Algorithm 3) in which inexact computations are allowed in both the inclusion and the equation in (1.2.7). The main results of this section are Theorem 2.1.5 and Theorem 2.1.6. In Section 2.2, we derive the second method (Algorithm 5 that combines Algorithm 3 (used as an outer iteration) with a Tseng's F-B splitting type method (Algorithm 4), used as an inner iteration) for solving the corresponding subproblems. The main results of this section are Theorem 2.2.3 and Theorem 2.2.4.

The results of this chapter were published in [4].

2.1 An inexact Douglas-Rachford splitting (DRS) method and its iteration-complexity

Consider problem (11), i.e., the problem of finding $z \in \mathcal{H}$ such that

$$0 \in A(z) + B(z) \tag{2.1.1}$$

where the following hold:

(D1) A and B are maximal monotone operators on $\mathcal{H}$;

(D2) the solution set $(A+B)^{-1}(0)$ of (2.1.1) is nonempty.

In this section, we propose and analyze the iteration-complexity of an inexact version of the *Douglas-Rachford splitting* (DRS) *method* [56] for finding approximate solutions of (2.1.1) according to the following termination criterion: given tolerances $\rho, \epsilon > 0$, find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that

$$\begin{aligned} &a \in A^{\varepsilon_a}(y), \;\; b \in B^{\varepsilon_b}(x), \\ &\gamma\|a+b\| = \|x-y\| \leq \rho, \\ &\varepsilon_a + \varepsilon_b \leq \epsilon, \end{aligned} \tag{2.1.2}$$

where $\gamma > 0$ is a scaling parameter. Note that if $\rho = \epsilon = 0$ in (2.1.2), then $z^* := x = y$ is a solution of (2.1.1).

As we mentioned earlier, the algorithm below is motivated by (1.2.7)–(1.2.8), as well as by the recent work of Eckstein and Yao [40].

Algorithm 3. An inexact Douglas-Rachford splitting method for (2.1.1)

(0) Let $z_0 \in \mathcal{H}$, $\gamma > 0$, $\tau_0 > 0$ and $0 < \sigma, \theta < 1$ be given and set $k \leftarrow 1$.

(1) Compute $(x_k, b_k, \varepsilon_{b,k}) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ such that

$$b_k \in B^{\varepsilon_{b,k}}(x_k), \quad \|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} \leq \tau_{k-1}. \tag{2.1.3}$$

(2) Compute $(y_k, a_k) \in \mathcal{H} \times \mathcal{H}$ such that

$$a_k \in A(y_k), \quad \gamma a_k + y_k = x_k - \gamma b_k. \tag{2.1.4}$$

(3) (3.a) If

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} \leq \sigma^2\|\gamma b_k + y_k - z_{k-1}\|^2, \tag{2.1.5}$$

then

$$z_k = z_{k-1} - \gamma(a_k + b_k), \quad \tau_k = \tau_{k-1} \qquad \text{[extragradient step]}. \tag{2.1.6}$$

(3.b) Else

$$z_k = z_{k-1}, \quad \tau_k = \theta\, \tau_{k-1} \qquad \text{[null step]}. \tag{2.1.7}$$

(4) Set $k \leftarrow k+1$ and go to step 1.

Remarks.

1. We emphasize that although it has been motivated by [40, Algorithm 3], Algorithm 3 is based on a slightly different mechanism of iteration. Moreover, it also allows for the computation of (x_k, b_k) in (2.1.3) in the $\varepsilon_{b,k}$– enlargement of B (it has the

property that $B^{\varepsilon_{b,k}}(x) \supset B(x)$ for all $x \in \mathcal{H}$); this will be crucial for the design and iteration-complexity analysis of the four-operator splitting method of Section 2.2. We also mention that, contrary to this work, no iteration-complexity analysis is performed in [40].

2. Computation of $(x_k, b_k, \varepsilon_{b,k})$ satisfying (2.1.3) will depend on the particular instance of the problem (2.1.1) under consideration. In Section 2.2, we will use Algorithm 3 for solving a four-operator splitting monotone inclusion. In this setting, at every iteration $k \geq 1$ of Algorithm 3, called an outer iteration, a Tseng's forward-backward (F-B) splitting type method will be used, as an inner iteration, to solve the (prox) subproblem (2.1.3).

3. Whenever the resolvent $J_{\gamma B} = (\gamma B + I)^{-1}$ is computable, then it follows that $(x_k, b_k) := (J_{\gamma B}(z_{k-1}), (z_{k-1} - x_k)/\gamma)$ and $\varepsilon_{b,k} := 0$ clearly solve (2.1.3). In this case, the left hand side of the inequality in (2.1.3) is zero and, as a consequence, the inequality (2.1.5) is always satisfied. In particular, (1.2.7)–(1.2.8) hold, i.e., in this case Algorithm 3 reduces to the (exact) DRS method.

4. In this chapter, we assume that the resolvent $J_{\gamma A} = (\gamma A + I)^{-1}$ is computable, which implies that $(y_k, a_k) := (J_{\gamma A}(x_k - \gamma b_k), (x_k - \gamma b_k - y_k)/\gamma)$ is the demanded pair in (2.1.4). An interesting topic for future investigation would be to relax (2.1.4) to allow inexact computations of (y_k, a_k) similarly to (2.1.3).

5. Algorithm 3 potentially performs extragradient steps and null steps, depending on the condition (2.1.5). It will be shown in Proposition 2.1.2 that iterations corresponding to extragradient steps reduce to a special instance of the HPE method, in which case pointwise and ergodic iteration-complexity results are available in the current literature (see Proposition 2.1.3). On the other hand, iterations corresponding to the null steps will demand a separate analysis (see Proposition 2.1.4).

As we mentioned in the latter remark, each iteration of Algorithm 3 is either an extragradient step or a null step (see (2.1.6) and (2.1.7)). This will be formally specified by considering the sets:

$$\begin{aligned} &\mathcal{A} := \text{indexes } k \geq 1 \text{ for which an extragradient step is executed at the iteration } k. \\ &\mathcal{B} := \text{indexes } k \geq 1 \text{ for which a null step is executed at the iteration } k. \end{aligned} \tag{2.1.8}$$

That said, we let

$$\mathcal{A} = \{k_j\}_{j \in J}, \quad J := \{j \geq 1 \mid j \leq \#\mathcal{A}\} \tag{2.1.9}$$

where $k_0 := 0$ and $k_0 < k_j < k_{j+1}$ for all $j \in J$, and let $\beta_0 := 0$ and

$$\beta_k := \text{the number of indexes for which a null step is executed until the iteration } k. \tag{2.1.10}$$

Note that direct use of the above definition and (2.1.7) yield

$$\tau_k = \theta^{\beta_k} \tau_0 \quad \forall k \geq 0. \tag{2.1.11}$$

In order to study the *ergodic iteration-complexity* of Algorithm 3 we also define the *ergodic sequences* associated to the sequences $\{x_{k_j}\}_{j\in J}$, $\{y_{k_j}\}_{j\in J}$, $\{a_{k_j}\}_{j\in J}$, $\{b_{k_j}\}_{j\in J}$, and $\{\varepsilon_{b,\,k_j}\}_{j\in J}$, for all $j \in J$, as follows:

$$\overline{x}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} x_{k_\ell}, \quad \overline{y}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} y_{k_\ell}\,, \tag{2.1.12}$$

$$\overline{a}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} a_{k_\ell}\,, \quad \overline{b}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} b_{k_\ell}\,, \tag{2.1.13}$$

$$\overline{\varepsilon}_{a,\,k_j} := \frac{1}{j}\sum_{\ell=1}^{j} \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell} - \overline{a}_{k_j}\rangle = \frac{1}{j}\sum_{\ell=1}^{j} \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell}\rangle, \tag{2.1.14}$$

$$\overline{\varepsilon}_{b,\,k_j} := \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell} - \overline{b}_{k_j}\rangle\right] = \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell}\rangle\right]. \tag{2.1.15}$$

Moreover, the results on iteration-complexity of Algorithm 3 (pointwise and ergodic) obtained in this chapter will depend on the following quantity:

$$d_{0,\gamma} := \operatorname{dist}(z_0, \operatorname{zer}(S_{\gamma,A,B})) = \min\{\|z_0 - z\| \mid z \in \operatorname{zer}(S_{\gamma,A,B})\} \tag{2.1.16}$$

which measures the quality of the initial guess z_0 in Algorithm 3 with respect to $\operatorname{zer}(S_{\gamma,A,B})$, where the operator $S_{\gamma,A,B}$ is such that $J_{\gamma B}(\operatorname{zer}(S_{\gamma,\,A,\,B})) = (A+B)^{-1}(0)$ (see (1.2.6)).

In the next proposition, we show that the procedure resulting by selecting the extragradient steps in Algorithm 3 can be embedded into HPE method.

First, we need the following lemma.

Lemma 2.1.1. *Let* $\{z_k\}$ *be generated by* Algorithm 3 *and let the set* J *be defined in* (2.1.9)*. Then,*

$$z_{k_{j-1}} = z_{k_j - 1} \qquad \forall j \in J. \tag{2.1.17}$$

Proof. Using (2.1.8) and (2.1.9) we have $\{k \geq 1 \,|\, k_{j-1} < k < k_j\} \subset \mathcal{B}$, for all $j \in J$. Consequently, using the definition of $\mathcal{B}$ in (2.1.8) and (2.1.7) we conclude that $z_k = z_{k_{j-1}}$ whenever $k_{j-1} \leq k < k_j$. As a consequence, we obtain that (2.1.17) follows from the fact that $k_{j-1} \leq k_j - 1 < k_j$. □

Proposition 2.1.2. *Let* $\{z_k\}$, $\{(x_k, b_k)\}$, $\{\varepsilon_{b,k}\}$ *and* $\{(y_k, a_k)\}$ *be generated by* Algorithm 3 *and let the operator* $S_{\gamma,\,A,\,B}$ *be defined in* (1.2.6)*. Define, for all* $j \in J$,

$$\widetilde{z}_{k_j} := y_{k_j} + \gamma b_{k_j}, \quad v_{k_j} := \gamma(a_{k_j} + b_{k_j}), \quad \varepsilon_{k_j} := \gamma\varepsilon_{b,k_j}. \tag{2.1.18}$$

Then, for all $j \in J$,

$$\begin{aligned} &v_{k_j} \in (S_{\gamma,\,A,\,B})^{\varepsilon_{k_j}}(\widetilde{z}_{k_j}), \qquad \|v_{k_j} + \widetilde{z}_{k_j} - z_{k_{j-1}}\|^2 + 2\varepsilon_{k_j} \leq \sigma^2\|\widetilde{z}_{k_j} - z_{k_{j-1}}\|^2, \\ &z_{k_j} = z_{k_{j-1}} - v_{k_j}. \end{aligned} \tag{2.1.19}$$

As a consequence, the sequences $\{\widetilde{z}_{k_j}\}_{j\in J}$, $\{v_{k_j}\}_{j\in J}$, $\{\varepsilon_{k_j}\}_{j\in J}$ *and* $\{z_{k_j}\}_{j\in J}$ *are generated by* Algorithm 1 *with* $\lambda_j \equiv 1$ *for solving* (1.2.9) *with* $T := S_{\gamma,\,A,\,B}$.

Proof. For any $(z', v') := (y + \gamma b, \gamma a + \gamma b) \in S_{\gamma, A, B}$ we have, in particular, $b \in B(x)$ and $a \in A(y)$ (see (1.2.6)). Using these inclusions, the inclusions in (2.1.3) and (2.1.4), the monotonicity of the operator A and (1.1.1) with $T = B$ we obtain

$$\begin{aligned} &\langle x_{k_j} - x, b_{k_j} - b\rangle \geq -\varepsilon_{b,k_j}, \\ &\langle y_{k_j} - y, a_{k_j} - a\rangle \geq 0. \end{aligned} \tag{2.1.20}$$

Moreover, using the identity in (2.1.4) and the corresponding one in (1.2.6) we find

$$(y_{k_j} - y) + \gamma(b_{k_j} - b) = (x_{k_j} - x) - \gamma(a_{k_j} - a). \tag{2.1.21}$$

Using (2.1.18), (2.1.20) and (2.1.21) we have

$$\begin{aligned} \langle \widetilde{z}_{k_j} - z', v_{k_j} - v'\rangle &= \langle (y_{k_j} + \gamma b_{k_j}) - (y + \gamma b), (\gamma a_{k_j} + \gamma b_{k_j}) - (\gamma a + \gamma b)\rangle \\ &= \langle y_{k_j} - y + \gamma(b_{k_j} - b), \gamma(a_{k_j} - a) + \gamma(b_{k_j} - b)\rangle \\ &= \gamma\langle y_{k_j} - y + \gamma(b_{k_j} - b), a_{k_j} - a\rangle + \gamma\langle y_{k_j} - y + \gamma(b_{k_j} - b), b_{k_j} - b\rangle \\ &= \gamma\langle y_{k_j} - y + \gamma(b_{k_j} - b), a_{k_j} - a\rangle + \gamma\langle x_{k_j} - x - \gamma(a_{k_j} - a), b_{k_j} - b\rangle \\ &= \gamma\langle y_{k_j} - y, a_{k_j} - a\rangle + \gamma\langle x_{k_j} - x, b_{k_j} - b\rangle \\ &\geq \gamma\langle x_{k_j} - x, b_{k_j} - b\rangle \\ &\geq -\varepsilon_{k_j}, \end{aligned}$$

which combined with definition (1.1.1) gives the inclusion in (2.1.19).

From (2.1.18), (2.1.17), the identity in (2.1.4) and (2.1.5) we also obtain

$$\begin{aligned} \|v_{k_j} + \widetilde{z}_{k_j} - z_{k_{j-1}}\|^2 &= \|\gamma(a_{k_j} + b_{k_j}) + (y_{k_j} + \gamma b_{k_j}) - z_{k_j - 1}\|^2 \\ &= \|(x_{k_j} - y_{k_j}) + (y_{k_j} + \gamma b_{k_j}) - z_{k_j - 1}\|^2 \\ &= \|\gamma b_{k_j} + x_{k_j} - z_{k_j - 1}\|^2 \\ &\leq \sigma^2\|\gamma b_{k_j} + y_{k_j} - z_{k_j - 1}\|^2 - 2\gamma\varepsilon_{b,k_j} \\ &= \sigma^2\|\widetilde{z}_{k_j} - z_{k_{j-1}}\|^2 - 2\varepsilon_{k_j}, \end{aligned}$$

which gives the inequality in (2.1.19). To finish the proof of (2.1.19), note that the desired identity in (2.1.19) follows from the first one in (2.1.6), the second one in (2.1.18) and (2.1.17). The last statement of the proposition follows from (2.1.18), (2.1.19) and Algorithm 1's definition. □

Proposition 2.1.3. (rate of convergence for extragradient steps) *Let* $\{(x_k, b_k)\}$, $\{(y_k, a_k)\}$ *and* $\{\varepsilon_{b,k}\}$ *be generated by* Algorithm 3 *and consider the ergodic sequences defined in* (2.1.12)–(2.1.15). *Let* $d_{0,\gamma}$ *and the set* J *be defined in* (2.1.16) *and* (2.1.9), *respectively. Then,*

(a) *For any* $j \in J$, *there exists* $i \in \{1, \dots, j\}$ *such that*

$$a_{k_i} \in A(y_{k_i}), \quad b_{k_i} \in B^{\varepsilon_{b,k_i}}(x_{k_i}), \tag{2.1.22}$$

$$\gamma\|a_{k_i} + b_{k_i}\| = \|x_{k_i} - y_{k_i}\| \leq \frac{d_{0,\gamma}}{\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}}, \tag{2.1.23}$$

$$\varepsilon_{b,k_i} \leq \frac{\sigma^2 d_{0,\gamma}^2}{2\gamma(1-\sigma^2)j}\,. \tag{2.1.24}$$

(b) *For any $j \in J$,*

$$\overline{a}_{k_j} \in A^{\overline{\varepsilon}_{a,k_j}}(\overline{y}_{k_j}), \quad \overline{b}_{k_j} \in B^{\overline{\varepsilon}_{b,\,k_j}}(\overline{x}_{k_j}), \tag{2.1.25}$$

$$\gamma\|\overline{a}_{k_j} + \overline{b}_{k_j}\| = \|\overline{x}_{k_j} - \overline{y}_{k_j}\| \leq \frac{2d_{0,\gamma}}{j}, \tag{2.1.26}$$

$$\overline{\varepsilon}_{a,\,k_j} + \overline{\varepsilon}_{b,\,k_j} \leq \frac{2(1+\sigma/\sqrt{1-\sigma^2})d_{0,\gamma}^2}{\gamma j}\,. \tag{2.1.27}$$

Proof. Note first that (2.1.22) follow from the inclusions in (2.1.3) and (2.1.4). Using the last statement in Proposition 2.1.2, Theorem 1.2.1 (with $\underline{\lambda} = 1$) and (2.1.16), we obtain that there exists $i \in \{1, \dots, j\}$ such that

$$\|v_{k_i}\| \leq \frac{d_{0,\gamma}}{\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}},$$
$$\varepsilon_{k_i} \leq \frac{\sigma^2 d_{0,\gamma}^2}{2(1-\sigma^2)j}\,,$$

which, in turn, combined with the identity in (2.1.4) and the definitions of v_{k_i} and ε_{k_i} in (2.1.18) gives the desired inequalities in (2.1.23) and (2.1.24) (concluding the proof of (a)) and

$$\|\overline{v}_j\| \leq \frac{2d_{0,\gamma}}{j},$$
$$\overline{\varepsilon}_j \leq \frac{2(1+\sigma/\sqrt{1-\sigma^2})d_{0,\gamma}^2}{j}\,, \tag{2.1.28}$$

where $\overline{v}_j$ and $\overline{\varepsilon}_j$ are defined in (1.2.14) and (1.2.15), respectively, with $\Lambda_j = j$ and

$$\lambda_\ell := 1, \quad v_\ell := v_{k_\ell}, \quad \varepsilon_\ell := \varepsilon_{k_\ell}, \quad \tilde{z}_\ell := \tilde{z}_{k_\ell} \qquad \forall \ell = 1, \dots, j. \tag{2.1.29}$$

Since the inclusions in (2.1.25) are a direct consequence of the ones in (2.1.3) and (2.1.4), Proposition 1.1.1(d), (2.1.12)–(2.1.15) and Theorem 1.1.2, it follows from (2.1.26), (2.1.27) and (2.1.28) that to finish the proof of (b), it suffices to prove that

$$\begin{aligned}
\overline{v_j} &= \gamma(\overline{a}_{k_j} + \overline{b}_{k_j}),\\
\gamma(\overline{a}_{k_j} + \overline{b}_{k_j}) &= \overline{x}_{k_j} - \overline{y}_{k_j},\\
\overline{\varepsilon}_j &= \gamma(\overline{\varepsilon}_{a,\,k_j} + \overline{\varepsilon}_{b,\,k_j}).
\end{aligned} \tag{2.1.30}$$

The first identity in (2.1.30) follows from (2.1.29), the second identities in (1.2.14) and (2.1.18), and (2.1.13). On the other hand, from (2.1.4) we have $\gamma(a_{k_\ell} + b_{k_\ell}) = x_{k_\ell} - y_{k_\ell}$, for all $\ell = 1, \dots, j$, which combined with (2.1.12) and (2.1.13) gives the second identity in (2.1.30). Using the latter identity and the second one in (2.1.30) we obtain

$$(y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}) = (x_{k_\ell} - \overline{x}_{k_j}) - \gamma(a_{k_\ell} - \overline{a}_{k_j}) \qquad \forall \ell = 1, \dots, j. \tag{2.1.31}$$

Moreover, it follows from (1.2.14), (2.1.29), the first identity in (2.1.18), (2.1.12) and (2.1.13) that

$$\bar{\tilde{z}}_j = \bar{\tilde{z}}_{k_j} = \frac{1}{j}\sum_{\ell=1}^{j} (y_{k_\ell} + \gamma b_{k_\ell}) = \overline{y}_{k_j} + \gamma \overline{b}_{k_j}. \tag{2.1.32}$$

Using (2.1.32), (2.1.29), (2.1.18) and (2.1.31) we obtain, for all $\ell = 1, \dots, j$,

$$\begin{aligned}
\langle \tilde{z}_\ell - \bar{\tilde{z}}_j, v_\ell\rangle &= \langle (y_{k_\ell} + \gamma b_{k_\ell}) - (\overline{y}_{k_j} + \gamma \overline{b}_{k_j}), \gamma(a_{k_\ell} + b_{k_\ell})\rangle \\
&= \gamma\langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), a_{k_\ell}\rangle + \gamma\langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), b_{k_\ell}\rangle \\
&= \gamma\langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), a_{k_\ell}\rangle + \gamma\langle (x_{k_\ell} - \overline{x}_{k_j}) - \gamma(a_{k_\ell} - \overline{a}_{k_j}), b_{k_\ell}\rangle \\
&= \gamma\langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell}\rangle + \gamma^2\langle b_{k_\ell} - \overline{b}_{k_j}, a_{k_\ell}\rangle + \gamma\langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell}\rangle - \gamma^2\langle a_{k_\ell} - \overline{a}_{k_j}, b_{k_\ell}\rangle,
\end{aligned}$$

which combined with (1.2.15), (2.1.29), (2.1.14) and (2.1.15) yields

$$\begin{aligned}
\overline{\varepsilon}_j = \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_\ell + \langle \tilde{z}_\ell - \bar{\tilde{z}}_j, v_\ell\rangle\right] &= \frac{1}{j}\sum_{\ell=1}^{j} \gamma\left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell}\rangle + \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell}\rangle\right] \\
&= \gamma(\overline{\varepsilon}_{a,\,k_j} + \overline{\varepsilon}_{b,\,k_j}),
\end{aligned}$$

which is exactly the last identity in (2.1.30). This finishes the proof. □

Proposition 2.1.4. (rate of convergence for null steps) *Let* $\{(x_k, b_k)\}$, $\{(y_k, a_k)\}$ *and* $\{\varepsilon_{b,k}\}$ *be generated by* Algorithm 3. *Let* $\{\beta_k\}$ *and the set* $\mathcal{B}$ *be defined in* (2.1.10) *and* (2.1.8), *respectively. Then, for* $k \in \mathcal{B}$,

$$\begin{aligned}
&a_k \in A(y_k), \quad b_k \in B^{\varepsilon_{b,\,k}}(x_k), \\
&\gamma\|a_k + b_k\| = \|x_k - y_k\| \le \frac{2\sqrt{\tau_0}}{\sigma}\,\theta^{\frac{\beta_{k-1}}{2}}, \\
&\gamma\varepsilon_{b,\,k} \le \frac{\tau_0}{2}\,\theta^{\beta_{k-1}}.
\end{aligned} \tag{2.1.33}$$

Proof. Note first that (2.1.33) follows from (2.1.3) and (2.1.4). Using (2.1.8), (2.1.3) and Step 3.b's definition (see Algorithm 3) we obtain

$$\tau_{k-1} \ge \|\underbrace{\gamma b_k + x_k - z_{k-1}}_{p_k}\|^2 + 2\gamma\varepsilon_{b,k} > \sigma^2\|\underbrace{\gamma b_k + y_k - z_{k-1}}_{q_k}\|^2,$$

which, in particular, gives

$$\gamma\varepsilon_{b,\,k} \le \frac{\tau_{k-1}}{2}, \tag{2.1.34}$$

and combined with the identity in (2.1.4) yields,

$$\begin{aligned}
\gamma\|a_k + b_k\| = \|x_k - y_k\| &= \|p_k - q_k\| \\
&\le \|p_k\| + \|q_k\| \\
&\le \left(1 + \frac{1}{\sigma}\right)\sqrt{\tau_{k-1}}.
\end{aligned} \tag{2.1.35}$$

To finish the proof, use (2.1.34), (2.1.35) and (2.1.11). □

Next we present the main results regarding the pointwise and ergodic iteration-complexity of Algorithm 3 for finding approximate solutions of (2.1.1) satisfying the termination criterion (2.1.2). While Theorem 2.1.5 is a consequence of Proposition 2.1.3(a) and Proposition 2.1.4, the ergodic iteration-complexity of Algorithm 3, namely Theorem 2.1.6, follows by combining the latter proposition and Proposition 2.1.3(b). Since the proof of Theorem 2.1.6 follows the same outline of Theorem 2.1.5's proof, it will be omitted.

Theorem 2.1.5. (pointwise iteration-complexity of Algorithm 3) *Assume that* $\max\{(1-\sigma)^{-1}, \sigma^{-1}\} = \mathcal{O}(1)$ *and let* $d_{0,\gamma}$ *be as in* (2.1.16). *Then, for given tolerances* $\rho, \epsilon > 0$, Algorithm 3 *finds* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *such that*

$$\begin{aligned} & a \in A(y),\; b \in B^{\varepsilon_b}(x), \\ & \gamma\|a+b\| = \|x-y\| \leq \rho, \\ & \varepsilon_b \leq \epsilon \end{aligned} \tag{2.1.36}$$

after performing at most

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\}\right) \tag{2.1.37}$$

extragradient steps *and*

$$\mathcal{O}\left(1 + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{2.1.38}$$

null steps. *As a consequence, under the above assumptions,* Algorithm 3 *terminates with* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *satisfying* (2.1.36) *in at most*

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{2.1.39}$$

iterations.

Proof. Let $\mathcal{A}$ be as in (2.1.8) and consider the cases:

$$\#\mathcal{A} \geq M_{\text{ext}} := \left\lceil \max\left\{\frac{2\,d_{0,\gamma}^2}{(1-\sigma)\rho^2}, \frac{\sigma^2 d_{0,\gamma}^2}{2\gamma(1-\sigma^2)\epsilon}\right\}\right\rceil \quad \text{and} \quad \#\mathcal{A} < M_{\text{ext}}. \tag{2.1.40}$$

In the first case, the desired bound (2.1.37) on the number of extragradient steps to find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_b \geq 0$ satisfying (2.1.36) follows from the definition of J in (2.1.9) and Proposition 2.1.3(a).

On the other hand, in the second case, i.e., $\#\mathcal{A} < M_{\text{ext}}$, the desired bound (2.1.38) is a direct consequence of Proposition 2.1.4. The last statement of the theorem follows from (2.1.37) and (2.1.38). □

Next is the main result on the ergodic iteration-complexity of Algorithm 3. As mentioned before, its proof follows the same outline of Theorem 2.1.5's proof, now applying Proposition 2.1.3(b) instead of the item (a) of the latter proposition.

Theorem 2.1.6. (ergodic iteration-complexity of Algorithm 3) *For given tolerances $\rho, \epsilon > 0$, under the same assumptions of* Theorem 2.1.5*,* Algorithm 3 *provides $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that*

$$\begin{aligned} & a \in A^{\varepsilon_a}(y), \;\; b \in B^{\varepsilon_b}(x), \\ & \gamma\|a + b\| = \|x - y\| \leq \rho, \\ & \varepsilon_a + \varepsilon_b \leq \epsilon. \end{aligned} \tag{2.1.41}$$

after performing at most

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\}\right)$$

extragradient steps *and*

$$\mathcal{O}\left(1 + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right)$$

null steps*. As a consequence, under the above assumptions,* Algorithm 3 *terminates with $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ satisfying* (2.1.41) *in at most*

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{2.1.42}$$

iterations.

Proof. The proof follows the same outline of Theorem 2.1.5's proof, now applying Proposition 2.1.3(b) instead of Proposition 2.1.3(a). □

Remarks.

1. Theorem 2.1.6 ensures that for given tolerances $\rho, \epsilon > 0$, up to an additive logarithmic factor, Algorithm 3 requires no more than

   $$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\}\right)$$

   iterations to find an approximate solution of the monotone inclusion problem (2.1.1) according to the termination criterion (2.1.2).

2. While the (ergodic) upper bound on the number of iterations provided in (2.1.42) is better than the corresponding one in (2.1.39) (in terms of the dependence on the tolerance $\rho > 0$) by a factor of $\mathcal{O}(1/\rho)$, the inclusion in (2.1.41) is potentially weaker than the corresponding one in (2.1.36), since one may have $\varepsilon_a > 0$ in (2.1.41), and the set $A^{\varepsilon_a}(y)$ is in general larger than $A(y)$.

3. Iteration-complexity results similar to the ones in Proposition 2.1.3 were recently obtained for a relaxed Peaceman-Rachford method in [69] . We emphasize that, in contrast to this work, the latter reference considers only the case where the resolvents $J_{\gamma A}$ and $J_{\gamma B}$ of A and B, respectively, are both computable.

The proposition below will be important in the next section.

Proposition 2.1.7. *Let* $\{z_k\}$ *be generated by* Algorithm 3 *and* $d_{0,\gamma}$ *be as in* (2.1.16). *Then,*

$$\|z_k - z_0\| \leq 2d_{0,\gamma} \quad \forall k \geq 1. \tag{2.1.43}$$

Proof. Note that (i) if $k = k_j \in \mathcal{A}$, for some $j \in J$, see (2.1.9), then (2.1.43) follows from the last statement in Proposition 2.1.2 and Proposition 1.2.2; (ii) if $k \in \mathcal{B}$, from the first identity in (2.1.7), see (2.1.8), we find that either $z_k = z_0$, in which case (2.1.43) holds trivially, or $z_k = z_{k_j}$ for some $j \in J$, in which case the results follows from (i). □

2.2 A Douglas-Rachford-Tseng's forward-backward four-operator splitting method

In this section, we consider problem (12), i.e., the problem of finding $z \in \mathcal{H}$ such that

$$0 \in A(z) + C(z) + F_1(z) + F_2(z) \tag{2.2.1}$$

where the following hold:

(E1) A and C are (set-valued) maximal monotone operators on $\mathcal{H}$.

(E2) $F_1 : D(F_1) \subset \mathcal{H} \to \mathcal{H}$ is monotone and L-Lipschitz continuous on a (nonempty) closed convex set Ω such that $D(C) \subset \Omega \subset D(F_1)$, i.e., F_1 is monotone on Ω and there exists $L \geq 0$ such that

$$\|F_1(z) - F_1(z')\| \leq L\|z - z'\| \qquad \forall z, z' \in \Omega. \tag{2.2.2}$$

(E3) $F_2 : \mathcal{H} \to \mathcal{H}$ is $\eta-$cocoercive, i.e., there exists $\eta > 0$ such that

$$\langle F_2(z) - F_2(z'), z - z' \rangle \geq \eta \|F_2(z) - F_2(z')\|^2 \qquad \forall z, z' \in \mathcal{H}.$$

(E4) $B^{-1}(0)$ is nonempty, where

$$B := C + F_1 + F_2. \tag{2.2.3}$$

(E5) The solution set of (2.2.1) is nonempty.

Aiming at solving the monotone inclusion (2.2.1), we present and study the iteration-complexity of a (four-operator) splitting method which combines Algorithm 3 (used as an outer iteration) and a Tseng's forward-backward (F-B) splitting type method (used as an inner iteration for solving, for each outer iteration, the prox subproblems in (2.1.3)). We prove results on pointwise and ergodic iteration-complexity of the proposed four-operator splitting algorithm by analyzing it in the framework of Algorithm 3 for solving (2.1.1) with B as in (2.2.3) and under assumptions (E1)–(E5). The (outer) iteration complexities will follow from results on pointwise and ergodic iteration complexities of Algorithm 3, obtained in Section 2.1, while the computation of an upper bound on the overall number of inner iterations required to achieve prescribed tolerances will require a separate analysis. Still regarding the results on iteration-complexity, we mention that we consider the following notion of approximate solution for (2.2.1): given tolerances $\rho, \epsilon > 0$, find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that

$$\begin{aligned} & a \in A^{\varepsilon_a}(y), \\ & \text{either } \; b \in C(x) + F_1(x) + F_2^{\varepsilon_b}(x) \; \text{ or } \; b \in (C + F_1 + F_2)^{\varepsilon_b}(x), \\ & \gamma\|a + b\| = \|x - y\| \leq \rho, \;\; \varepsilon_a + \varepsilon_b \leq \epsilon, \end{aligned} \tag{2.2.4}$$

where $\gamma > 0$. Note that (i) for $\rho = \epsilon = 0$, the above conditions imply that $z^* := x = y$ is a solution of the monotone inclusion (2.2.1); (ii) the second inclusion in (2.2.4), which will appear in the ergodic iteration-complexity, is potentially weaker than the first one (see Proposition 1.1.1(b)), which will appear in the corresponding pointwise iteration-complexity of the proposed method.

We also mention that problem (2.2.1) falls in the framework of the monotone inclusion (2.1.1) due to the facts that, in view of assumptions (E1), (E2) and (E3), the operator A is maximal monotone, and the operator $F_1 + F_2$ is monotone and $(L + 1/\eta)$–Lipschitz continuous on the closed convex set $\Omega \supset D(C)$, which combined with the assumption on the operator C in (E1) and with [66, Proposition A.1] implies that the operator B defined in (2.2.3) is maximal monotone as well. These facts combined with assumption (E5) give that conditions (D1) and (D2) of Section 2.1 hold for A and B as in (E1) and (2.2.3), respectively. In particular, it gives that Algorithm 3 may be applied to solve the four-operator monotone inclusion (2.2.1).

In this regard, we emphasize that any implementation of Algorithm 3 will heavily depend on specific strategies for solving each subproblem in (2.1.3), since (y_k, a_k) required in (2.1.4) can be computed by using the resolvent operator of A, available in closed form in many important cases. In the next subsection, we show how the specific structure (2.2.1) allows for an application of a Tseng's F-B splitting type method for solving each subproblem in (2.1.3).

2.2.1 Solving the subproblems in (2.1.3) for B as in (2.2.3)

In this subsection, we present and study a Tseng's F-B splitting type method [11, 20, 66, 84] for solving the corresponding proximal subproblem in (2.1.3) at each (outer) iteration of Algorithm 3, when used to solve (2.2.1). To begin with, first consider the (strongly) monotone inclusion

$$0 \in B(z) + \frac{1}{\gamma}(z - \mathring{z}) \tag{2.2.5}$$

where B is as in (2.2.3), $\gamma > 0$ and $\mathring{z} \in \mathcal{H}$, and note that the task of finding $(x_k, b_k, \varepsilon_{b,k})$ satisfying (2.1.3) is related to the task of solving (2.2.5) with $\mathring{z} := z_{k-1}$.

In the remaining part of this subsection, we present and study a Tseng's F-B splitting type method for solving (2.2.5). As we have mentioned before, the resulting algorithm will be used as an inner procedure for solving the subproblems (2.1.3) at each iteration of Algorithm 3, when applied to solve (2.2.1).

Algorithm 4. A Tseng's F-B splitting type method for (2.2.5)

Input: C, F_1, Ω, L, F_2 and η as in conditions (E1)–(E5), $\mathring{z} \in \mathcal{H}$, $\mathring{\tau} > 0$, $\sigma \in (0,1)$ and γ such that

$$0 < \gamma \leq \frac{4\eta\sigma^2}{1 + \sqrt{1 + 16L^2\eta^2\sigma^2}}. \tag{2.2.6}$$

(0) Set $z_0 \leftarrow \mathring{z}$ and $j \leftarrow 1$.

(1) Let $z'_{j-1} \leftarrow P_\Omega(z_{j-1})$ and compute

$$\begin{aligned} \widetilde{z}_j &= \left(\frac{\gamma}{2}C + I\right)^{-1}\left(\frac{\mathring{z} + z_{j-1} - \gamma(F_1 + F_2)(z'_{j-1})}{2}\right), \\ z_j &= \widetilde{z}_j - \gamma\left(F_1(\widetilde{z}_j) - F_1(z'_{j-1})\right). \end{aligned} \tag{2.2.7}$$

(2) If

$$\|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \leq \mathring{\tau}, \tag{2.2.8}$$

then **terminate**. Otherwise, set $j \leftarrow j + 1$ and go to step 1.

Output: $(z_{j-1}, z'_{j-1}, z_j, \widetilde{z}_j)$.

Remark.

Algorithm 4 combines ideas from the standard Tseng's F-B splitting algorithm [84] as well as from recent insights on the convergence and iteration-complexity of some variants the latter method [6, 20, 66]. In this regard, evaluating the cocoercive component F_2 just once per iteration (see [20, Theorem 1]) is potentially important in many applications, where the evaluation of cocoercive operators is in general computationally expensive (see [20] for a discussion). Nevertheless, we emphasize that the results obtained in this chapter regarding the analysis of Algorithm 4 do not follow from any of the just mentioned references.

Next corollary ensures that Algorithm 4 always terminates with the desired output.

Corollary 2.2.1. *Assume that* $(1 - \sigma^2)^{-1} = \mathcal{O}(1)$ *and let* $d_{\mathring{z},b}$ *denote the distance of* $\mathring{z}$ *to* $B^{-1}(0) \neq \emptyset$*. Then,* Algorithm 4 *terminates with the desired output after performing no more than*

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{\mathring{z},b}}{\sqrt{\mathring{\tau}}}\right)\right) \tag{2.2.9}$$

iterations.

Proof. See Subsection 2.2.3. □

2.2.2 A Douglas-Rachford-Tseng's F-B four-operator splitting method

In this subsection, we present and study the iteration-complexity of the main algorithm in this chapter, for solving (2.2.1), namely Algorithm 5, which combines Algorithm 3, used as an outer iteration, and Algorithm 4, used as an inner iteration, for solving the corresponding subproblem in (2.1.3). Algorithm 5 will be shown to be a special instance of Algorithm 3, for which pointwise and ergodic iteration-complexity results are available in Section 2.1. Corollary 2.2.1 will be specially important to compute a bound on the total number of inner iterations performed by Algorithm 5 to achieve prescribed tolerances.

Algorithm 5. A Douglas-Rachford-Tseng's F-B splitting type method for (2.2.1)

(0) Let $z_0 \in \mathcal{H}$, $\tau_0 > 0$ and $0 < \sigma, \theta < 1$ be given, let C, F_1, Ω, L, F_2 and η as in conditions (E1)–(E5) and γ satisfying condition (2.2.6), and set $k \leftarrow 1$.

(1) Call Algorithm 4 with inputs C, F_1, Ω, L, F_2 and η, $(\mathring{z}, \mathring{\tau}) := (z_{k-1}, \tau_{k-1})$, σ and γ to obtain as output $(z_{j-1}, z'_{j-1}, z_j, \widetilde{z}_j)$, and set

$$x_k = \widetilde{z}_j, \qquad b_k = \frac{z_{k-1} + z_{j-1} - (z_j + \widetilde{z}_j)}{\gamma}, \qquad \varepsilon_{b,k} = \frac{\|z'_{j-1} - \widetilde{z}_j\|^2}{4\eta}. \tag{2.2.10}$$

(2) Compute $(y_k, a_k) \in \mathcal{H} \times \mathcal{H}$ such that

$$a_k \in A(y_k), \quad \gamma a_k + y_k = x_k - \gamma b_k. \tag{2.2.11}$$

(3) (3.a) If

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma \varepsilon_{b,k} \le \sigma^2 \|\gamma b_k + y_k - z_{k-1}\|^2,$$

then

$$z_k = z_{k-1} - \gamma(a_k + b_k), \quad \tau_k = \tau_{k-1} \qquad \text{[extragradient step]}. \tag{2.2.12}$$

(3.b) Else

$$z_k = z_{k-1}, \quad \tau_k = \theta\, \tau_{k-1} \qquad \text{[null step]}. \tag{2.2.13}$$

(4) Set $k \leftarrow k + 1$ and go to step 1.

In what follows we present the pointwise and ergodic iteration complexities of Algorithm 5 for solving the four-operator monotone inclusion problem (2.2.1). The results will follow essentially from the corresponding ones for Algorithm 3 previously obtained in Section 2.1. On the other hand, bounds on the number of inner iterations executed before achieving prescribed tolerances will be proved by using Corollary 2.2.1.

We start by showing that Algorithm 5 is a special instance of Algorithm 3.

Proposition 2.2.2. *The triple* $(x_k, b_k, \varepsilon_{b,k})$ *in* (2.2.10) *satisfies condition* (2.1.3) *in* Step 1 *of* Algorithm 3*, i.e.,*

$$b_k \in C(x_k) + F_1(x_k) + F_2^{\varepsilon_{b,k}}(x_k) \subset B^{\varepsilon_{b,k}}(x_k), \quad \|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} \le \tau_{k-1}, \tag{2.2.14}$$

where B *is as in* (2.2.3)*. As a consequence,* Algorithm 5 *is a special instance of* Algorithm 3 *for solving* (2.1.1) *with* B *as in* (2.2.3)*.*

Proof. Using the first identity in (2.2.21), the definition of b_k in (2.2.10) as well as the fact that $\mathring{z} := z_{k-1}$ in Step 1 of Algorithm 5 we find

$$b_k = v_j - \frac{1}{\gamma}(\widetilde{z}_j - z_{k-1}) = v_j - \frac{1}{\gamma}(\widetilde{z}_j - \mathring{z}). \tag{2.2.15}$$

Combining the latter identity with the second inclusion in (2.2.22), the second identity in (2.2.21) and the definitions of x_k and $\varepsilon_{b,k}$ in (2.2.10) we obtain the first inclusion in (2.2.14). The second desired inclusion follows from (2.2.3) and Proposition 1.1.1(b). To finish the proof of (2.2.14), note that from the first identity in (2.2.15), the definitions of x_k and $\varepsilon_{b,k}$ in (2.2.10), the definition of v_j in (2.2.21) and (2.2.8) we have

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} = \|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \le \mathring{\tau} = \tau_{k-1},$$

which gives the inequality in (2.2.14). The last statement of the proposition follows from (2.2.14), (2.1.3)–(2.1.7) and (2.2.11)–(2.2.13). □

Theorem 2.2.3. (pointwise iteration-complexity of Algorithm 5) *Let the operator* B *and* $d_{0,\gamma}$ *be as in* (2.2.3) *and* (2.1.16)*, respectively, and assume that* $\max\{(1-\sigma)^{-1}, \sigma^{-1}\} = \mathcal{O}(1)$*. Let also* $d_{0,b}$ *be the distance of* z_0 *to* $B^{-1}(0) \neq \emptyset$*. Then, for given tolerances* $\rho, \epsilon > 0$*, the following hold:*

(a) Algorithm 5 *finds* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \ge 0$ *such that*

$$a \in A(y),\ b \in C(x) + F_1(x) + F_2^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \le \rho,\ \varepsilon_b \le \epsilon \tag{2.2.16}$$

after performing no more than

$$k_{\text{p; outer}} := \mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right)$$

outer iterations.

(b) *Before achieving the desired tolerance* $\rho, \epsilon > 0$*, each iteration of* Algorithm 5 *performs at most*

$$k_{\text{inner}} := \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{2.2.17}$$

inner iterations; and hence evaluations of the η*–cocoercive operator* F_2*.*

As a consequence, Algorithm 5 *finds* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *satisfying* (2.2.16) *after performing no more than* $k_{\mathrm{p;\,outer}} \times k_{\mathrm{inner}}$ *inner iterations.*

Proof. (a) The desired result is a direct consequence of the last statements in Proposition 2.2.2 and Theorem 2.1.5, and the inclusions in (2.2.14).

(b) Using Step 1's definition and Corollary 2.2.1 we conclude that, at each iteration $k \geq 1$ of Algorithm 5, the number of inner iterations is bounded by

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{z_{k-1},b}}{\sqrt{\tau_{k-1}}}\right)\right) \tag{2.2.18}$$

where $d_{z_{k-1},b}$ denotes the distance of z_{k-1} to $B^{-1}(0)$. Now, using the last statements in Propositions 2.2.2 and 2.1.2, Proposition 1.2.2 and a simple argument based on the triangle inequality we obtain

$$d_{z_{k-1},b} \leq 2d_{0,\gamma} + d_{0,b} \qquad \forall k \geq 1. \tag{2.2.19}$$

By combining (2.2.18) and (2.2.19) and using (2.1.11) we find that, at every iteration $k \geq 1$, the number of inner iterations is bounded by

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\theta^{\beta_{k-1}}\tau_0}}\right)\right) = \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \beta_{k-1}\right).$$

Using the latter bound, the last statement in Proposition 2.2.2, the bound on the number of null steps of Algorithm 3 given in Theorem 2.1.5, and (2.1.10) we conclude that, before achieving the prescribed tolerance $\rho, \epsilon > 0$, each iteration Algorithm 5 performs at most the number of iterations given in (2.2.17). This concludes the proof of (b).

To finish the proof, note that the last statement of the theorem follows directly from (a) and (b). □

Theorem 2.2.4. (ergodic iteration-complexity of Algorithm 5) *For given tolerances* $\rho, \epsilon > 0$*, under the same assumptions of* Theorem 2.2.3 *the following hold:*

(a) Algorithm 5 *provides* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_a, \varepsilon_b \geq 0$ *such that*

$$a \in A^{\varepsilon_a}(y),\ b \in (C + F_1 + F_2)^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \leq \rho,\ \varepsilon_a + \varepsilon_b \leq \epsilon \tag{2.2.20}$$

after performing no more than

$$k_{\mathrm{e;\,outer}} := \mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right)$$

outer iterations.

(b) *Before achieving the desired tolerance* $\rho, \epsilon > 0$*, each iteration of* Algorithm 5 *performs at most*

$$k_{\mathrm{inner}} := \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right)$$

inner iterations; and hence evaluations of the η*–cocoercive operator* F_2*.*

As a consequence, Algorithm 5 *provides* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *satisfying* (2.2.20) *after performing no more than* $k_{\mathrm{e;\,outer}} \times k_{\mathrm{inner}}$ *inner iterations.*

Proof. The proof follows the same outline of Theorem 2.2.3's proof. □

2.2.3 Proof of Corollary 2.2.1

We start this subsection by showing that Algorithm 4 is a special instance of Algorithm 2 for solving the strongly monotone inclusion (2.2.5).

Proposition 2.2.5. *Let* $\{z_j\}$, $\{z'_j\}$ *and* $\{\tilde{z}_j\}$ *be generated by* Algorithm 4 *and let the operator* B *be as in* (2.2.3). *Define,*

$$v_j := \frac{z_{j-1} - z_j}{\gamma}, \qquad \varepsilon_j := \frac{\|z'_{j-1} - \tilde{z}_j\|^2}{4\eta}, \quad \forall j \geq 1. \tag{2.2.21}$$

Then, for all $j \geq 1$,

$$v_j \in (1/\gamma)(\tilde{z}_j - \mathring{z}) + C(\tilde{z}_j) + F_1(\tilde{z}_j) + F_2^{\varepsilon_j}(\tilde{z}_j) \subset (1/\gamma)(\tilde{z}_j - \mathring{z}) + B^{\varepsilon_j}(\tilde{z}_j), \tag{2.2.22}$$

$$\|\gamma v_j + \tilde{z}_j - z_{j-1}\|^2 + 2\gamma\varepsilon_j \leq \sigma^2 \|\tilde{z}_j - z_{j-1}\|^2, \tag{2.2.23}$$

$$z_j = z_{j-1} - \gamma v_j. \tag{2.2.24}$$

As a consequence, Algorithm 4 *is a special instance of* Algorithm 2 *with* $\lambda_j \equiv \gamma$ *for solving* (1.2.16) *with* $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$.

Proof. Note that the first identity in (2.2.7) gives

$$\frac{z_{j-1} - \tilde{z}_j}{\gamma} - F_1(z'_{j-1}) \in (1/\gamma)(\tilde{z}_j - \mathring{z}) + C(\tilde{z}_j) + F_2(z'_{j-1}).$$

Adding $F_1(\tilde{z}_j)$ in both sides of the above identity and using the second and first identities in (2.2.7) and (2.2.21), respectively, we find

$$v_j = \frac{z_{j-1} - z_j}{\gamma} \in (1/\gamma)(\tilde{z}_j - \mathring{z}) + C(\tilde{z}_j) + F_1(\tilde{z}_j) + F_2(z'_{j-1}),$$

which, in turn, combined with Lemma A.2 and the definition of ε_j in (2.2.21) proves the first inclusion in (2.2.22). Note now that the second inclusion in (2.2.22) is a direct consequence of (2.2.3) and Proposition 1.1.1(b). Moreover, (2.2.24) is a direct consequence of the first identity in (2.2.21).

To prove (2.2.23), note that from (2.2.21), the second identity in (2.2.7), (2.2.6) and (2.2.2) we have

$$\begin{aligned} \|\gamma v_j + \tilde{z}_j - z_{j-1}\|^2 + 2\gamma\varepsilon_j &= \gamma^2 \|F_1(\tilde{z}_j) - F_1(z'_{j-1})\|^2 + \frac{\gamma \|z'_{j-1} - \tilde{z}_j\|^2}{2\eta} \\ &\leq \left(\gamma^2 L^2 + \frac{\gamma}{2\eta}\right) \|z'_{j-1} - \tilde{z}_j\|^2 \\ &\leq \sigma^2 \|z_{j-1} - \tilde{z}_j\|^2, \end{aligned}$$

which is exactly the desired inequality, where we also used the facts that $z'_{j-1} = P_\Omega(z_{j-1})$, $\tilde{z}_j \in D(C) \subset \Omega$ and that P_Ω is nonexpansive. The last statement of the proposition follows from (2.2.22)–(2.2.24), (2.2.5), (1.2.17) and (1.2.18). □

Proof of Corollary 2.2.1. Let, for all $j \geq 1$, $\{v_j\}$ and $\{\varepsilon_j\}$ be defined in (2.2.21). Using the last statement in Proposition 2.2.5 and Proposition 1.2.3 with $\mu := 1/\gamma$ and $\underline{\lambda} := \gamma$ we find

$$\|\gamma v_j\|^2 + 2\gamma\varepsilon_j \leq \frac{((1+\sigma)^2 + \sigma^2)(1-\alpha)^{j-1}\|\mathring{z} - z_\gamma^*\|^2}{1-\sigma^2}, \tag{2.2.25}$$

where $z_\gamma^* := (S+B)^{-1}(0)$ with $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$, i.e., $z_\gamma^* = (\gamma B + I)^{-1}(\mathring{z})$. Now, using (2.2.25), (2.2.21) and Lemma A.1 we obtain

$$\|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \leq \frac{((1+\sigma)^2 + \sigma^2)(1-\alpha)^{j-1} d_{\mathring{z},b}^2}{1-\sigma^2},$$

which in turn combined with (2.2.8), after some direct calculations, gives (2.2.9).

Chapter 3

Relative-error inertial-relaxed inexact versions of Douglas-Rachford and ADMM splitting algorithms

In this chapter, we analyze the asymptotic behavior of new variants of the DRS and ADMM splitting methods, both under relaxation and inertial effects and with inexact (relative-error) criterion for subproblems. The first is an inexact version of the proximal point algorithm that includes both an inertial step and overrelaxation. The second is an inexact variant of the Douglas-Rachford splitting method for maximal monotone operators, while the latter is an inexact variant of the alternating direction method of multipliers (ADMM) for convex optimization.

This chapter is organized as follows. In Section 3.1, we present our inertial-relaxed HPP method (Algorithm 6) and its convergence analysis. In Section 3.2, we use the HPP method (Algorithm 6) to develop an inexact inertial-relaxed DR method (Algorithm 7), for which convergence is established in Theorem 3.2.3. In Section 3.3, we use inertial-relaxed DR method to derive a partially inexact relative-error ADMM method (Algorithm 8). The main result of this section is Theorem 3.3.4. In Section 3.4, we performed numerical experiments using our new inexact ADMM method (proposed in Section 3.3) to LASSO and logistic regression problems.

The results of this chapter were published in [3].

3.1 An inertial-relaxed hybrid proximal projection (HPP) method

We begin by developing a new method for the problem (8), i.e.,

$$0 \in T(z), \tag{3.1.1}$$

where $T : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a maximal monotone operator; we assume that this problem has a solution. Our new proposed procedure for this problem, related to the method of [79] but having a new "inertial" step feature, is given below as Algorithm 6.

Algorithm 6. A relative-error inertial-relaxed HPP method for solving (3.1.1)

Initialization: Choose $z^0 = z^{-1} \in \mathbb{R}^n$ and $0 \leq \alpha, \sigma < 1$ and $0 < \underline{\rho} < \overline{\rho} < 2$

for $k = 0, 1, \ldots$ **do**

Choose $\alpha_k \in [0, \alpha]$ and define

$$w^k = z^k + \alpha_k(z^k - z^{k-1}) \tag{3.1.2}$$

Find $(\tilde{z}^k, v^k) \in \mathbb{R}^n \times \mathbb{R}^n$ and $\lambda_k > 0$ such that

$$v^k \in T(\tilde{z}^k), \quad \|\lambda_k v^k + \tilde{z}^k - w^k\|^2 \leq \sigma^2 \left(\|\tilde{z}^k - w^k\|^2 + \|\lambda_k v^k\|^2\right) \tag{3.1.3}$$

If $v^k = 0$, then **stop**. Otherwise, choose $\rho_k \in [\underline{\rho}, \overline{\rho}]$ and set

$$z^{k+1} = w^k - \rho_k \frac{\langle w^k - \tilde{z}^k, v^k \rangle}{\|v^k\|^2} v^k \tag{3.1.4}$$

end for

We make the following remarks concerning this algorithm:

(i) The extrapolation step in (3.1.2) introduces inertial effects — see *e.g.* [1, 2] — controlled by the parameter α_k. The effect of the overrelaxation parameter ρ_k in (3.1.4) is similar but not identical, as shown in Figure 3.1 below. Conditions on $\{\alpha_k\}$, $\alpha \in [0, 1)$ and $\overline{\rho} \in (0, 2)$ that guarantee the convergence of Algorithm 6 are given in Theorem 3.1.5 — see (3.1.20) and (3.1.21) and Figure 3.2 below.

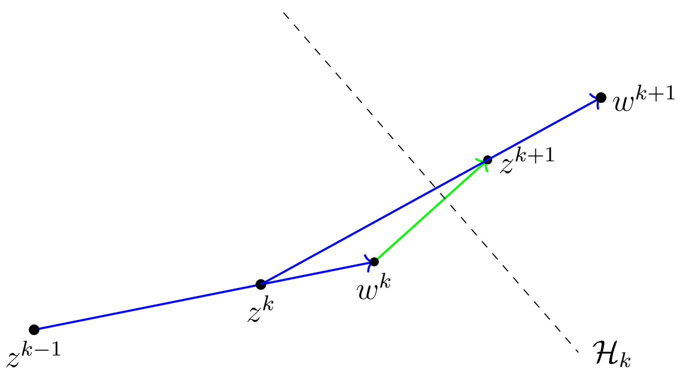


Figure 3.1: Geometric interpretation of steps (3.1.2) and (3.1.4) in Algorithm 6. The overrelaxed projection step (3.1.4) is orthogonal to the separating hyperplane $\mathcal{H}_k$, which can differ from the direction between z^{k-1}, z^k, and w^k when $\alpha_k > 0$.

(ii) If $\alpha = 0$, in which case $\alpha_k \equiv 0$, Algorithm 6 reduces to a special case of the HPP method of [79]; see also [77]. Algorithm 6 is also closely related to the inertial version of the HPP method presented in [1], although that method uses a different relative error criterion. During the reviewing process of [3], one of the referees pointed out that Algorithm 6 is a special instance of Algorithm 1 in [59]. While this is true, it also appears that the convergence analysis in [59, Theorem 2.1] has a flaw: in particular, the key inequality (54) in that analysis reduces to $3 \leq -1$ if one sets $\alpha = -1$, $\gamma = 1$, $\mu = 0$ and $\tau_n \equiv \tau = 1$, so it is unclear whether the convergence result claimed in [59, Theorem 2.1] is valid.

(iii) At each iteration k, condition (3.1.3) is a relative error criterion for the inexact solution of the proximal subproblem $\tilde{z}^k = (I + \lambda_k T)^{-1}(w^k) := J_{\lambda_k T}(w^k)$. If $\sigma = 0$, then this equation must be solved exactly and the pair $(\tilde{z}^k, v^k)$ may be written $(\tilde{z}^k, v^k) = (J_{\lambda_k T}(w^k), \lambda_k^{-1}(w^k - \tilde{z}^k))$. Here, we are primarily concerned with situations in which the calculation of $J_{\lambda_k T}(w^k)$ is relatively difficult and must be approached with an iterative algorithm. In such cases, we use the condition (3.1.3) as an acceptance criterion to truncate such an iterative calculation, possibly saving computational effort. We do not specify the exact form of the iterative algorithm used to produce a pair $(\tilde{z}^k, v^k)$ satisfying (3.1.3), as it depends on the class of problems to which the algorithm is being applied (and thus the structure of the operator T). See [77, 79] for a related discussion; an abstract formalism of the class of algorithm needed to find a solution to (3.1.3) is the "$\mathcal{B}$-procedure" described in [40] and also used in Section 3.2 below.

(iv) The point z^{k+1} in (3.1.4) may be viewed as $z^{k+1} = w^k + \rho_k(P_{\mathcal{H}_k}(w^k) - w^k)$, where $P_{\mathcal{H}_k}$ denotes orthogonal projection onto the hyperplane

$$\mathcal{H}_k := \{z \in \mathbb{R}^n \mid \langle z, v^k \rangle = \langle \tilde{z}^k, v^k \rangle\}, \tag{3.1.5}$$

which strictly separates w^k from the solution set $T^{-1}(0)$ of (3.1.1). This kind of projective approach to approximate proximal point algorithms was pioneered in [77].

(v) Algorithm 6 is an inexact variant of the proximal point algorithm (PPA) [74]. In particular, each of its iterations performs an approximate resolvent calculation subject a relative error criterion, and then executes a projection operation in the manner introduced in [77]; see [76, 79] for related work. The main difference from [77] is the inertial step (3.1.2).

If $v^k = 0$ in Algorithm 6, then it follows from the inclusion in (3.1.3) that $\tilde{z}^k$ is a solution of (3.1.1), that is, $0 \in T(\tilde{z}^k)$, so we halt immediately with the solution $\tilde{z}^k$. For the remainder of this section, we assume that $v^k \not\equiv 0$ and hence that Algorithm 6 generates an infinite sequence of iterates.

The following well-known identity will be useful in the analysis of Algorithm 6:

$$\|(1-\rho)p + \rho q\|^2 = (1-\rho)\|p\|^2 + \rho\|q\|^2 - \rho(1-\rho)\|p-q\|^2 \quad \forall p, q \in \mathbb{R}^n \quad \forall \rho \in \mathbb{R}. \tag{3.1.6}$$

Lemma 3.1.1. [79, Lemma 2] *For each $k \geq 0$, condition* (3.1.3) *implies that*

$$\frac{1-\sigma^2}{1+\sqrt{1-(1-\sigma^2)^2}}\|\tilde{z}^k - w^k\| \leq \|\lambda_k v^k\| \leq \frac{1-\sigma^2}{1-\sqrt{1-(1-\sigma^2)^2}}\|\tilde{z}^k - w^k\|. \tag{3.1.7}$$

An immediate implication of Lemma 3.1.1 is that $v^k = 0$ if and only if $\tilde{z}^k = w^k$.

The proof of the following proposition can be found, using different notation, in [79]. For the convenience of the reader, we also present it here.

Proposition 3.1.2. *Let $\{z^k\}$, $\{\tilde{z}^k\}$ and $\{w^k\}$ be generated by* Algorithm 6 *and define, for all $k \geq 1$,*

$$s_k = (2-\overline{\rho})\max\left\{\overline{\rho}^{-1}\|z^k - w^{k-1}\|^2,\ \underline{\rho}\,(1-\sigma^2)^2\|\tilde{z}^{k-1} - w^{k-1}\|^2\right\}. \tag{3.1.8}$$

Then, for any $z^* \in T^{-1}(0)$,

$$\|z^{k+1} - z^*\|^2 + s_{k+1} \leq \|w^k - z^*\|^2, \qquad \forall k \geq 0. \tag{3.1.9}$$

Proof. We start by defining $\widehat{z}^{k+1}$ as the orthogonal projection of w^k onto the hyperplane $\mathcal{H} := \{z \in \mathbb{R}^n \mid \langle z, v^k\rangle = \langle \tilde{z}^k, v^k\rangle\}$, *i.e.*,

$$\widehat{z}^{k+1} := w^k - \frac{\langle w^k - \tilde{z}^k, v^k\rangle}{\|v^k\|^2}v^k. \tag{3.1.10}$$

Next we show that the hyperplane $\mathcal{H}$ strictly separates the current point w^k from the solution set $\Omega := T^{-1}(0) \neq \emptyset$, that is,

$$\langle w^k, v^k\rangle > \langle \tilde{z}^k, v^k\rangle \geq \langle z^*, v^k\rangle \quad \forall z^* \in \Omega. \tag{3.1.11}$$

To this end, $0 \in T(z^*)$, $v^k \in T(\tilde{z}^k)$ and the monotonicity of T yield $\langle \tilde{z}^k - z^*, v^k\rangle \geq 0$, which is equivalent to the second inequality in (3.1.11). On the other hand, note that from (3.1.3) and the Young inequality $2ab \leq a^2 + b^2$ we have

$$\langle w^k - \tilde{z}^k, v^k\rangle \geq \frac{1-\sigma^2}{2\lambda_k}\left(\|\tilde{z}^k - w^k\|^2 + \|\lambda_k v^k\|^2\right) \geq (1-\sigma^2)\|w^k - \tilde{z}^k\|\|v^k\|,$$

which in turn yields

$$\frac{\langle w^k - \tilde{z}^k, v^k\rangle}{\|v^k\|} \geq (1-\sigma^2)\|w^k - \tilde{z}^k\| > 0. \tag{3.1.12}$$

One consequence of (3.1.12) is the first inequality in (3.1.11), so (3.1.11) must hold.

From (3.1.10) and (3.1.11), we may infer that $\widehat{z}^{k+1}$ is the projection w^k onto the halfspace $\{z \in \mathbb{R}^n \mid \langle z, v^k\rangle \leq \langle \tilde{z}^k, v^k\rangle\}$, which is a convex set containing z^*. The well-known firm nonexpansivess properties of the projection operation then imply that

$$\|w^k - z^*\|^2 - \|\widehat{z}^{k+1} - z^*\|^2 \geq \|w^k - \widehat{z}^{k+1}\|^2. \tag{3.1.13}$$

Algebraic manipulation of (3.1.4) and (3.1.10) yields $z^{k+1} - z^* = (1-\rho_k)(w^k - z^*) + \rho_k(\widehat{z}^{k+1} - z^*)$. Combining this equation with (3.1.6) with $(p, q) = (w^k - z^*, \widehat{z}^{k+1} - z^*)$ gives

$$\|z^{k+1} - z^*\|^2 = (1-\rho_k)\|w^k - z^*\|^2 + \rho_k\|\widehat{z}^{k+1} - z^*\|^2 - \rho_k(1-\rho_k)\|w^k - \widehat{z}^{k+1}\|^2,$$

which after some rearrangement yields

$$\|w^k - z^*\|^2 - \|z^{k+1} - z^*\|^2 = \rho_k \left(\|w^k - z^*\|^2 - \|\widehat{z}^{k+1} - z^*\|^2\right) + \rho_k(1-\rho_k)\|w^k - \widehat{z}^{k+1}\|^2.$$

Using (3.1.13) in the first term on the right-hand side of this identity produces

$$\begin{aligned}
\|w^k - z^*\|^2 - \|z^{k+1} - z^*\|^2 &\geq \rho_k\|w^k - \widehat{z}^{k+1}\|^2 + \rho_k(1-\rho_k)\|w^k - \widehat{z}^{k+1}\|^2 \\
&= (\rho_k + \rho_k(1-\rho_k))\|w^k - \widehat{z}^{k+1}\|^2 \\
&= \rho_k(2-\rho_k)\left(\frac{\langle w^k - \tilde{z}^k, v^k\rangle}{\|v^k\|}\right)^2 \qquad \text{[by (3.1.10)]} \quad (3.1.14) \\
&\geq \rho_k(2-\rho_k)(1-\sigma^2)^2\|w^k - \tilde{z}^k\|^2. \qquad \text{[by (3.1.12)]} \quad (3.1.15)
\end{aligned}$$

To finish the proof, we observe that (3.1.14) and (3.1.4) yield

$$\|w^k - z^*\|^2 - \|z^{k+1} - z^*\|^2 \geq \rho_k^{-1}(2-\rho_k)\|z^{k+1} - w^k\|^2.$$

Combining this inequality with (3.1.15), (3.1.8) and the bounds $\rho_k \in [\,\underline{\rho}, \overline{\rho}\,]$ results in (3.1.9). □

The inequality (3.1.17) presented in the following proposition plays a role in the convergence analysis of inertial proximal algorithms — see *e.g.* [2] — similar to that played by Fejér monotonicity in the analysis of standard proximal algorithms.

Proposition 3.1.3. *Let* $\{z^k\}$, $\{w^k\}$ *and* $\{\alpha_k\}$ *be generated by* Algorithm 6 *and let* $\{s_k\}$ *be as in* (3.1.8). *Further let* $z^* \in T^{-1}(0)$ *and define*

$$(\forall k \geq -1) \quad \varphi_k := \|z^k - z^*\|^2 \quad \text{and} \quad (\forall k \geq 0) \quad \delta_k := \alpha_k(1+\alpha_k)\|z^k - z^{k-1}\|^2. \quad (3.1.16)$$

Then, $\varphi_0 = \varphi_{-1}$ *and*

$$\varphi_{k+1} - \varphi_k + s_{k+1} \leq \alpha_k(\varphi_k - \varphi_{k-1}) + \delta_k \qquad \forall k \geq 0, \qquad (3.1.17)$$

that is, the sequences $\{\varphi_k\}$, $\{s_k\}$, $\{\alpha_k\}$ *and* $\{\delta_k\}$ *satisfy the assumptions of Lemma A.7.*

Proof. From (3.1.2) we obtain $z^k - z^* = (1+\alpha_k)^{-1}(w^k - z^*) + \alpha_k(1+\alpha_k)^{-1}(z^{k-1} - z^*)$, which in conjunction with (3.1.6) and some algebraic manipulation yields

$$\|w^k - z^*\|^2 = (1+\alpha_k)\|z^k - z^*\|^2 - \alpha_k\|z^{k-1} - z^*\|^2 + \alpha_k(1+\alpha_k)\|z^k - z^{k-1}\|^2.$$

Using the above identity and (3.1.16) we obtain, for all $k \geq 0$, that

$$\|w^k - z^*\|^2 = (1+\alpha_k)\varphi_k - \alpha_k\varphi_{k-1} + \delta_k.$$

From (3.1.9) in Proposition 3.1.2 and the definition of φ_k in (3.1.16), the above inequality yields (3.1.17). Finally, $\varphi_0 = \varphi_{-1}$ follows from the initialization $z^0 = z^{-1}$ and the first definition in (3.1.16). □

The following theorem presents our first result on the asymptotic convergence of Algorithm 6 under the summability assumption (3.1.18). Next, Theorem 3.1.5 gives sufficient conditions (3.1.20) and (3.1.21) on the inertial and relaxation parameters to assure that (3.1.18) is satisfied.

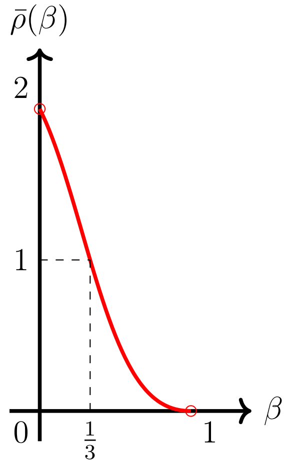


Figure 3.2: The relaxation parameter upper bound $\overline{\rho}(\beta)$ from (3.1.21) as a function of inertial step upper bound $\beta > 0$ of (3.1.20). Note that $\overline{\rho}(1/3) = 1$, while $\overline{\rho}(\beta) > 1$ whenever $\beta < 1/3$.

Theorem 3.1.4 (Convergence of Algorithm 6). *Let* $\{z^k\}$, $\{\tilde{z}^k\}$, $\{v^k\}$, $\{\lambda_k\}$ *and* $\{\alpha_k\}$ *be generated by* Algorithm 6. *If* $\inf_k \lambda_k > 0$ *and*

$$\sum_{k=0}^{\infty} \alpha_k \|z^k - z^{k-1}\|^2 < +\infty \tag{3.1.18}$$

then $\{z^k\}$ *converges to a solution of the monotone inclusion problem* (3.1.1). *Moreover,* $\{\tilde{z}^k\}$ *converges to the same solution and* $\{v^k\}$ *converges to zero.*

Proof. Define $\{s_k\}$ is as in (3.1.8). Using Proposition 3.1.3, (3.1.18), that $\alpha_k \leq \alpha < 1$ for all $k \geq 0$, and Lemma A.7, it follows that (i) $\lim_{k\to\infty} \|z^k - z^*\|$ exist for every $z^* \in \Omega := T^{-1}(0) \neq \emptyset$ and $\sum_{k=1}^{\infty} s_k < +\infty$. So, in particular, $\{z^k\}$ is bounded and (ii) $\lim_{k\to\infty} s_k = 0$. From the form of (3.1.8), that $\lim_{k\to\infty} s_k = 0$, and the assumption that $\inf \lambda_k > 0$, and Lemma 3.1.1, we conclude that

$$\lim_{k\to\infty} \|z^k - w^{k-1}\| = \lim_{k\to\infty} \|\tilde{z}^k - w^k\| = \lim_{k\to\infty} \|v^k\| = 0. \tag{3.1.19}$$

Now let $z^\infty \in \mathbb{R}^n$ be any cluster point of the bounded sequence $\{z^k\}$. By (3.1.19), this point is also a cluster point of $\{w^k\}$ and $\{\tilde{z}^k\}$. Let $\{k_j\}_{j=0}^{\infty}$ be an increasing sequence of indices such that $\tilde{z}^{k_j} \to z^\infty$. We then have

$$(\forall j \geq 0) \quad v^{k_j} \in T(\tilde{z}^{k_j}), \quad \lim_{j\to\infty} v^{k_j} = 0 \text{ and } \lim_{j\to\infty} \tilde{z}^{k_j} = z^\infty,$$

which by the standard closure property of maximal monotone operators yields $z^\infty \in \Omega = T^{-1}(0)$. Hence, the desired result on $\{z^k\}$ follows from (i) and Opial's lemma (stated below as Lemma A.6). On the other hand, the convergence of $\{z^k\}$ and (3.1.19) yields the remaining results regarding $\{\tilde{z}^k\}$ and $\{v^k\}$. □

Theorem 3.1.5 (Convergence of Algorithm 6). *Let $\{z^k\}$, $\{\alpha_k\}$ and $\{\lambda_k\}$ be generated by* Algorithm 6. *Assume that $\alpha \in [0,1)$, $\overline{\rho} \in (0,2)$ and $\{\alpha_k\}$ satisfy the following (for some $\beta > 0$):*

$$0 \le \alpha_k \le \alpha_{k+1} \le \alpha < \beta < 1 \qquad \forall k \ge 0 \tag{3.1.20}$$

and

$$\overline{\rho} = \overline{\rho}(\beta) := \frac{2(\beta-1)^2}{2(\beta-1)^2 + 3\beta - 1}. \tag{3.1.21}$$

Then,

$$\sum_{k=1}^{\infty} \|z^k - z^{k-1}\|^2 < +\infty. \tag{3.1.22}$$

As a consequence, it follows that under the assumptions (3.1.20) *and* (3.1.21) *the sequence $\{z^k\}$ generated by* Algorithm 6 *converges to a solution of the monotone inclusion problem* (3.1.1) *whenever* $\inf \lambda_k > 0$. *Moreover, under the above assumptions, $\{\tilde{z}^k\}$ converges to the same solution and $\{v^k\}$ converges to zero.*

Proof. Using (3.1.2), the Cauchy-Schwarz inequality and the Young inequality $2ab \le a^2 + b^2$ with $a := \|z^{k+1} - z^k\|$ and $b := \|z^k - z^{k-1}\|$ we find

$$\begin{aligned} \|z^{k+1} - w^k\|^2 &= \|z^{k+1} - z^k\|^2 + \alpha_k^2 \|z^k - z^{k-1}\|^2 - 2\alpha_k \langle z^{k+1} - z^k, z^k - z^{k-1}\rangle \\ &\ge \|z^{k+1} - z^k\|^2 + \alpha_k^2 \|z^k - z^{k-1}\|^2 - \alpha_k \left(2\|z^{k+1} - z^k\| \|z^k - z^{k-1}\|\right) \\ &\ge (1-\alpha_k)\|z^{k+1} - z^k\|^2 - \alpha_k(1-\alpha_k)\|z^k - z^{k-1}\|^2. \end{aligned} \tag{3.1.23}$$

Starting with a rearrangement of (3.1.17), we then obtain

$$\begin{aligned} &\varphi_{k+1} - \varphi_k - \alpha_k(\varphi_k - \varphi_{k-1}) \le \delta_k - s_{k+1} \\ &\quad \le \alpha_k(1+\alpha_k)\|z^k - z^{k-1}\|^2 - (2-\overline{\rho})\overline{\rho}^{-1}\|z^{k+1} - w^k\|^2 \qquad \text{[by (3.1.8) and (3.1.16)]} \\ &\quad \le \alpha_k(1+\alpha_k)\|z^k - z^{k-1}\|^2 \\ &\qquad\qquad - (2-\overline{\rho})\overline{\rho}^{-1}\left[(1-\alpha_k)\|z^{k+1} - z^k\|^2 - \alpha_k(1-\alpha_k)\|z^k - z^{k-1}\|^2\right] \qquad \text{[by (3.1.23)]} \\ &\quad = -(2-\overline{\rho})\overline{\rho}^{-1}(1-\alpha_k)\|z^{k+1} - z^k\|^2 + \left[\alpha_k(1+\alpha_k) + (2-\overline{\rho})\overline{\rho}^{-1}\alpha_k(1-\alpha_k)\right]\|z^k - z^{k-1}\|^2 \\ &\quad = -(2-\overline{\rho})\overline{\rho}^{-1}(1-\alpha_k)\|z^{k+1} - z^k\|^2 + \gamma_k\|z^k - z^{k-1}\|^2, \end{aligned} \tag{3.1.24}$$

where

$$\gamma_k := -2(\overline{\rho}^{\,-1} - 1)\alpha_k^2 + 2\,\overline{\rho}^{\,-1}\alpha_k \qquad \forall k \ge 0. \tag{3.1.25}$$

Some elementary algebraic manipulations of (3.1.24) then yield

$$\varphi_{k+1} - \varphi_k - \alpha_k(\varphi_k - \varphi_{k-1}) - \gamma_k\|z^k - z^{k-1}\|^2 \le -(2\,\overline{\rho}^{\,-1} - 1)(1-\alpha_k)\|z^{k+1} - z^k\|^2 \quad \forall k \ge 0. \tag{3.1.26}$$

Define now the scalar function:

$$q(\nu) := 2(\overline{\rho}^{\,-1} - 1)\nu^2 - (4\overline{\rho}^{\,-1} - 1)\nu + 2\overline{\rho}^{\,-1} - 1, \tag{3.1.27}$$

and

$$\begin{aligned} &\mu_0 := (1-\alpha_0)\varphi_0 \geq 0, \\ &\mu_k := \varphi_k - \alpha_{k-1}\varphi_{k-1} + \gamma_k \|z^k - z^{k-1}\|^2 \quad \forall k \geq 1, \end{aligned} \tag{3.1.28}$$

where φ_k is as in (3.1.16). Using (3.1.26)-(3.1.28) and the assumption that $\{\alpha_k\}$ is nondecreasing — see (3.1.20) — we obtain, for all $k \geq 0$,

$$\begin{aligned} \mu_{k+1} - \mu_k &\leq \left[\varphi_{k+1} - \varphi_k - \alpha_k(\varphi_k - \varphi_{k-1}) - \gamma_k \|z^k - z^{k-1}\|^2\right] + \gamma_{k+1}\|z^{k+1} - z^k\|^2 \\ &\leq \left[\gamma_{k+1} - (2\overline{\rho}^{-1} - 1)(1 - \alpha_{k+1})\right] \|z^{k+1} - z^k\|^2 \\ &= -\left[2(\overline{\rho}^{-1} - 1)\alpha_{k+1}^2 - (4\overline{\rho}^{-1} - 1)\alpha_{k+1} + 2\overline{\rho}^{-1} - 1\right] \|z^{k+1} - z^k\|^2 \\ &= -q(\alpha_{k+1})\|z^{k+1} - z^k\|^2. \end{aligned} \tag{3.1.29}$$

We will now show that $q(\alpha_{k+1})$ admits a uniform positive lower bound. To this end, note first that from (3.1.21) and Lemma A.4 that we have

$$\beta = \frac{2(2-\overline{\rho})}{4 - \overline{\rho} + \sqrt{\overline{\rho}(16 - 7\overline{\rho})}}.$$

Using the latter identity, (3.1.27), and Lemma A.5 with $a = 2(\overline{\rho}^{-1} - 1)$, $b = 4\overline{\rho}^{-1} - 1$, and $c = 2\overline{\rho}^{-1} - 1$, we conclude that $q(\cdot)$ is decreasing in $[0, \beta]$ and $\beta > 0$ is a root of $q(\cdot)$. Thus, in view of (3.1.20), we conclude that

$$q(\alpha_{k+1}) \geq q(\alpha) > q(\beta) = 0, \tag{3.1.30}$$

which gives the desired uniform positive lower bound on $q(\alpha_{k+1})$.

Using (3.1.29) and (3.1.30) we find

$$\|z^{k+1} - z^k\|^2 \leq \frac{1}{q(\alpha)}(\mu_k - \mu_{k+1}), \quad \forall k \geq 0, \tag{3.1.31}$$

which, in turn, combined with (3.1.20) and the definition of μ_k in (3.1.28), gives

$$\begin{aligned} \sum_{j=0}^{k} \|z^{j+1} - z^j\|^2 &\leq \frac{1}{q(\alpha)}(\mu_0 - \mu_{k+1}), \\ &\leq \frac{1}{q(\alpha)}(\mu_0 + \alpha\varphi_k) \quad \forall k \geq 0. \end{aligned} \tag{3.1.32}$$

Note now that (3.1.31), (3.1.20) and (3.1.28) also yield

$$\begin{aligned} \mu_0 \geq \ldots \geq \mu_{k+1} &= \varphi_{k+1} - \alpha_k\varphi_k + \gamma_{k+1}\|z^{k+1} - z^k\|^2 \\ &\geq \varphi_{k+1} - \alpha\varphi_k, \quad \forall k \geq 0, \end{aligned}$$

and so,

$$\varphi_{k+1} \leq \alpha^{k+1}\varphi_0 + \frac{\mu_0}{1-\alpha} \leq \varphi_0 + \frac{\mu_0}{1-\alpha} \qquad \forall k \geq -1. \tag{3.1.33}$$

Hence, (3.1.22) follows directly from (3.1.32) and (3.1.33). On the other hand, the second statement of the theorem follows from (3.1.22) and Theorem 3.1.4 (recall that $\alpha_k \leq \alpha < 1$ for all $k \geq 0$). □

We close this section with a few further remarks about the analysis of Algorithm 6:

(i) If we set $\beta = 1/3$ in (3.1.20), then it follows immediately from (3.1.21) that $\overline{\rho} = 1$. On the other hand, we have $\overline{\rho} > 1$ in (3.1.21) whenever $\beta < 1/3$ (see also Figure 3.2). Setting $\beta = 1/3$ in (3.1.20) corresponds to the standard strategy in the literature of inertial proximal algorithms; see *e.g.* [2, 26].

(ii) Conditions (3.1.20) and (3.1.21) on $\{\alpha_k\}$, α and $\overline{\rho}$ guarantee that the summability condition (3.1.18) is satisfied, thus guaranteeing the convergence of Algorithm 6. Similar conditions were also recently proposed and studied in [5, 8]. Since Algorithm 6 is be the basis of the DR and ADMM methods that will be developed in the Chapter 3, conditions (3.1.20) and (3.1.21) will also play an important role in their convergence analysis.

3.2 A partially inexact inertial-relaxed Douglas-Rachford (DR) algorithm

Consider problem (11), i.e., the problem of finding $z \in \mathbb{R}^n$ such that

$$0 \in A(z) + B(z) \tag{3.2.1}$$

where:

(F1) A and B are (set-valued) maximal monotone operators on $\mathbb{R}^n$;

(F2) the solution set $(A + B)^{-1}(0)$ of (3.2.1) is nonempty.

As mentioned previously, a popular operator splitting algorithms for finding approximate solutions to (3.2.1) is the Douglas-Rachford (DR) algorithm [56, 37], the iteration of the method is given in (13), i.e.,

$$z^{k+1} = J_{\gamma A}\big(2J_{\gamma B}(z^k) - z^k\big) + z^k - J_{\gamma B}(z^k) \quad \forall k \geq 0, \tag{3.2.2}$$

where $\gamma > 0$ is a scaling parameter, z^k is the current iterate and $J_{\gamma A} = (\gamma A + I)^{-1}$ and $J_{\gamma B} = (\gamma B + I)^{-1}$ are the resolvent operators of A and B, respectively. The DR algorithm (3.2.2) is a splitting algorithm for solving the (structured) inclusion (3.2.1) in the sense that the resolvents $J_{\gamma A}$ and $J_{\gamma B}$ are employed separately, but the resolvent $J_{\gamma(A+B)}$ of $A + B$ is not. Such methods are useful in the situations in which the values of $J_{\gamma A}$ and $J_{\gamma B}$ are relatively easy to evaluate in comparison to those of $J_{\gamma(A+B)}$.

This section will develop an inexact version of the DR algorithm (3.2.2) for the situation in which the resolvent of one of the operators, say B, is relatively hard, but evaluating $J_{\gamma A}$ is a simple calculation. To this end, we consider the following equivalent formulation of (3.2.2) (see, *e.g.*, [37]) : given some $r^k, b^k \in \mathbb{R}^n$,

$$\text{Find } (s^{k+1}, b^{k+1}) \in B \text{ such that } s^{k+1} + \gamma b^{k+1} = r^k + \gamma b^k; \tag{3.2.3}$$

$$\text{Find } (r^{k+1}, a^{k+1}) \in A \text{ such that } r^{k+1} + \gamma a^{k+1} = s^{k+1} - \gamma b^{k+1}. \tag{3.2.4}$$

In this case, $z^k = r^k + \gamma b^k$. Since the resolvent $J_{\gamma A}$ of A is assumed to be easily computable, the pair (r^{k+1}, a^{k+1}) in (3.2.4) is explicitly given by

$$r^{k+1} = J_{\gamma A}(s^{k+1} - \gamma b^{k+1})$$

and

$$a^{k+1} = \gamma^{-1}(s^{k+1} - r^{k+1}) - b^{k+1}.$$

For B, we by contrast suppose that exact computation of the pair (s^{k+1}, b^{k+1}) satisfying (3.2.3) requires a relatively time-consuming iterative process, which we model immediately below by the notion of a $\mathcal{B}$-procedure as introduced in [40]. We first remark that (3.2.3) can be posed in the more general framework of solving monotone inclusion problems of the form

$$0 \in s + \gamma B(s) - (r + \gamma b), \tag{3.2.5}$$

where $r, b \in \mathbb{R}^n$ and $\gamma > 0$.

Definition 3.2.1 (**$\mathcal{B}$–procedure for solving** (3.2.5))**.** *A $\mathcal{B}$–procedure for (approximately) solving any instance of* (3.2.5) *is a mapping $\mathcal{B} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{N}^* \to \mathbb{R}^n \times \mathbb{R}^n$ such that if one lets $(s^\ell, b^\ell) = \mathcal{B}(r, b, \gamma, \bar{s}, \bar{b}, \ell)$ for all $\ell \in \mathbb{N}^*$ and any given $r, b, \bar{s}, \bar{b} \in \mathbb{R}^n$ and $\gamma > 0$, then $b^\ell \in B(s^\ell)$, for all $\ell \in \mathbb{N}^*$, the sequence $\{(s^\ell, b^\ell)\}$ is convergent, and $s^\ell + \gamma b^\ell \to r + \gamma b$.*

Following [40], the intuitive meaning of $(s^\ell, b^\ell) = \mathcal{B}(r, b, \gamma, \bar{s}, \bar{b}, \ell)$ is that (s^ℓ, b^ℓ) is the ℓ^{th} trial approximation generated by some iterative procedure for solving (3.2.5), starting from some initial guess $(\bar{s}, \bar{b}) \in \mathbb{R}^n \times \mathbb{R}^n$. We refer the interested reader to [40, Section 5] for a more detalied discussion and interpretation on the $\mathcal{B}$-procedure concept.

We make the following standing assumption:

Assumption 1. *There exists a $\mathcal{B}$-procedure (according to Definition 3.2.1) for approximately solving any instance of* (3.2.5)*.*

We now combine the hypothesized $\mathcal{B}$-procedure with an acceptance criterion for the approximate solution of (3.2.3). We will follow the general approach of [40], which is to exploit the connection between the DR algorithm (3.2.3)-(3.2.4) and the proximal point algorithm as established in [37]. Specifically, the DR algorithm (3.2.3), (3.2.4) is a special instance of the PP algorithm in the sense that,

$$r^{k+1} + \gamma b^{k+1} = (S_{\gamma,A,B} + I)^{-1}(r^k + \gamma b^k) \qquad \forall k \geq 0$$

where the "splitting" operator $S_{\gamma,A,B}$ is defined as (1.2.6), i.e.,

$$S_{\gamma,A,B} := \{(r + \gamma b, s - r) \in \mathbb{R}^n \times \mathbb{R}^n \,|\, b \in B(s), \;\; a \in A(r), \;\; \gamma a + r = s - \gamma b\}. \tag{3.2.14}$$

As we presented in Subsection 1.2.1, the operator defined in (3.2.14) is maximal monotone and

$$(A + B)^{-1}(0) = J_{\gamma B}\big(S_{\gamma,A,B}^{-1}(0)\big), \tag{3.2.15}$$

which, in particular, gives that any solution $z^* \in \mathbb{R}^n$ of the monotone inclusion problem (3.1.1) with $T := S_{\gamma,A,B}$, namely

$$0 \in S_{\gamma,A,B}(z) \tag{3.2.16}$$

yields a solution $x^* := J_{\gamma B}(z^*)$ of (3.2.1).

Here, we follow a similar derivation to [40], but use Algorithm 6 of Section 3.1 to (3.2.16) in place of the HPE method of [76]. The result is an inertial-relaxed inexact relative-error DR algorithm for solving (3.2.1). We should emphasize that even when $\alpha_k \equiv 0$ (there is no inertial step) and $\rho_k \equiv 1$ (no overrelaxation), the resulting algorithm differs from that of [40]. This difference arises because the underlying "convergence engine" of Algorithm 6 is a form of hybrid proximal-projection (HPP) algorithm, whereas [40] used an HPE algorithm in the equivalent role, using an extragradient step instead of projection.

The proposed algorithm for solving (3.2.1) is shown as Algorithm 7. We should mention that a different inexact DR splitting algorithm in which relative errors are allowed in both (3.2.7) and (3.2.9) was recently proposed and studied in [82], but without computational testing. The following proposition shows that Algorithm 7 is indeed a special instance of Algorithm 6 for solving (3.1.1) with $T := S_{\gamma,A,B}$.

Proposition 3.2.2. *Consider the sequences evolved by Algorithm 7 and for each $k \geq 0$ let $\ell(k)$ denote the value of ℓ for which* (3.2.10) *is satisfied. For each $k \geq 0$, define, with γ as in Algorithm 7,*

$$\begin{aligned} z^k &:= r^k + \gamma b^k, \\ w^k &:= \hat{r}^k + \gamma \hat{b}^k, \\ \tilde{z}^k &:= r^{k,\ell(k)} + \gamma b^{k,\ell(k)}, \\ v^k &:= s^{k,\ell(k)} - r^{k,\ell(k)}. \end{aligned} \tag{3.2.17}$$

Then these latter sequences satisfy the conditions (3.1.2)-(3.1.4) *of* Algorithm 6 *with $\lambda_k \equiv 1$ and $T = S_{\gamma,A,B}$.*

Algorithm 7. A partially inexact inertial–relaxed Douglas-Rachford splitting algorithm for solving (3.2.1)

Choose $\gamma > 0$, $0 \le \alpha, \sigma < 1$ and $0 < \underline{\rho} < \overline{\rho} < 2$.
Initialize $(s^0, b^0, r^0) = (s^{-1}, b^{-1}, r^{-1}) \in (\mathbb{R}^n)^3$.

for $k = 0, 1, 2, \ldots$ **do**

Choose $\alpha_k \in [0, \alpha]$ and define

$$(\hat{s}^k, \hat{b}^k, \hat{r}^k) = (s^k, b^k, r^k) + \alpha_k \left[(s^k, b^k, r^k) - (s^{k-1}, b^{k-1}, r^{k-1})\right] \tag{3.2.6}$$

repeat {**for** $\ell = 1, 2, \ldots$ }

Improve the solution to

$$b^{k,\ell} \in B(s^{k,\ell}), \quad s^{k,\ell} + \gamma b^{k,\ell} \approx \hat{r}^k + \gamma \hat{b}^k \tag{3.2.7}$$

by setting

$$(s^{k,\ell}, b^{k,\ell}) = \mathcal{B}(\hat{r}^k, \hat{b}^k, \gamma, \hat{s}^k, \hat{b}^k, \ell) \tag{3.2.8}$$

(thus incrementally executing a step of the $\mathcal{B}$–procedure)

Exactly find $(r^{k,\ell}, a^{k,\ell}) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$a^{k,\ell} \in A(r^{k,\ell}), \quad r^{k,\ell} + \gamma a^{k,\ell} = s^{k,\ell} - \gamma b^{k,\ell} \tag{3.2.9}$$

until

$$\|s^{k,\ell} + \gamma b^{k,\ell} - (\hat{r}^k + \gamma \hat{b}^k)\|^2 \le \sigma^2 \left(\|r^{k,\ell} + \gamma b^{k,\ell} - (\hat{r}^k + \gamma \hat{b}^k)\|^2 + \|s^{k,\ell} - r^{k,\ell}\|^2 \right) \tag{3.2.10}$$

if $s^{k,\ell} = r^{k,\ell}$, then **stop**

otherwise, choose $\rho_k \in [\underline{\rho}, \overline{\rho}]$ and set

$$s^{k+1} = s^{k,\ell}, \quad r^{k+1} = r^{k,\ell} \tag{3.2.11}$$

$$\theta_{k+1} = \frac{\langle (\hat{r}^k - r^{k,\ell}) + \gamma(\hat{b}^k - b^{k,\ell}), s^{k,\ell} - r^{k,\ell} \rangle}{\|s^{k,\ell} - r^{k,\ell}\|^2} \tag{3.2.12}$$

$$b^{k+1} = \hat{b}^k - \gamma^{-1} \left[(1 - \rho_k \theta_{k+1}) r^{k+1} + \rho_k \theta_{k+1} s^{k+1} - \hat{r}^k\right] \tag{3.2.13}$$

end for

Proof. Fix any $k \geq 0$. From (3.2.6) and the definitions of z^k and w^k in (3.2.17) we have

$$\begin{aligned} w^k &= \hat{r}^k + \gamma \hat{b}^k = r^k + \gamma b^k + \alpha_k[r^k + \gamma b^k - (r^{k-1} + \gamma b^{k-1})] \\ &= z^k + \alpha_k(z^k - z^{k-1}), \end{aligned}$$

which is exactly (3.1.2). Now note that the inclusion in (3.1.3) follows from the fact that $T := S_{\gamma,A,B}$, (3.2.14), (3.2.9), $b^{k,\ell(k)} \in B(s^{k,\ell(k)})$ from (3.2.7), and the definitions of v^k and $\tilde{z}^k$ in (3.2.17).

Further, (3.2.17) and (3.2.10) yield

$$\begin{aligned} \|v^k + \tilde{z}^k - w^k\|^2 &= \|s^{k,\ell(k)} + \gamma b^{k,\ell(k)} - (\hat{r}^k + \gamma \hat{b}^k)\|^2 \\ &\leq \sigma^2 \left(\|r^{k,\ell(k)} + \gamma b^{k,\ell(k)} - (\hat{r}^k + \gamma \hat{b}^k)\|^2 + \|s^{k,\ell} - r^{k,\ell}\|^2 \right) \\ &= \sigma^2 \left(\|\tilde{z}^k - w^k\|^2 + \|v^k\|^2 \right), \end{aligned}$$

which is exactly the inequality in (3.1.3) with $\lambda_k = 1$. Finally,

$$\begin{aligned} z^{k+1} &= r^{k+1} + \gamma b^{k+1} && \text{[by (3.2.17)]} \\ &= r^{k,\ell(k)} + \gamma \left(\hat{b}^k - \frac{1}{\gamma} \left[(1 - \rho_k \theta_{k+1}) r^{k,\ell(k)} + \rho_k \theta_{k+1} s^{k,\ell(k)} - \hat{r}^k \right] \right) && \text{[by (3.2.11) and (3.2.13)]} \\ &= \hat{r}^k + \gamma \hat{b}^k + \rho_k \theta_{k+1} \left(r^{k,\ell(k)} - s^{k,\ell(k)} \right) \\ &= w^k - \rho_k \theta_{k+1} v^k && \text{[by (3.2.17)]} \\ &= w^k - \rho_k \frac{\langle w^k - \tilde{z}^k, v^k \rangle}{\|v^k\|^2} v^k, && \text{[by (3.2.12) and (3.2.17)]} \end{aligned}$$

which establishes (3.1.4) and thus completes the proof of the proposition. □

The following theorem states the asymptotic convergence properties of Algorithm 7, which are essentially direct consequences of Proposition 3.2.2 and Theorem 3.1.5.

Theorem 3.2.3 (Convergence of Algorithm 7). *Consider the sequences evolved by Algorithm 7 with the parameters $\alpha \in [0, 1)$, $\overline{\rho} \in (0, 2)$ and $\{\alpha_k\}$ satisfying the conditions* (3.1.20) *and* (3.1.21) *of Theorem 3.1.5. Then*

(a) *If the outer loop (over k) executes an infinite number of times, with each inner loop (over ℓ) terminating in a finite number of iterations $\ell = \ell(k)$, then $\{s^k\}$ and $\{r^k\}$ both converge to some solution $x^* \in \mathbb{R}^n$ of* (3.2.1)*, and $\{b^{k,\ell(k)}\}$ and $\{b^k\}$ both converge to some $b^* \in B(x^*)$, with $\{a^{k,\ell(k)}\}$ converging to $-b^* \in A(x^*)$.*

(b) *If the outer loop executes only a finite number of times, ending with $k = \bar{k}$, with the last invocation of the inner loop executing an infinite number of times, then $\{s^{\bar{k},\ell}\}_{\ell=1}^\infty$ and $\{r^{\bar{k},\ell}\}_{\ell=1}^\infty$ both converge to some solution $x^* \in \mathbb{R}^n$ of* (3.2.1)*, and $\{b^{\bar{k},\ell}\}_{\ell=1}^\infty$ converges to some $b^* \in B(x^*)$, with $\{a^{\bar{k},\ell}\}_{\ell=1}^\infty$ converging to $-b^* \in A(x^*)$.*

(c) *If* Algorithm 7 *stops with $s^{k,\ell} = r^{k,\ell}$, then $z^* := s^{k,\ell} = r^{k,\ell}$ is a solution of* (3.2.1).

Proof. (a) For each $k \geq 0$, again let $\ell = \ell(k)$ be the index of inner iteration that first meets the inner-loop termination condition. Using Proposition 3.2.2, (3.2.11), the descriptions of algorithms 6 and 7, and Theorem 3.1.5, we conclude that there exists $z^* \in \mathbb{R}^n$ such that $0 \in S_{\gamma,A,B}(z^*)$ and

$$z^k = r^k + \gamma b^k \to z^* \quad \tilde{z}^{k-1} = r^k + \gamma b^{(k-1),\ell(k-1)} \to z^* \quad v^{k-1} = s^k - r^k \to 0. \quad (3.2.18)$$

From $0 \in S_{\gamma,A,B}(z^*)$ and (3.2.15) we obtain that $x^* := J_{\gamma B}(z^*)$ is a solution of (3.2.1). Moreover, it follows from (3.2.18), the inclusion in (3.2.7), (3.2.11), and the continuity of $J_{\gamma B}$ that

$$s^k + \gamma b^{(k-1),\ell} = v^{k-1} + \tilde{z}^{(k-1)} \to 0 + z^* = z^* \quad (3.2.19)$$

and

$$s^k = J_{\gamma B}(s^k + \gamma b^{(k-1),\ell}) \to J_{\gamma B}(z^*) = x^*. \quad (3.2.20)$$

We also have $r^k \to x^*$ since, from (3.2.18), $s^k - r^k \to 0$. Altogether, we have that x^* is a solution of (3.2.1) and $\{s^k\}$ and $\{r^k\}$ both converge to x^*. From (3.2.19) and (3.2.20) we now have

$$b^{k,\ell(k)} = \gamma^{-1}(s^{k+1} + \gamma b^{k,\ell(k)} - s^{k+1}) \to \gamma^{-1}(z^* - x^*) := b^*. \quad (3.2.21)$$

From $x^* = J_{\gamma B}(z^*)$ we then obtain $b^* \in B(x^*)$. On the other hand, using the equation in (3.2.9), (3.2.11), (3.2.18) and (3.2.21) we find

$$a^{k,\ell(k)} = \gamma^{-1}(s^{k+1} - r^{k+1}) - b^{k,\ell(k)} \to 0 - b^* = -b^*.$$

Using the above convergence result, that $r^{k,\ell(k)} = r^{k+1} \to x^*$, the inclusion in (3.2.9), and Lemma A.3, we obtain that $-b^* \in A(x^*)$. Finally, $b^k = \gamma^{-1}(z^k - r^k) \to \gamma^{-1}(z^* - r^*) = b^*$.

(b) First note that using (3.2.8) we obtain $(s^{\bar{k},\ell}, b^{\bar{k},\ell}) = \mathcal{B}(\hat{r}^{\bar{k}}, \hat{b}^{\bar{k}}, \gamma, \hat{s}^{\bar{k}}, \hat{b}^{\bar{k}}, \ell)$, which in view of Definition 3.2.1 yields $(s^{\bar{k},\ell}, b^{\bar{k},\ell}) \in B$, for all $\ell \geq 1$, $s^{\bar{k},\ell} + \gamma b^{\bar{k},\ell} \to \hat{r}^{\bar{k}} + \gamma \hat{b}^{\bar{k}}$, $s^{\bar{k},\ell} \to x^*$, and $b^{\bar{k},\ell} \to b^*$, for some $x^*, b^* \in \mathbb{R}^n$. Combining limits, we obtain that $\hat{r}^{\bar{k}} + \gamma \hat{b}^{\bar{k}} = x^* + \gamma b^*$. From Lemma A.3, we also have $b^* \in B(x^*)$. Now combining the limits with (3.2.9) and the continuity of $J_{\gamma A}$, we also find

$$r^{\bar{k},\ell} = J_{\gamma A}(s^{\bar{k},\ell} - \gamma b^{\bar{k},\ell}) \to J_{\gamma A}(x^* - \gamma b^*) =: r^*$$

and so

$$a^{\bar{k},\ell} = \gamma^{-1}\left(s^{\bar{k},\ell} - r^{\bar{k},\ell}\right) - b^{\bar{k},\ell} \to \gamma^{-1}(x^* - r^*) - b^* =: a^*. \quad (3.2.22)$$

From the inclusion in (3.2.9) and (again) Lemma A.3 we obtain that $a^* \in A(r^*)$. On the other hand, using (3.2.10) and the hypothesis that the inner loop executes an infinite number of times at iteration $k = \bar{k}$, we obtain, for all $\ell \geq 1$, that

$$\|s^{\bar{k},\ell} + \gamma b^{\bar{k},\ell} - (\hat{r}^{\bar{k}} + \gamma \hat{b}^{\bar{k}})\|^2 > \sigma^2 \left(\|r^{\bar{k},\ell} + \gamma b^{\bar{k},\ell} - (\hat{r}^{\bar{k}} + \gamma \hat{b}^{\bar{k}})\|^2 + \|s^{\bar{k},\ell} - r^{\bar{k},\ell}\|^2\right). \quad (3.2.23)$$

Since the left-hand side of the above inequality converges to zero and the right-hand side is nonnegative, the right-hand side also converges to zero and in particular $s^{\bar{k},\ell} - r^{\bar{k},\ell} \to 0$. Since $s^{\bar{k},\ell} \to x^*$ and $r^{\bar{k},\ell} \to r^*$, we conclude that $x^* = r^*$ and, hence, from (3.2.22), that $a^* = -b^*$.

(c) If $s^{k,\ell} = r^{k,\ell} =: z^*$, then it follows from the inclusion in (3.2.7) and (3.2.9) that $0 = \gamma^{-1}(s^{k,\ell} - r^{k,\ell}) = a^{k,\ell} + b^{k,\ell} \in A(r^{k,\ell}) + B(s^{k,\ell}) = (A + B)(z^*)$. □

3.3 A partially inexact relative-error inertial-relaxed ADMM

We now consider the convex optimization problem (18), i.e.,

$$\min_{x\in\mathbb{R}^n} \{f(x)+g(x)\} \tag{3.3.1}$$

where $f, g : \mathbb{R}^n \to (-\infty, \infty]$ are proper, convex and lower semicontinuous functions for which $(\partial f + \partial g)^{-1}(0) \neq \emptyset$.

The alternating direction method of multipliers (ADMM) [45] is a first-order algorithm for solving (3.3.1) which has become popular over the last decade largely due to its wide range of applications in data science (see, *e.g.*, [17]). As applied to (3.3.1), one iteration of the ADMM may be described as in (19)-(20), that is:

$$x^{k+1} \in \operatorname{argmin}_{x\in\mathbb{R}^n} \left\{ f(x) + \langle p^k, x\rangle + \frac{c}{2}\|x - z^k\|^2 \right\}, \tag{3.3.2}$$

$$z^{k+1} \in \operatorname{argmin}_{z\in\mathbb{R}^n} \left\{ g(z) - \langle p^k, z\rangle + \frac{c}{2}\|x^{k+1} - z\|^2 \right\}, \tag{3.3.3}$$

$$p^{k+1} = p^k + c(x^{k+1} - z^{k+1}). \tag{3.3.4}$$

In many applications, the function g is such that (3.3.3) has a closed-form or otherwise straightforward solution (*e.g.*, $g(\cdot) = \|\cdot\|_1$). We consider situations in which this is the case, but solving (3.3.2) is more difficult and requires some form of iterative process. Eckstein and Yao [40, Section 6] proposed and studied the asymptotic convergence of an inexact version of the ADMM tailored to such situations: at each iteration, (3.3.2) may be approximately solved within a relative-error tolerance. This method is a special version of their inexact relative-error Douglas-Rachford (DR) algorithm mentioned in Section 2.1, as applied to the monotone inclusion (22), i.e.,

$$0 \in \partial f(x) + \partial g(x) \tag{3.3.5}$$

which is, in particular, a special case of (3.2.1) with $A = \partial g$ and $B = \partial f$ (or *vice versa*). Problem (3.3.5) is, under standard qualification conditions, equivalent to (3.3.1). Recall that we are assuming $(\partial f + \partial g)^{-1}(0) \neq \emptyset$, *i.e.*, that (3.3.5) admits at least one solution.

In this section, we propose and study the asymptotic behaviour of a (partially) inexact relative-error *inertial-relaxed* ADMM algorithm for solving (3.3.1). The proposed method, namely Algorithm 8, is a special version of Algorithm 7 when applied to solving (3.3.5) and may be viewed as an alternative to the Eckstein-Yao approximate ADMM [40] that incorporates inertial and relaxation effects to accelerate convergence.

To formalize the inexact solution process for the subproblems (3.3.2), we introduce the notion of an $\mathcal{F}$-procedure [40]. First, we note that any instance of (3.3.2) can be posed slightly more abstractly as

$$\min_{x\in\mathbb{R}^n} \left\{ f(x) + \langle p, x\rangle + \frac{c}{2}\|x - z\|^2 \right\} \tag{3.3.6}$$

where $p, z \in \mathbb{R}^n$ and $c > 0$.

Definition 3.3.1 (**$\mathcal{F}$-procedure for solving** (3.3.6))**.** *A $\mathcal{F}$–procedure for (approximately) solving any instance of* (3.3.6) *is a mapping $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{N}^* \to \mathbb{R}^n \times \mathbb{R}^n$ such that if one lets $(x^\ell, y^\ell) = \mathcal{F}(p, z, c, \bar{x}, \ell)$ for all $\ell \in \mathbb{N}$ and any given $p, z, \bar{x} \in \mathbb{R}^n$ and $c > 0$, then*

$$\lim_{\ell\to\infty} y^\ell = 0 \ \ \textit{and} \ \ (\forall \ell \in \mathbb{N}) \ \ y^\ell \in \partial_x \left[f(x) + \langle p, x\rangle + \frac{c}{2}\|x - z\|^2 \right]_{x=x^\ell}. \tag{3.3.7}$$

Quoting [40, Assumption 2], "the idea behind this definition is that $\mathcal{F}(p, z, c, \bar{x}, \ell)$ is the ℓ^{th} iterate produced by the x-subproblem solution procedure with penalty parameter c, the Lagrange multiplier estimate p^k equal to p, and $z^k = z$, starting from the solution estimate $\bar{x}$". For the remainder of this section, we assume the following.

Assumption 2. *There exists a $\mathcal{F}$–procedure (according to* Definition 3.3.1*) for approximately solving any instance of* (3.3.6)*.*

The next lemma shows that the $\mathcal{F}$-procedure is essentially a form of $\mathcal{B}$–procedure (see Definition 3.2.1). Although the proof essentially duplicates analysis in [39, 40], it is not presented as a separate result there. Therefore we include the proof in the interest of rigor and completeness.

Lemma 3.3.2. *Let $\mathcal{F}(\cdot) = (\mathcal{F}_1(\cdot), \mathcal{F}_2(\cdot))$ be a $\mathcal{F}$–procedure for solving* (3.3.6)*, where $\mathcal{F}_i : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{N}^* \to \mathbb{R}^n$, for $i = 1, 2$, and define $\mathcal{B} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{N}^* \to \mathbb{R}^n \times \mathbb{R}^n$ by*

$$\mathcal{B}(r, b, \gamma, \bar{s}, \bar{b}, \ell) = \mathcal{F}(-b, r, \gamma^{-1}, \bar{s}, \ell) + \big(0,\ b - \gamma^{-1}(\mathcal{F}_1(-b, r, \gamma^{-1}, \bar{s}, \ell) - r)\big). \tag{3.3.8}$$

Then, $\mathcal{B}$ is a $\mathcal{B}$–procedure (see Definition 3.2.1*) for approximately solving* (3.2.5) *in which $s := x$, $B := \partial f$, $\gamma = c^{-1}$, $r := z$ and $b := -p$.*

Proof. Assume that $(s^\ell, b^\ell) = \mathcal{B}(r, b, \gamma, \bar{s}, \bar{b}, \ell)$ for some $r, b, \bar{s}, \bar{b} \in \mathbb{R}^n$, $\gamma > 0$ and all $\ell \in \mathbb{N}^*$. In view of (3.3.8) and the fact that $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2)$ we have

$$(s^\ell, b^\ell) = \big(\mathcal{F}_1(-b, r, \gamma^{-1}, \bar{s}, \ell), \mathcal{F}_2(-b, r, \gamma^{-1}, \bar{s}, \ell)\big) + \big(0,\ b - \gamma^{-1}(\mathcal{F}_1(-b, r, \gamma^{-1}, \bar{s}, \ell) - r)\big)$$

and so, for all $\ell \in \mathbb{N}^*$,

$$(s^\ell, b^\ell - b + \gamma^{-1}(s^\ell - r)) = \big(\mathcal{F}_1(-b, r, \gamma^{-1}, \bar{s}, \ell), \mathcal{F}_2(-b, r, \gamma^{-1}, \bar{s}, \ell)\big) = \mathcal{F}(-b, r, \gamma^{-1}, \bar{s}, \ell).$$

Using the latter identity and the fact that $\mathcal{F}(\cdot)$ is a $\mathcal{F}$–procedure (see Definition 3.3.1) we obtain

$$\lim_{\ell\to\infty} (\underbrace{b^\ell - b + \gamma^{-1}(s^\ell - r)}_{=:y^\ell}) = 0 \ \text{ and } \ (\forall \ell \in \mathbb{N}) \ \ y^\ell \in \partial_x \left[f(x) - \langle b, x\rangle + \frac{1}{2\gamma}\|x - r\|^2 \right]_{x=s^\ell}$$

which, in particular, after some computations, yields $(s^\ell, b^\ell) \in G(\partial f)$, i.e., $b^\ell \in \partial f(s^\ell)$ for all $\ell \in \mathbb{N}^*$. Using this fact and the definition of y^ℓ we find $s^\ell = (\gamma\partial f + I)^{-1}(r + \gamma(y^\ell + b))$, which in turn combined with the fact that $\lim_{\ell\to\infty} y^\ell = 0$ and the continuity of $J_{\gamma\partial f} := (\gamma\partial f + I)^{-1}$ implies that $s^\ell \to J_{\gamma\partial f}(r + \gamma b)$. On the other hand, using the definition of y^ℓ (again) we also obtain $\gamma b^\ell + s^\ell = \gamma(y^\ell + b) + r$, which gives that $\{b^\ell\}$ is convergent and $\gamma b^\ell + s^\ell \to r + \gamma b$. Altogether, we proved that $(s^\ell, b^\ell) \in \partial f$, for all $\ell \in \mathbb{N}^*$, that the sequence $\{(s^\ell, b^\ell)\}$ is convergent and $s^\ell + \gamma b^\ell \to r + \gamma b$, which finishes the proof. □

Algorithm 8. Partially inexact relative-error inertial-relaxed ADMM for (3.3.1)

Choose $c > 0$, $0 \leq \alpha, \sigma < 1$ and $0 < \underline{\rho} < \overline{\rho} < 2$.
Initialize $(x^0, z^0, p^0) = (x^{-1}, z^{-1}, p^{-1}) \in (\mathbb{R}^n)^3$.

for $k = 0, 1, 2, \dots$ **do**

Choose $\alpha_k \in [0, \alpha]$ and define

$$(\hat{x}^k, \hat{z}^k, \hat{p}^k) = (x^k, z^k, p^k) + \alpha_k[(x^k, z^k, p^k) - (x^{k-1}, z^{k-1}, p^{k-1})] \tag{3.3.9}$$

repeat {**for** $\ell = 1, 2, \dots$}

Improve the solution

$$x^{k+1} \approx \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ f(x) + \langle \hat{p}^k, x \rangle + \frac{c}{2} \|x - \hat{z}^k\|^2 \right\} \tag{3.3.10}$$

by setting

$$(x^{k,\ell}, y^{k,\ell}) = \mathcal{F}(\hat{p}^k, \hat{z}^k, c, \hat{x}^k, \ell) \tag{3.3.11}$$

(thus incrementally executing a step of the $\mathcal{F}$–procedure)
Define

$$p^{k,\ell} = \hat{p}^k + c(x^{k,\ell} - \hat{z}^k) - y^{k,\ell} \tag{3.3.12}$$

Exactly find

$$z^{k,\ell} = \operatorname{argmin}_{z \in \mathbb{R}^n} \left\{ g(z) - \langle p^{k,\ell}, z \rangle + \frac{c}{2} \|x^{k,\ell} - z\|^2 \right\} \tag{3.3.13}$$

until

$$\|y^{k,\ell}\|^2 \leq \sigma^2 \left(\|p^{k,\ell} - \hat{p}^k - c(z^{k,\ell} - \hat{z}^k)\|^2 + c^2 \|x^{k,\ell} - z^{k,\ell}\|^2 \right) \tag{3.3.14}$$

if $x^{k,\ell} = z^{k,\ell}$ then **stop**

otherwise, choose $\rho_k \in [\,\underline{\rho}, \overline{\rho}\,]$ and set

$$x^{k+1} = x^{k,\ell}, \quad z^{k+1} = z^{k,\ell} \tag{3.3.15}$$

$$\theta_{k+1} = \frac{\langle c(\hat{z}^k - z^{k,\ell}) - (\hat{p}^k - p^{k,\ell}), x^{k,\ell} - z^{k,\ell} \rangle}{c\|x^{k,\ell} - z^{k,\ell}\|^2} \tag{3.3.16}$$

$$p^{k+1} = \hat{p}^k + c\left[(1 - \rho_k\, \theta_{k+1}) z^{k+1} + \rho_k\, \theta_{k+1} x^{k+1} - \hat{z}^k\right] \tag{3.3.17}$$

end for

Our inertial-relaxed inexact ADMM for solving (3.3.1) is presented as Algorithm 8. Before establishing its convergence, we make the following remarks regarding this algorithm:

(i) Similarly to Algorithm 7, Algorithm 8 benefits from inertial and relaxation effects — see (3.3.9) and (3.3.17) — as well as from the relative error criterion (3.3.14) allowing inexact solution of the f-subproblem (3.3.10).

(ii) Algorithm 8 can be viewed as an inertial-relaxed version of Algorithm 4 in [40], but we emphasize that even without inertia or relaxation (that is, when $\alpha = 0$ and $\rho_k \equiv 1$) it differs from the latter algorithm since Algorithm 4 is based on an approximate proximal point algorithm using an extragradient "corrector" step, while Algorithm 8 is instead based indirectly on Algorithm 6, an approximate proximal point method using projective corrector steps. In developing Algorithm 8, we also experimented with using extragradient correction, but obtained better numerical performance from projective correction.

(iii) The derivation of Algorithm 8 mirrors that in [40], except that the underlying convergence "engine" from [77] is replaced by Algorithm 6. It should be noted that [39] provides a different way of deriving approximate ADMM algorithms. This approach results in different approximate forms of the ADMM, allowing for both relative and absolute error criteria, both of a practically verifiable form. It is also possible that the work in [82] could lead to still more approximate forms of the ADMM.

(iv) As in [40], the derivation of our algorithm is based on a primal reformulation (3.3.5) of the optimization problem (3.3.1) as a monotone inclusion. Using a primal formulation is necessary here, as in [40], because Algorithm 7 requires pairs $(s^{k,\ell}, b^{k,\ell})$ that are in the graph of B. Working with the dual inclusion $0 \in \partial f^*(-p) + g^*(p)$ would in general require exact optimization of linear or quadratic perturbations of f and g, and would thus not lead to a method in which (3.3.2) is solved approximately in a practical manner. In the case of problem (3.3.1), applying exact Douglas-Rachford splitting to either the primal inclusion (3.3.5) or the dual inclusion $0 \in \partial f^*(-p) + g^*(p)$ is known to yield the same ADMM algorithm (3.3.2)-(3.3.4); see for example [36, Proposition 3.43]. Here, we select the primal approach since it leads to a tractable form of approximation for (3.3.2).

The drawback of the primal approach is that does not readily adapt to more general problem formulations such as $\min_{x\in\mathbb{R}^n}\{f(x)+g(Mx)\}$ (where M is an $m \times n$ matrix and g is now defined over $\mathbb{R}^m$ instead of $\mathbb{R}^n$) or the linearly constrained formulation used in [17]. Such formulations require different techniques, such as those employing primal-dual inclusion formulations as in [39].

Proposition 3.3.3. *For any given execution of Algorithm 8, define*

$$(s^k, b^k, r^k) := (x^k, -p^k, z^k), \tag{3.3.18}$$

$$(\hat{s^k}, \hat{b^k}, \hat{r^k}) := (\hat{x}^k, -\hat{p}^k, \hat{z}^k), \tag{3.3.19}$$

$$(s^{k,\ell}, b^{k,\ell}, r^{k,\ell}) := (x^{k,\ell}, -p^{k,\ell}, z^{k,\ell}), \tag{3.3.20}$$

$$a^{k,\ell} := c(s^{k,\ell} - r^{k,\ell}) - b^{k,\ell}, \tag{3.3.21}$$

for all applicable k and l. Then these sequences conform to the recursions (3.2.6)-(3.2.13) *in Algorithm 7 with $\gamma = 1/c$, the $\mathcal{B}$-procedure* (3.3.8), *and the maximal monotone operators $A = \partial g$ and $B = \partial f$.*

Proof. In view of (3.3.18), (3.3.19) and (3.3.9) we have

$$\begin{aligned}(\hat{s}^k, \hat{b}^k, \hat{r}^k) = (\hat{x}^k, -\hat{p}^k, \hat{z}^k) &= \left(x^k + \alpha_k(x^k - x^{k-1}), -p^k - \alpha_k(p^k - p^{k-1}), z^k + \alpha_k(z^k - z^{k-1})\right) \\ &= (s^k, b^k, r^k) + \alpha_k \left[(s^k, b^k, r^k) - (s^{k-1}, b^{k-1}, r^{k-1})\right],\end{aligned}$$

which is identical to (3.2.6) in Algorithm 7. Fix $\gamma = 1/c$. Then (3.3.11), Definition 3.3.1, (3.3.19) lead to

$$x^{k,\ell} = \mathcal{F}_1(\hat{p}^k, \hat{z}^k, c, \hat{x}^k, \ell) = \mathcal{F}_1(-\hat{b}^k, \hat{r}^k, \gamma^{-1}, \hat{s}^k, \ell). \tag{3.3.22}$$

Combining (3.3.20), (3.3.12), (3.3.11), (3.3.22), (3.3.19), and (3.3.8), we deduce that

$$\begin{aligned}(s^{k,\ell}, b^{k,\ell}) &= (x^{k,\ell}, -p^{k,\ell}) \\ &= (x^{k,\ell}, y^{k,\ell}) + (0, -\hat{p}^k - \gamma^{-1}(x^{k,\ell} - \hat{z}^k)) \\ &= \mathcal{F}(-\hat{b}^k, \hat{r}^k, \gamma^{-1}, \hat{s}^k, \ell) + (0, \hat{b}^k - \gamma^{-1}(\mathcal{F}_1(-\hat{b}^k, \hat{r}^k, \gamma^{-1}, \hat{s}^k, \ell) - \hat{r}^k)) \\ &= \mathcal{B}(\hat{r}^k, \hat{b}^k, \gamma, \hat{s}^k, \hat{b}^k, \ell),\end{aligned}$$

which yields (3.2.7) and (3.2.8). Note now that (3.3.13) is equivalent to the condition $0 \in \partial g(z^{k,\ell}) - p^{k,\ell} + c(z^{k,\ell} - x^{k,\ell})$, which, in view of (3.3.20) and (3.3.21), is clearly equivalent to (3.2.9) with $A = \partial g$. To prove (3.2.10), note that from (3.3.18), (3.3.20), (3.3.12) and (3.3.14) we obtain

$$\begin{aligned}\|s^{k,\ell} + \gamma b^{k,\ell} - (\hat{r}^k + \gamma\hat{b}^k)\|^2 &= \|\gamma\, y^{k,\ell}\|^2 \\ &\le \gamma^2\sigma^2 \left(\|p^{k,\ell} - \hat{p}^k - c(z^{k,\ell} - \hat{z}^k)\|^2 + c^2\|x^{k,\ell} - z^{k,\ell}\|^2\right)\end{aligned}$$

which in view of (3.3.19) and (3.3.20) is equivalent to (3.2.10). Finally, similar reasoning establishes that (3.2.11)-(3.2.13) are equivalent to (3.3.15)-(3.3.17). □

Theorem 3.3.4 (Convergence of Algorithm 8). *Consider any execution of Algorithm 8 for which $\alpha \in [0, 1)$, $\overline{\rho} \in (0, 2)$, and $\{\alpha_k\}$ satisfy conditions* (3.1.20) *and* (3.1.21) *of Theorem 3.1.5. Then:*

(a) *If for each $k \ge 0$ the outer loop (over k) executes an infinite number of times, with each inner loop (over ℓ) terminating in a finite number of iterations $\ell = \ell(k)$, then $\{x^k\}$ and $\{z^k\}$ both converge to some $x^* \in \mathbb{R}^n$ solution of* (3.3.5), *and $\{p^k\}$ converges to some $p^* \in \partial g(x^*)$ such that $-p^* \in \partial f(x^*)$.*

(b) *If the outer loop executes only a finite number of times, ending with $k = \bar{k}$, with the last invocation of the inner loop executing an infinite number of times, then $\{x^{\bar{k},\ell}\}_\ell$ and $\{z^{\bar{k},\ell}\}_\ell$ both converge to some $x^* \in \mathbb{R}^n$ solution of* (3.3.5), *and $\{p^{\bar{k},\ell}\}_\ell$ converges to some $p^* \in \partial g(x^*)$ such that $-p^* \in \partial f(x^*)$.*

(c) *If* Algorithm 8 *stops with either $p^{k,\ell} - \hat{p}^k = c(z^{k,\ell} - \hat{z}^k)$ or $x^{k,\ell} = z^{k,\ell}$ then $x^* := x^{k,\ell} = z^{k,\ell}$ is a solution of* (3.3.5).

Proof. The result follows from immediately by combining Proposition 3.3.3, Theorem 3.2.3, and the definitions of Algorithms 7 and 8. □

3.4 Numerical experiments

This section describes numerical experiments on the LASSO and logistic regression problems, which are both instances of the minimization problem (3.3.1). We tested the following algorithms: the inexact relative-error ADMM *admm_primDR* from [40]; the relative-error method *relerr* from [39]; Algorithm 8 from this thesis, which we denote as *admm_primDR_relx_in*; the absolute-error aproximate ADMM *absgeom* discussed in [39, 40], and (for logistic regression problems only) a backtracking variant of *FISTA* [12] (also discussed in [40]). We implemented all algorithms in MATLAB, and, analogously to [40], we used the following condition to terminate the outer loop:

$$\operatorname{dist}_\infty\left(0, \partial_x[f(x)+g(x)]_{x=x^k}\right) \leq \epsilon, \tag{3.4.1}$$

where $\operatorname{dist}_\infty(t,S) := \inf\{\|t-s\|_\infty \mid s \in S\}$, and $\epsilon > 0$ is a tolerance parameter set to 10^{-6}.

Moreover, in our implementation of Algorithm 8 from this thesis, we replaced the error condition (3.3.14) with the stronger condition

$$\|y^{k,\ell}\| \leq \sigma \max\left\{\|p^{k,\ell} - \hat{p}^k - c(z^{k,\ell} - \hat{z}^k)\|, c\|x^{k,\ell} - z^{k,\ell}\|\right\}, \tag{3.4.2}$$

which we empirically found to yield better numerical performance.

3.4.1 Numerical experiments on the LASSO problem

In this subsection, we report numerical experiments on the LASSO or “compressed sensing” problem which readily fits the form (18) [42]. The LASSO is an l_1 regularized version of linear regression written as [83]

$$\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax-b\|_2^2 + \nu\|x\|_1, \tag{3.4.3}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\nu > 0$, which is an instance of (3.3.1) with $f(x) := (1/2)\|Ax-b\|_2^2$ and $g(x) := \nu\|x\|_1$.

For the data A and b, we used four categories of (non-artificial) datasets, as in [40]:

> *Gene expression*: This category consists of six standard cancer DNA microarray datasets from [33]. These instances have dense, wide, and relatively small matrices A, with the number of rows $m \in [42, 102]$, and the number of columns $n \in [2000, 6033]$. These problems are called: brain (with $m = 42$ and $n = 5597$), colon (with $m = 62$ and $n = 2000$), leukemia (with $m = 72$ and $n = 3571$), lymphoma (with $m = 62$ and $n = 4026$), prostate (with $m = 102$ and $n = 6033$) and srbct (with $m = 63$ and $n = 2308$).
>
> *Single-Pixel camera*: This category consists of four compressed image sensing datasets from [34]. These instances have dense, wide, and relatively small matrices A, with $m \in [410, 4770]$ and $n \in [1024, 16384]$. These problems are called: Ball64_singlepixcam (with $m = 1638$ and $n = 4096$), Logo64_singlepixcam (with $m = 1638$ and $n = 4096$), Mug32_singlepixcam (with $m = 410$ and $n = 1024$) and Mug128_singlepixcam (with $m = 410$ and $n = 1024$).

Finance: This category consists in a single large dense financial dataset from [54], with $m = 30465$ and $n = 216842$.

PEMS: This category consists in a single large, wide, and dense dataset from the California Department of Transportation from [55], with $m = 267$ and $n = 138672$.

We tested three algorithms for solving (3.4.3):

- The inexact relative-error ADMM *admm_primDR* from [40]. For this algorithm, we used the same parameter values as in [40], namely $\sigma = 0.99$ and $c = 1$ (except for the PEMS problem instance, for which $c = 20$).

- The relative-error algorithm *relerr* from [39]. We also used $\sigma = 0.99$, $c = 1$ (for all problem instances except PEMS, which we used $c = 20$). For this set of LASSO problems, the experiments in [39, 40] already show *admm_primDR* to outperform the algorithms of [39], as well as FISTA [12].

- Algorithm 8 from this thesis which we denote as *admm_primDR_relx_in*. We used the parameter settings $\alpha_k \equiv \alpha = 0.18966$, $\beta = 0.18976$ and $\rho_k \equiv \underline{\rho} = \overline{\rho} = 1.4882$ — see conditions (3.1.20) and (3.1.21) and Figure 3.2. We also set $\sigma = 0.99$ and $c = 1$ (except for the PEMS problem instance, for which $c = 20$).

We implemented all of the algorithms in MATLAB, using a conjugate gradient procedure to approximately solve the subproblems corresponding to $f(x) = (1/2)\|Ax - b\|^2$, exactly as in [40]. As in [17], we set the regulation parameter ν as $0.1\|A^T b\|_\infty$ and scaled the vector b and the columns of matrix A to have unit l_2 norm.

Table 3.1 shows the number of outer iterations required by each algorithm on each problem instance and the geometric mean taken over the set of test problems. Table 3.2 shows the cumulative total number of inner iterations required by each algorithm on each problem (conjugate gradient) and again the geometric mean taken over the set of test problems. Table 3.3 shows runtimes in seconds demanded by each algorithm to terminate each problem, with the geometric mean taken over the set of test problems. Figure 3.3 shows the same results graphically. In each table, the smallest value in each row appears in bold. In terms of runtime, the new algorithm outperforms that of [40] for all problem except the *finance1000* instance.

3.4.2 Numerical experiments on logistic regression problems

This section describes numerical experiments on the ℓ_1–regularized logistic regression problem [44, 72]

$$\min_{(w,v)\in\mathbb{R}^{n-1}\times\mathbb{R}} \sum_{i=1}^{q} \log\left(1 + \exp(-b_i(a_i^T w + v))\right) + \nu\|w\|_1, \tag{3.4.4}$$

using a training dataset consisting of q pairs (a_i, b_i), where $a_i \in \mathbb{R}^{n-1}$ is a feature vector, $b_i \in \{-1, +1\}$ is the corresponding label, $w \in \mathbb{R}^{n-1}$ represents a weighting of the feature

Table 3.1: LASSO outer iterations; $\alpha = 0.18966$, $\beta = 0.18976$ and $\bar{\rho} = 1.4882$

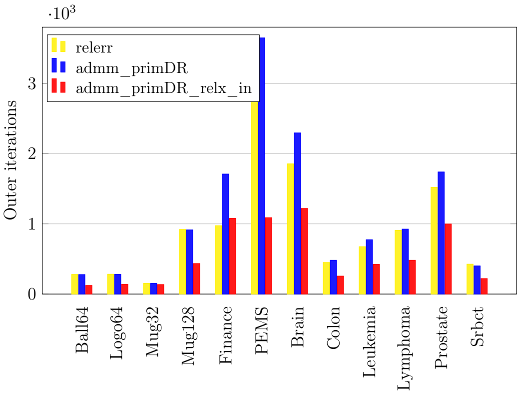
Problem	relerr $(iteration1)$	admm_primDR $(iteration2)$	admm_primDR_relx_in $(iteration3)$	$\frac{iteration3}{iteration1}$	$\frac{iteration3}{iteration2}$
Ball64_singlepixcam	280	278	**123**	0.439	0.442
Logo64_singlepixcam	283	282	**139**	0.491	0.493
Mug32_singlepixcam	153	153	**136**	0.888	0.888
Mug128_singlepixcam	920	914	**435**	0.473	0.476
finance1000	**974**	1709	1079	1.107	0.631
PEMS	3354	3648	**1088**	0.324	0.298
Brain	1855	2295	**1219**	0.657	0.531
Colon	450	482	**256**	0.568	0.531
Leukemia	675	774	**424**	0.628	0.547
Lymphoma	908	925	**482**	0.531	0.521
Prostate	1520	1739	**998**	0.656	0.574
srbct	426	401	**221**	0.519	0.551
Geometric mean	692.06	761.02	**399.85**	0.577	0.525

Table 3.2: LASSO total inner iterations; $\alpha = 0.18966$, $\beta = 0.18976$ and $\bar{\rho} = 1.4882$

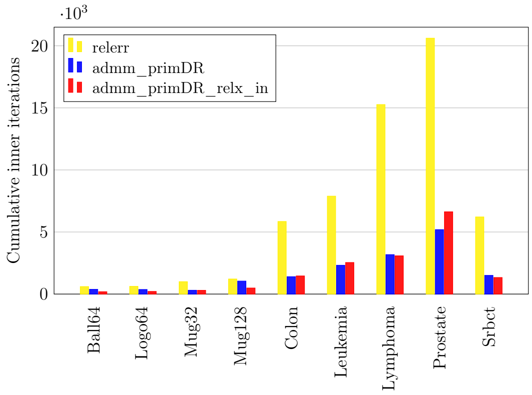
Problem	relerr $(iteration1)$	admm_primDR $(iteration2)$	admm_primDR_relx_in $(iteration3)$	$\frac{iteration3}{iteration1}$	$\frac{iteration3}{iteration2}$
Ball64_singlepixcam	603	382	**191**	0.316	0.500
Logo64_singlepixcam	621	369	**212**	0.341	0.574
Mug32_singlepixcam	998	307	**302**	0.303	0.984
Mug128_singlepixcam	1214	1046	**488**	0.402	0.466
finance1000	18944	**7852**	9737	0.514	1.240
PEMS	85858	9318	**9235**	0.107	0.991
Brain	24612	**7116**	7655	0.311	1.075
Colon	5847	**1401**	1461	0.249	1.042
Leukemia	7888	**2321**	2543	0.322	1.095
Lymphoma	15266	3179	**3083**	0.202	0.969
Prostate	20615	**5193**	6629	0.321	1.276
srbct	6213	1505	**1334**	0.215	0.886
Geometric mean	5859.43	1876.32	**1652.97**	0.282	0.880

Figure 3.3: Comparison of performance in LASSO problems

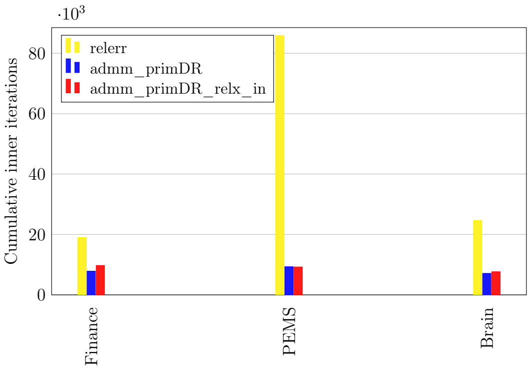
(a) LASSO outer iterations



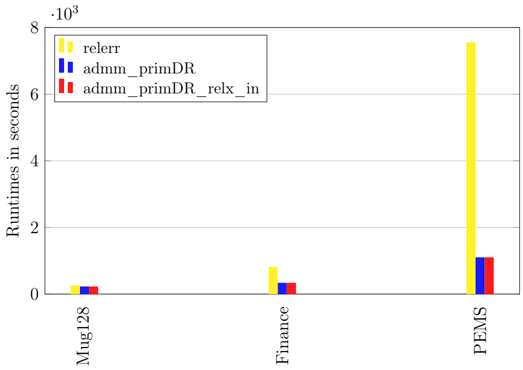
(b) LASSO total inner iterations



(c) LASSO total inner iterations



(d) LASSO runtimes in seconds



(e) LASSO runtimes in seconds

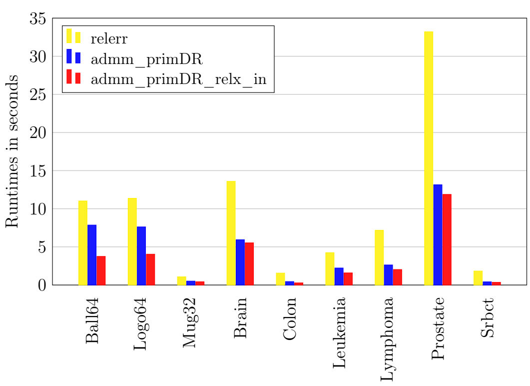


Table 3.3: LASSO runtimes in seconds; $\alpha = 0.18966$, $\beta = 0.18976$ and $\bar{\rho} = 1.4882$

Problem	relerr ($time1$)	admm_primDR ($time2$)	admm_primDR_relx_in ($time3$)	$\frac{time3}{time1}$	$\frac{time3}{time2}$
Ball64_singlepixcam	11.02	7.86	**3.75**	0.341	0.477
Logo64_singlepixcam	11.37	7.62	**4.04**	0.355	0.531
Mug32_singlepixcam	1.07	0.51	**0.43**	0.374	0.862
Mug128_singlepixcam	248.38	218.08	**101.17**	0.407	0.464
finance1000	805.17	**327.56**	347.97	0.432	1.062
PEMS	7546.11	1092.16	**988.12**	0.131	0.905
Brain	13.59	5.94	**5.53**	0.407	0.929
Colon	1.56	0.45	**0.28**	0.179	0.620
Leukemia	4.24	2.23	**1.59**	0.375	0.717
Lymphoma	7.18	2.63	**2.03**	0.283	0.773
Prostate	33.21	13.15	**11.88**	0.357	0.904
srbct	1.83	0.42	**0.35**	0.192	0.847
Geometric mean	21.13	8.75	**6.41**	0.303	0.733

and $v \in \mathbb{R}$ reresents a kind of bias. Problem (3.4.4) is clearly a special instance of (3.3.1) with $x = (v, w)$ and

$$f(v, w) := \sum_{i=1}^{q} \log\left(1 + \exp(-b_i(a_i^T w + v))\right) \quad \text{and} \quad g(v, w) := \nu\|w\|_1. \tag{3.4.5}$$

For test data, we selected three standard cancer DNA microarray non-artificial datasets (*Gene expression*, described in the previous subsection) from [33] (also used in [40, Subsection 7.2]), that have $b_i \in \{-1, 1\}$ for all i. In addition, we also Arcene [50] datasets, from the UCI Machine Learning Repository. This dataset is sparse and has $m = 900$ and $n = 10000$. We tested five algorithms: *absgeom*, *relerr*, *admm_primDR*, *FISTA* and *admm_primDR_relx_in*. For *relerr* and *admm_primDR* algorithms we used the same parameter values as in Subsection 3.4.1; for *admm_primDR_relx_in* we used the parameter settings $\alpha_k \equiv \alpha = 0.1$, $\beta = 0.1001$ and $\rho_k \equiv \underline{\rho} = \overline{\rho} = 1.7606$ — see conditions (3.1.20) and (3.1.21) and Figure 3.2. We also set $\sigma = 0.99$ and $c = 1$.

Analogously to [40], we used an L-BFGS procedure to approximately solve the subproblems corresponding to $f(\cdot)$ from (3.4.5). As in [17], we set the regulation parameter ν as $0.1\|A^T b\|_\infty$ and scaled the vector b and the columns of matrix A to have unit l_2 norm. Table 3.4 shows the number of outer iterations required by each algorithm on each problem and the geometric mean taken over the set of test problems. Note that the relative-error methods take similar numbers of outer iterations, while the absolute-error method requires more outer iterations. Table 3.5 shows the cumulative total number of inner iterations required by each algorithm on each problem and again the geometric mean taken over the set of test problems. The next to last column of Table 3.5 shows the total number of iterations of the FISTA algorithm when run with the same termination accuracy as the ADMM methods. Table 3.6 shows runtimes in seconds demanded by each algorithm to terminate each problem, with the geometric mean taken over the set of test problems. These results are also graphically summarized in Figure 3.4. The new algorithm has the best aggregate performance by all measures, and the best run time for all the datasets.

Table 3.4: Outer iterations for logistic regression problems.

Problem	absgeom $(iteration1)$	relerr $(iteration2)$	admm_primDR $(iteration3)$	admm_primDR_relx_in $(iteration4)$
Colon	2666	2145	1979	**1578**
Leukemia	1662	1116	922	**788**
Prostate	1936	1583	1677	**1198**
Arcene	419	**276**	359	290
Geometric mean	1376.91	1011.28	1023.76	**810.72**
Problem	$\frac{iteration4}{iteration1}$	$\frac{iteration4}{iteration2}$	$\frac{iteration4}{iteration3}$	$\frac{iteration2}{iteration3}$
Colon	0.5919	0.7356	0.7974	1.0839
Leukemia	0.4741	0.7061	0.8546	1.2104
Prostate	0.6188	0.7568	0.7144	0.9439
Arcene	0.6921	1.0507	0.8078	0.7688
Geometric mean	0.5887	0.8016	0.7924	0.9849

Table 3.5: Total inner iterations for logistic regression problems.

Problem	absgeom $(iteration1)$	relerr $(iteration2)$	admm_primDR $(iteration3)$	FISTA $(iteration4)$	admm_primDR_relx_in $(iteration5)$
Colon	20612	23919	21697	26247	**8283**
Leukemia	7715	12086	11625	6536	**4448**
Prostate	18901	27505	24548	13730	**10997**
Arcene	**780**	3236	3589	4648	1450
Geometric mean	6958.73	12665.18	12209.43	10228.97	**4923.21**
Problem	$\frac{iteration5}{iteration1}$	$\frac{iteration5}{iteration2}$	$\frac{iteration5}{iteration3}$	$\frac{iteration5}{iteration4}$	$\frac{iteration4}{iteration1}$
Colon	0.4018	0.3463	0.3817	0.3156	0.9499
Leukemia	0.5765	0.3681	0.3826	0.6805	0.6636
Prostate	0.5818	0.3998	0.4479	0.8009	0.7699
Arcene	1.8589	0.4481	0.4041	0.3119	0.2173
Geometric mean	0.7074	0.4032	0.4032	0.4813	0.5699

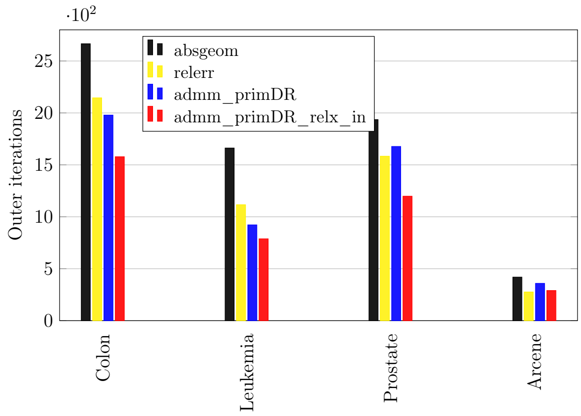
Table 3.6: Logistic regression runtimes in seconds.

Problem	absgeom ($time1$)	relerr ($time2$)	admm_primDR ($time3$)	FISTA ($time4$)	admm_primDR_relx_in ($time5$)
Colon	182.3601	36.5207	91.5726	73.2987	**12.8243**
Leukemia	112.7412	105.4221	241.1378	60.9476	**23.0547**
Prostate	342.1609	719.6731	850.8159	206.3883	**128.6972**
Arcene	122.7208	312.1101	370.9415	184.3489	**46.1276**
Geometric mean	171.41	224.11	288.93	114.18	**36.39**

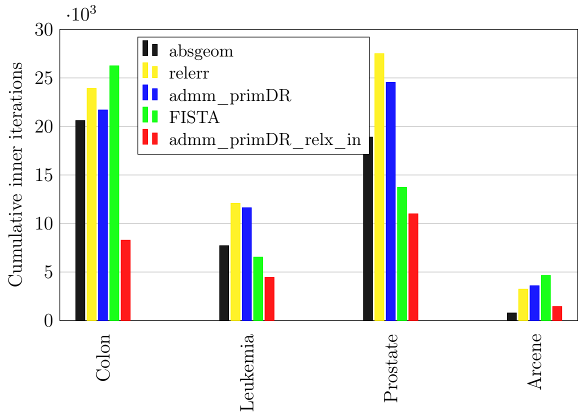
Problem	$\frac{time5}{time1}$	$\frac{time5}{time2}$	$\frac{time5}{time3}$	$\frac{time5}{time4}$	$\frac{time4}{time3}$
Colon	0.0703	0.1203	0.1401	0.1749	0.8003
Leukemia	0.2045	0.2186	0.0956	1.0215	0.2527
Prostate	0.3761	0.1788	0.1513	0.6236	0.2426
Arcene	0.3759	0.1478	0.1244	0.2502	0.4969
Geometric mean	0.2123	0.1623	0.1259	0.3187	0.3951

Figure 3.4: Comparison of performance in logistic regression problems

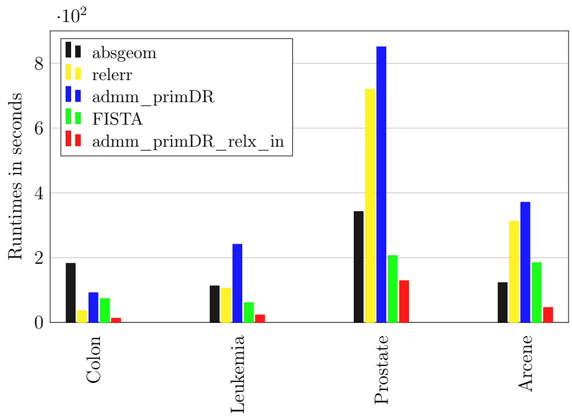
(a) Outer iterations



(b) Total inner iterations



(c) Runtimes in seconds



Chapter 4

Final Remarks

In this thesis, we proposed and analyzed some variants of the Douglas-Rachford method for solving monotone inclusions and of the alternating direction method of multipliers for convex optimization. Initially, we proposed and studied the iteration complexity of an inexact Douglas-Rachford splitting method and a Douglas-Rachford-Tseng's forward-backward splitting method for solving two-operator and four-operator monotone inclusions, respectively. The former method (although based on a slightly different mechanism of iteration) has motivated by the recent work of J. Eckstein and W. Yao, in which an inexact DRS method is derived from a special instance of the hybrid proximal extragradient (HPE) method of Solodov and Svaiter, while the latter one combines the proposed inexact DRS method (used as an outer iteration) with a Tseng's forward-backward splitting type method (used as an inner iteration) for solving the corresponding subproblems. We proved iteration complexity bounds for both algorithms in the pointwise (non-ergodic) as well as in the ergodic sense by showing that they admit two different iterations: one that can be embedded into the HPE method, for which the iteration complexity is known since the work of Monteiro and Svaiter, and another one which demands a separate analysis. Secondly, we studied the asymptotic behavior of new variants of the Douglas-Rachford splitting and ADMM splitting methods, both under relaxation and inertial effects and with inexact (relative-error) criterion for subproblems. Our analysis has essentially based on a new inexact version of the proximal point algorithm, also proposed by this thesis, that includes both an inertial step and overrelaxation. To demonstrate the applicability of the proposed methods, we performed numerical experiments applying the ADMM (relaxed and inertial) on LASSO and logistic regression problems. We obtained present better computational performance than earlier inexact ADMM methods. Moreover, our numerical results indicate that the proposed inexact versions are a useful tool for solving some real-life applications that can be posed in the general framework of convex optimization.

Appendix A

Auxiliary Results

Lemma A.1. ([6, Lemma 3.1]) *Let* $z_\gamma^* := (\gamma B + I)^{-1}(\mathring{z})$ *be the (unique) solution of* (2.2.5)*. Then,*

$$\|\mathring{z} - z_\gamma^*\| \leq \|\mathring{z} - x^*\| \qquad \forall x^* \in B^{-1}(0). \tag{A.1}$$

Lemma A.2. ([81, Lemma 2.2]) *Let* $F : \mathcal{H} \to \mathcal{H}$ *be* η*–cocoercive, for some* $\eta > 0$*, and let* $z', \widetilde{z} \in \mathcal{H}$*. Then,*

$$F(z') \in F^{\varepsilon}(\widetilde{z}) \quad \textit{where} \quad \varepsilon := \frac{\|z' - \widetilde{z}\|^2}{4\eta}.$$

Lemma A.3 (See for example Proposition 20.33 of [11])**.** *If* T *is maximal monotone on* $\mathbb{R}^n$*,* $\{(\tilde{z}^j, v^j)\}$ *is such that* $v^j \in T(\tilde{z}^j)$ *for all* $j \geq 0$*,* $\lim_{j\to\infty} \tilde{z}^j = z^\infty$*, and* $\lim_{j\to\infty} v^j = v^\infty$*, then* $v^\infty \in T(z^\infty)$*.*

Lemma A.4. *The inverse function of the scalar map*

$$(0,2) \ni \rho \mapsto \phi(\rho) := \frac{2(2-\rho)}{4 - \rho + \sqrt{16\rho - 7\rho^2}} \in (0,1)$$

is

$$(0,1) \ni \beta \mapsto \psi(\beta) := \frac{2(\beta-1)^2}{2(\beta-1)^2 + 3\beta - 1} \in (0,2).$$

Proof. We first claim that $\psi(\beta) \in [0,2]$ for all $\beta \in [0,1]$ and $\psi(\beta) \in (0,2)$ for all $\beta \in (0,1)$. To establish this claim, we first note that by elementary calculus and some simplifications, we have

$$\tfrac{d}{d\beta}\psi(\beta) = \frac{6\beta^2 - 4\beta - 2}{\big(2(\beta-1)^2 + 3\beta - 1\big)^2} = \frac{6\beta^2 - 4\beta - 2}{\big(2\beta^2 - \beta + 1\big)^2}. \tag{A.2}$$

The discriminant of $2\beta^2 - \beta + 1$ is negative, so it has no real roots and the denominator of (A.2) is always positive. The expression in the numerator is convex and applying the quadratic formula yields that that its roots are $-1/3$ and 1, so therefore it is nonpositive on $[0,1]$ and negative on $(0,1)$. Therefore, $\frac{d}{d\beta}\psi(\beta)$ exists for all $\beta \in [0,1]$ and is negative for all $\beta \in (0,1)$, implying that ψ is a decreasing function on $(0,1)$. By direct calculation, $\psi(0) = 2$ and $\psi(1) = 0$, so therefore $\big\{\psi(\beta) \mid \beta \in [0,1]\big\} = [0,2]$ and

$\{\psi(\beta) \mid \beta \in (0,1)]\} = (0,2)$, establishing the initial claim. To continue the proof, we next establish that

$$\phi(\psi(\beta)) = \beta \qquad \forall \beta \in (0,1). \tag{A.3}$$

To this end, fix any $\beta \in (0,1)$ and define

$$(0,2) \ni \rho := \psi(\beta) = \frac{2(\beta-1)^2}{2(\beta-1)^2 + 3\beta - 1} = \frac{2\beta^2 - 4\beta + 2}{2\beta^2 - \beta + 1},$$

which implies the quadratic equation

$$2(1-\rho)\beta^2 - (4-\rho)\beta + (2-\rho) = 0. \tag{A.4}$$

We now consider three cases in (A.4): $\rho = 1$, $\rho < 1$, and $\rho > 1$.

$\rho = 1$: in this case, simplification of (A.4) and the definition of ϕ yield that $\beta = 1/3 = \phi(1)$.

$\rho < 1$: the unique minimizer of the quadratic function in (A.4) is $\beta^* := (4-\rho)/\big(4(1-\rho)\big)$, which must be greater than 1 because $\rho > 0$. Thus, we have $\beta^* > 1 > \beta > 0$, so β is the smaller root of the quadratic equation in (A.4). Using the quadratic formula and rationalizing the denominator,

$$\beta = \frac{4-\rho-\sqrt{(\rho-4)^2 - 4\cdot 2(1-\rho)(2-\rho)}}{2\cdot 2(1-\rho)} = \frac{4-\rho-\sqrt{16\rho-7\rho^2}}{4(1-\rho)} \tag{A.5}$$

$$= \frac{4-\rho-\sqrt{16\rho-7\rho^2}}{4(1-\rho)} \cdot \frac{4-\rho+\sqrt{16\rho-7\rho^2}}{4-\rho+\sqrt{16\rho-7\rho^2}}$$

$$= \frac{16-24\rho+8\rho^2}{4(1-\rho)\big(4-\rho+\sqrt{16\rho-7\rho^2}\big)} = \frac{8(1-\rho)(2-\rho)}{4(1-\rho)\big(4-\rho+\sqrt{16\rho-7\rho^2}\big)}$$

$$= \frac{2(2-\rho)}{4-\rho+\sqrt{16\rho-7\rho^2}} = \phi(\rho). \tag{A.6}$$

$\rho > 1$: in this case, β^* as defined in the previous case is the unique maximizer of the quadratic function in (A.4) and $\beta^* < 0$. So $\beta^* < 0 < \beta < 1$ and β is the larger root of the quadratic in (A.4). Since the coefficient of the quadratic term is negative in this case, this root also takes the form (A.5), and consequently (A.6) still holds.

The proof of (A.3) is now complete. Finally, we now prove that

$$\psi(\phi(\rho)) = \rho \qquad \forall \rho \in (0,2). \tag{A.7}$$

To this end, let $0 < \rho < 2$ and define

$$(0,1) \ni \beta := \phi(\rho) = \frac{2(2-\rho)}{4-\rho+\sqrt{16\rho-7\rho^2}}.$$

Using the above definition and the quadratic formula, we conclude that β also satisfies the quadratic equation (A.4), which after some simple algebra gives

$$\rho = \frac{2(\beta-1)^2}{2(\beta-1)^2 + 3\beta - 1},$$

that is, $\rho = \psi(\beta)$, which in turn is equivalent to (A.7). □

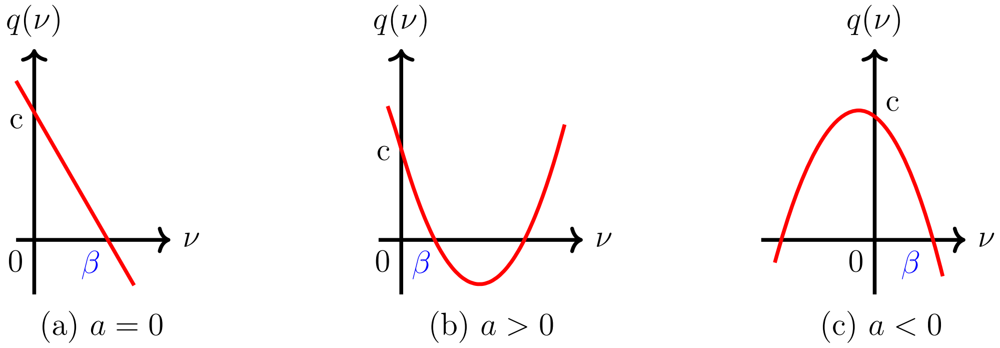


Figure A.1: Possible cases for the quadratic function $q(\cdot)$ in Lemma A.5.

Lemma A.5. *Let $\mathbb{R} \ni \nu \mapsto q(\nu) := a\nu^2 - b\nu + c$ be a real function and assume that $b, c > 0$ and $b^2 - 4ac > 0$. Define*

$$\beta := \frac{2c}{b + \sqrt{b^2 - 4ac}} > 0. \tag{A.8}$$

(i) *If $a = 0$, then $q(\cdot)$ is a decreasing affine function and $\beta > 0$ as in* (A.8) *is its unique root* (*see* Figure A.1(a)).

(ii) *If $a > 0$* (*resp. $a < 0$*), *then $q(\cdot)$ is a convex* (*resp. concave*) *quadratic function and $\beta > 0$ as in* (A.8) *is its smallest* (*resp. largest*) *root* (*see* Figure A.1(b) *and* Figure A.1(c), *resp.*).

In both cases (i) *and* (ii), *$\beta > 0$ as in* (A.8) *is a root of $q(\cdot)$, and $q(\cdot)$ is decreasing in the interval* $[0, \beta]$ (*see* Figure A.1).

Proof. The proof of (i) is straightforward. To prove (ii), note that rationalizing the denominator of (A.8) results in $\beta = \left(b - \sqrt{b^2 - 4ac}\right)/2a$, which in turn implies that (ii) follows from the quadratic formula and the assumption that $b, c > 0$. The last statement of the lemma is a direct consequence of (i), (ii) and the assumption that $b, c > 0$. □

Lemma A.6 (Opial [73])**.** *Let $\emptyset \neq \Omega \subset \mathbb{R}^n$ and $\{z^k\}$ be a sequence in $\mathbb{R}^n$ such that every cluster point of $\{z^k\}$ belongs to Ω and $\lim_{k\to\infty} \|z^k - z^*\|$ exists for every $z^* \in \Omega$. Then $\{z^k\}$ converges to a point in Ω.*

The following lemma was essentially proved by Alvarez and Attouch in [2, Theorem 2.1].

Lemma A.7. *Let the sequences $\{\varphi_k\}$, $\{s_k\}$, $\{\alpha_k\}$ and $\{\delta_k\}$ in $[0, +\infty)$ and $\alpha \in \mathbb{R}$ be such that $\varphi_0 = \varphi_{-1}$, $0 \le \alpha_k \le \alpha < 1$ and*

$$\varphi_{k+1} - \varphi_k + s_{k+1} \le \alpha_k(\varphi_k - \varphi_{k-1}) + \delta_k \qquad \forall k \ge 0. \tag{A.9}$$

The following hold:

(a) *For all $k \ge 1$,*

$$\varphi_k + \sum_{j=1}^{k} s_j \le \varphi_0 + \frac{1}{1-\alpha} \sum_{j=0}^{k-1} \delta_j. \tag{A.10}$$

(b) *If* $\sum_{k=0}^{\infty} \delta_k < +\infty$*, then* $\lim_{k\to\infty} \varphi_k$ *exists,* i.e., *the sequence* $\{\varphi_k\}$ *converges to some element in* $[0, +\infty)$.

Proof. It was proved in [2, Theorem 2.1] that $\mathcal{M} := (1-\alpha)^{-1} \sum_{j=0}^{k} \delta_j \geq \sum_{j=1}^{k+1} [\varphi_j - \varphi_{j-1}]_+$, where $[\cdot]_+ = \max\{\cdot, 0\}$. Using this, the assumptions $\varphi_0 = \varphi_{-1}$, $0 \leq \alpha_k \leq \alpha < 1$ and (A.9), and some algebraic manipulations we find, for all $k \geq 0$,

$$
\begin{aligned}
\varphi_{k+1} + \sum_{j=1}^{k+1} s_j &\leq \varphi_0 + \alpha \sum_{j=1}^{k+1} [\varphi_j - \varphi_{j-1}]_+ + \sum_{j=0}^{k} \delta_j \\
&\leq \varphi_0 + \alpha\mathcal{M} + (1-\alpha)\mathcal{M} = \varphi_0 + \mathcal{M},
\end{aligned}
$$

which proves (a). To finish the proof, we note that (b) was established within the proof of [2, Theorem 2.1]. □

Bibliography

[1] F. Alvarez. Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space. *SIAM J. Optim.*, 14(3):773–782, 2003.

[2] F. Alvarez and H. Attouch. An inertial proximal method for maximal monotone operators via discretization of a nonlinear oscillator with damping. *Set-Valued Var. Anal.*, 9(1-2):3–11, 2001.

[3] M. Marques Alves, J.Eckstein, M. Geremia and Melo,J.G. Relative-error inertial-relaxed inexact versions of Douglas-Rachford and ADMM splitting algorithms. *Comput. Optim. Appl.*, 75:389–422, 2020.

[4] M. Marques Alves and M. Geremia. Iteration complexity of an inexact Douglas-Rachford method and of a Douglas-Rachford-Tseng's F-B four-operator splitting method for solving monotone inclusions. *Numer. Algorithms*, 82(1):263–295, 2019.

[5] M. Marques Alves and R.T. Marcavillaca. On inexact relative-error hybrid proximal extragradient, forward-backward and Tseng's modified forward-backward methods with inertial effects. *Set-Valued Var. Anal.*, 28:301—325, 2020.

[6] M. Marques Alves, R. D. C. Monteiro, and B. F. Svaiter. Regularized HPE-type methods for solving monotone inclusions with improved pointwise iteration-complexity bounds. *SIAM J. Optim.*, 26(4):2730–2743, 2016.

[7] H. Attouch and A. Cabot. Convergence of a relaxed inertial forward-backward algorithm for structured monotone inclusions. *Appl Math Optim*, 80:547—598, 2019.

[8] H. Attouch and A. Cabot. Convergence of a relaxed inertial proximal algorithm for maximally monotone operators. *Math. Program.*, 184:243—287, 2020.

[9] H. Attouch, Z. Chbani, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Math. Program.*, 168(1-2, Ser. B):123–175, 2018.

[10] H. Attouch, J. Peypouquet, and P. Redont. Fast convex optimization via inertial dynamics with Hessian driven damping. *J. Differ. Equ.*, 261(10):5734–5783, 2016.

[11] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

[12] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[13] R. I. Boţ, E. R. Csetnek, A. Heinrich, and C. Hendrich. On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems. *Math. Program.*, 150(2, Ser. A):251–279, 2015.

[14] R. I. Boţ and E. R. Csetnek. A hybrid proximal-extragradient algorithm with inertial effects. *Numer. Funct. Anal. Optim.*, 36(8):951–963, 2015.

[15] R. I. Boţ, E. R. Csetnek, and C. Hendrich. Recent developments on primal-dual splitting methods with applications to convex minimization. In *Mathematics without boundaries*, pages 57–99. Springer, New York, 2014.

[16] R. I. Bot, E. R. Csetnek and C. Hendrich. Inertial Douglas-Rachford splitting for monotone inclusion problems. *Appl. Math. Comput.*, 256:472–487, 2015.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

[18] L. M. Briceño Arias. Forward-Douglas-Rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization*, 64(5):1239–1261, 2015.

[19] L. M. Briceño Arias. Forward-partial inverse-forward splitting for solving monotone inclusions. *J. Optim. Theory Appl.*, 166(2):391–413, 2015.

[20] L. Briceno-Arias and D. Davis. Forward-backward-half forward algorithm with non self-adjoint linear operators for solving monotone inclusions. *SIAM J. Optim.*, 28(4):2839-–2871, 2018.

[21] R. S. Burachik, A. N. Iusem, and B. F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Anal.*, 5(2):159–180, 1997.

[22] R. S. Burachik, C. A. Sagastizábal, and B. F. Svaiter. ϵ-enlargements of maximal monotone operators: theory and applications. In *Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods (Lausanne, 1997)*, volume 22 of *Appl. Optim.*, pages 25–43. Kluwer Acad. Publ., Dordrecht, 1999.

[23] L. C. Ceng, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *J. Optim. Theory Appl.*, 146(2):267–303, 2010.

[24] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, 2011.

[25] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, (2):27:1–27:27, 2011.

[26] C. Chen, R. H. Chan, S. Ma, and J. Yang. Inertial proximal ADMM for linearly constrained separable convex optimization. *SIAM J. Imaging Sci.*, 8(4):2239–2267, 2015.

[27] C. Chen, S. Ma, and J. Yang. A general inertial proximal point algorithm for mixed variational inequality problem. *SIAM J. Optim.*, 25(4):2120–2142, 2015.

[28] P. L. Combettes. Systems of structured monotone inclusions: duality, algorithms, and applications. *SIAM J. Optim.*, 23(4):2420–2447, 2013.

[29] P. L. Combettes and J.-C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Var. Anal.*, 20(2):307–330, 2012.

[30] P. L. Combettes and L. E. Glaudin. Quasi-nonexpansive iterations on the affine hull of orbits: from Mann's mean value algorithm to inertial methods. *SIAM J. Optim.*, 27(4):2356–2380, 2017.

[31] D. Davis. Convergence rate analysis of the forward-Douglas-Rachford splitting scheme. *SIAM J. Optim.*, 25(3):1760–1786, 2015.

[32] D Davis and W. Yin. A three-operator splitting scheme and its optimization applications. *Set-valued and variational analysis*, 2017.

[33] M. Dettling and P. Bühlmann. Finding predictive gene groups from microarray data. *J. Multivariate Anal.*, 90(1):106–131, 2004.

[34] M. F. Duarte, M. A. Davenport, D. Takbar, J. N. Laska, T. Sun, K. F. Kelly and R. G. Baraniuk. Singlepixel imaging via compressive sampling: building simpler, smaller, and less-expensive digital cameras. *IEEE Sig. Process. Mag.*, 25(2):83—91, 2008.

[35] J. Douglas, Jr. and H. H. Rachford, Jr. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[36] J. Eckstein. *Splitting methods for Monotone Operators and Applications to Parallel Optimization*. Doctoral dissertation, Massachusetts Institute of Technology, 1989.

[37] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, 1992.

[38] J. Eckstein and P. J. S. Silva. A practical relative error criterion for augmented Lagrangians. *Math. Program.*, 141(1-2, Ser. A):319–348, 2013.

[39] J. Eckstein and W. Yao. Approximate ADMM algorithms derived from Lagrangian splitting. *Comput. Optim. Appl.*, 68(2):363–405, 2017.

[40] J. Eckstein and W. Yao. Relative-error approximate versions of Douglas-Rachford splitting and special cases of the ADMM. *Math. Program.*, 170(2):417–444, 2018.

[41] H. W. Engl, M. Hanke and A. Neubauer. Regularization of inverse problems. *Springer Netherlands: Mathematics and Its Applications*, 1996.

[42] M. A. T. Figueiredo, R. D. Nowak and S. J. Wright. Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems. *IEEE Journal of Selected Topics in Signal Processing*, 4(1):586–597, 2007.

[43] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and its Applications*, pages 97–146. North-Holland, Amsterdam, 1983.

[44] J. Franklin. The elements of statistical learning: data mining, inference and prediction. *Math. Intelligencer*, 27(2):83–85, 2005.

[45] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.*, 2:17–40, 1976.

[46] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and its Applications*, pages 299–331. North-Holland, Amsterdam, 1983.

[47] R. Glowinski, A. Marroco. *Sur I'approximation, par elements finis d'ordre un, et la resolution, par penalisation-dualité, d'une classe de problems de Dirichlet non lineares*. Revue Française d'Automatique, Infomatique, et Recherche Opérationelle, 9:41–76, 1975.

[48] R. Glowinski, S. J. Osher, and W. Yin, editors. *Splitting methods in communication, imaging, science, and engineering*. Scientific Computation. Springer, Cham, 2016.

[49] M. L. N. Gonçalves, J. G. Melo, and R. D. C. Monteiro. Improved pointwise iteration-complexity of a regularized ADMM and of a regularized non-Euclidean HPE framework. *SIAM J. Optim.*, 27(1):379–407, 2017.

[50] I. Guyon, S. Gunn, A. Ben-Hur and G. Dror. Result analysis of the NIPS 2003 feature selection challenge. *Saul, L., Weiss, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems*, 17:545–552, MIT Press, Cambridge, 2005.

[51] Y. He and R. D. C. Monteiro. An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM J. Optim.*, 26(1):29–56, 2016.

[52] A. N. Iusem and W. Sosa. On the proximal point method for equilibrium problems in Hilbert spaces. *Optimization*, 59(8):1259–1274, 2010.

[53] R. I. Kachurovskii. On monotone operators and convex functionals. *Uspekhi Matemacheskaya Nauk*, 15(4):213–215, 1960.

[54] S. Kogan, D. Levin, B. R. Routledge, J. S. Sagi and N. A. Smith. Predicting risk from financial reports with regression. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. NAACL*, 9:272—280. Association for Computational Linguistics, Stroudsburg, PA, USA, 2009.

[55] M. Lichman. UCI machine learning repository: Center for Machine Learning and Intelligent Systems, 2013. http://archive.ics.uci.edu/ml

[56] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[57] P. A. Lotito, L. A. Parente, and M. V. Solodov. A class of variable metric decomposition methods for monotone variational inclusions. *J. Convex Anal.*, 16(3-4):857–880, 2009.

[58] D. A. Lorenz and T. Pock. An inertial forward-backward algorithm for monotone inclusions. *J. Math. Imaging Vision*, 51(2):311–325, 2015.

[59] P.-E. Maingé and N. Merabet. A new inertial-type hybrid projection-proximal algorithm for monotone inclusions. *Appl. Math. Comput.*, 215(9):3149–3162, 2010.

[60] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Duke Mathematical Journal*, 4(Ser. R-3):341–346, 1962.

[61] J.-E.Martinez-Legaz and M. Théra. ϵ-subdifferentials in terms of subdifferentials. *Set-Valued Anal.*, 4(4):327–332, 1996.

[62] G. J. Minty. Monotone (nonlinear) operators in Hilbert space. *Rev. Française Informat. Recherche Opérationnelle*, 29:154–158, 1970.

[63] R. D. C. Monteiro, C. Ortiz, and B. F. Svaiter. A first-order block-decomposition method for solving two-easy-block structured semidefinite programs. *Math. Program. Comput.*, 6(2):103–150, 2014.

[64] R. D. C. Monteiro, C. Ortiz, and B. F. F. Svaiter. Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems. *Comput. Optim. Appl.*, 57(1):45–69, 2014.

[65] R. D. C. Monteiro, C. Ortiz, and B. F. F. Svaiter. An adaptive accelerated first-order method for convex optimization. *Comput. Optim. Appl.*, 64(1):31–73, 2016.

[66] R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of Tseng's modified F-B splitting and Korpelevich's methods for hemivariational inequalities with applications to saddle point and convex optimization problems. *SIAM Journal on Optimization*, 21:1688–1720, 2010.

[67] R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20:2755–2787, 2010.

[68] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM J. Optim.*, 23(1):475–507, 2013.

[69] R.D.C Monteiro and S. Chee-Khian. Complexity of the relaxed Peaceman-Rachford splitting method for the sum of two maximal strongly monotone operators. *Comput. Optim. Appl.*, 70:763–790, 2018.

[70] J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965.

[71] Y. Nesterov. Introductory lectures on convex optimization. A basic course.Kluwer Academic Publishers, Boston, 2004.

[72] A. Y. Ng. Feature selection, L_1 vs. L_2 regularization, and rotational invariance. In *Proceedings, Twenty-First International Conference on Machine Learning, ICML*, pages 615–622, 2004.

[73] Z. Opial. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bull. Amer. Math. Soc.*, 73:591–597, 1967.

[74] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[75] R. T. Rockafellar. On the maximal monotonicity of subdifferential mappings. *Pacific J. Math*, 33:209–216, 1970.

[76] M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Var. Anal.*, 7(4):323–345, 1999.

[77] M. V. Solodov and B. F. Svaiter. A hybrid projection-proximal point algorithm. *J. Convex Anal.*, 6(1):59–70, 1999.

[78] M. V. Solodov and B. F. Svaiter. An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions. *Math. Oper. Res.*, 25(2):214–230, 2000.

[79] M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numer. Funct. Anal. Optim.*, 22(7-8):1013–1035, 2001.

[80] B. F. Svaiter. On weak convergence of the Douglas-Rachford method. *SIAM J. Control Optim.*, 49(1):280–287, 2011.

[81] B. F. Svaiter. A class of Fejér convergent algorithms, approximate resolvents and the hybrid proximal-extragradient method. *J. Optim. Theory Appl.*, 162(1):133–153, 2014.

[82] B. F. Svaiter. A weakly convergent fully inexact Douglas-Rachford method with relative error tolerance. *ESAIM: COCV*, 25:57–70, 2018.

[83] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[84] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.*, 38(2):431–446 (electronic), 2000.