



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística
e Computação Científica

TIARA MARTINI DOS SANTOS

**UM ESTUDO SOBRE A RESOLUÇÃO DE SISTEMAS
NÃO LINEARES: PERSPECTIVAS TEÓRICAS E APLICAÇÕES**

**CAMPINAS
2016**



TIARA MARTINI DOS SANTOS

UM ESTUDO SOBRE A RESOLUÇÃO DE SISTEMAS
NÃO LINEARES: PERSPECTIVAS TEÓRICAS E APLICAÇÕES

Tese apresentada ao Instituto de Matemática,
Estatística e Computação Científica da Univer-
sidade Estadual de Campinas como parte dos
requisitos exigidos para a obtenção do título de
Doutora em matemática aplicada.

Orientador: Prof. Dr. José Mario Martínez Pérez

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA
TESE DEFENDIDA PELA ALUNA TIARA MARTINI DOS
SANTOS, E ORIENTADA PELO PROF. DR. JOSÉ MARIO
MARTÍNEZ PÉREZ.

CAMPINAS
2016

Agência(s) de fomento e nº(s) de processo(s): FAPESP, 2012/10444-0; CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

Santos, Tiara Martini dos, 1988-
Sa59c Um estudo sobre a resolução de sistemas não lineares : perspectivas teóricas e aplicações / Tiara Martini dos Santos. – Campinas, SP : [s.n.], 2016.

Orientador: José Mario Martínez Pérez.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica.

1. Sistemas não lineares. 2. Programação matemática com restrições de complementaridade. 3. Iterações de ponto fixo. 4. Métodos quase-Newton. 5. Métodos homotópicos. I. Martínez Pérez, José Mario, 1948-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: A study about solving non-linear systems : theoretical perspectives and applications

Palavras-chave em inglês:

Nonlinear systems

Mathematical programming with complementarity constraints

Fixed point iterations

Quasi-Newton methods

Homotopy methods

Área de concentração: Matemática Aplicada

Titulação: Doutora em Matemática Aplicada

Banca examinadora:

José Mario Martínez Pérez [Orientador]

Sandra Augusta Santos

Lúcio Tunes dos Santos

Juliano de Bem Francisco

Luís Felipe Cesar da Rocha Bueno

Data de defesa: 22-02-2016

Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 22 de fevereiro de 2016 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ

Prof(a). Dr(a). SANDRA AUGUSTA SANTOS

Prof(a). Dr(a). LÚCIO TUNES DOS SANTOS

Prof(a). Dr(a). JULIANO DE BEM FRANCISCO

Prof(a). Dr(a). LUÍS FELIPE CESAR DA ROCHA BUENO

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no processo de vida acadêmica do aluno.

À minha família.
À Deus.

Agradecimentos

À Deus acima de tudo.

Aos meus pais, Luiz Gonzaga e Maria Goreti, e ao meu esposo, Bruno, por me incentivarem sempre, por todo carinho e compreensão.

Aos meus amigos e familiares, por toda ajuda, incentivo e pelas conversas descontraídas.

Ao meu orientador, professor Mario Martínez, agradeço pelos ensinamentos transmitidos, por toda paciência, dedicação, confiança e constante incentivo.

Aos professores Roberto Andreani e Joaquim Júdice, coautores de parte deste trabalho, pela colaboração e importantes sugestões.

Aos professores Felipe, Juliano, Lúcio e Sandra por aceitarem compor a banca, dedicarem tempo para a leitura do trabalho e contribuírem para seu aperfeiçoamento.

Também agradeço todos os outros professores que me auxiliaram desde a graduação até o término desta tese.

À FAPESP¹ e à CAPES, pelo auxílio financeiro.

¹Processo FAPESP 2012/10444-0.

Resumo

O objetivo deste trabalho é estudar e analisar diferentes abordagens para resolver sistemas não lineares. Em primeiro lugar, uma versão esparsa do método de Newton é aplicada para encontrar uma solução do problema de complementaridade horizontal não linear (HNCP) associado a uma solução viável do problema de programação matemática com restrições de complementaridade (MPCC). O algoritmo combina direções do tipo Newton e Gradientes Projetados com um procedimento de busca linear que garante convergência global a um ponto estacionário da função de mérito associada a este problema. Convergência local quadrática é estabelecida sob hipóteses razoáveis. Experiência numérica em problemas teste de uma coleção bem conhecida ilustra a eficiência do algoritmo para encontrar soluções viáveis de MPCC na prática. Em seguida, uma estratégia quase-Newton para acelerar a convergência de iterações de ponto fixo é analisada. Para isso, atualizações secantes clássicas são consideradas. Experimentos numéricos em um conjunto treino são desenvolvidos, a fim de validar esta estratégia. Posteriormente, a estratégia quase-Newton é aplicada ao problema prático de representar o comportamento cinético de um marcador PET (Tomografia por Emissão de Pósitrons) durante a perfusão cardíaca. O desempenho do método quando aplicado a problemas com dados reais é ilustrado numericamente. Finalmente, um método híbrido que combina direções de Newton e Homotopia é introduzido para resolver problemas onde o método de Newton apresenta dificuldades. Experimentos iniciais constituem uma base para validação da técnica apresentada.

Palavras-chave: Sistemas não Lineares. Programação Matemática com Restrições de Complementaridade. Iterações de Ponto Fixo. Métodos quase-Newton. Métodos Homotópicos.

Abstract

The aim of this work is to study and analyse different approaches for solving nonlinear systems. First of all, a sparse version of Newton's method is applied for finding a solution of a horizontal nonlinear complementarity problem (HNCP) associated to a feasible solution of a mathematical programming problem with complementarity constraints (MPCC). The algorithm combines Newton-like and Projected-Gradient directions with a line-search procedure that guarantees global convergence to a stationary point of the merit function associated to this problem. Local quadratic convergence is stated under reasonable hypothesis. Numerical experience on test problems from a well-known collection illustrates the efficiency of the algorithm to find feasible solutions of MPCC in practice. Next, a quasi-Newton strategy for accelerating the convergence of fixed-point iterations is analysed. For that, classical secant updates are considered. Numerical experiments on a training set are developed in order to validate this strategy. After that, the quasi-Newton strategy is applied on the practical problem of represent the kinetic behavior of a PET (Positron Emission Tomography) tracer during cardiac perfusion. The performance of the method when applied to real data problems is illustrated numerically. Finally, a hybrid method combining Newton and Homotopy directions is introduced for solving problems where Newton's method presented difficulties. Initial experiments provide a basis for the presented technic validation.

Keywords: Nonlinear Systems. Mathematical Programming with Complementarity Constraints. Fixed Point Iterations. Quasi-Newton Methods. Homotopy Methods.

Sumário

1	Introdução	11
2	O Método de Newton	17
2.1	Introdução	17
2.2	Implementação Computacional	18
2.2.1	Sistemas subdeterminados e sobredeterminados	18
2.2.2	Mau condicionamento ou singularidade	18
2.2.3	Controle de passo	19
2.2.4	Progresso e convergência	19
2.2.5	Esparsidade	20
2.3	Estratégia de pontos interiores para MPCC: Uma aplicação do Algoritmo 2.1	21
2.3.1	Algoritmo PGUN	26
2.3.2	Convergência local	28
2.4	Experimentos computacionais	36
2.4.1	O método de Levenberg-Marquardt Projetado	37
2.4.2	Implementação e experimentos numéricos	38
2.5	Conclusões	47
3	Avaliação da estratégia quase-Newton de aceleração algorítmica	51
3.1	Introdução	51
3.2	Implementação Computacional	53
3.3	Resultados Numéricos e Problemas	61
3.3.1	Experimento 1: problemas treino	62
3.3.2	Experimento 2: aplicação prática	71
3.4	Conclusões	78
4	Aplicação da estratégia quase-Newton de aceleração algorítmica em PET	79
4.1	Introdução	79
4.2	Aplicação da aceleração quase-Newton	80
4.3	Resultados numéricos e Problemas	81
4.4	Conclusões	97
5	Introdução aos métodos homotópicos	99
5.1	Introdução	99
5.2	O Método Homotópico	99
5.2.1	Seguindo a curva γ com Euler	101
5.2.2	Calculando y'	102

5.3	Algoritmo híbrido de Newton com Homotopia (MNH)	104
5.4	Resultados Numéricos e Problemas	105
5.5	Conclusões	107
6	Considerações Finais	109
	Bibliografias	110

Capítulo 1

Introdução

A necessidade de resolver um sistema de equações não lineares emerge naturalmente durante a modelagem de problemas práticos relacionados às mais diferentes áreas. Ortega & Rheinboldt [56], Burden & Faires [10] e Moré [48] listam uma série de aplicações práticas relacionadas com engenharia, física, ciência da computação, biologia, entre outras. Dentre elas destacamos: cálculo de trajetórias, equações de transferência de radiação, dinâmica populacional, problemas de estabilidade em aeronaves e problemas de valor de contorno. Além das aplicações citadas, ressaltamos também outros importantes problemas práticos que envolvem a resolução de sistemas não lineares, como por exemplo, cálculo de estruturas eletrônicas [34], modelos complexos de transporte [7] e modelagem estatística [68].

Diante da vasta gama de aplicações existentes, o estudo e a análise de diferentes técnicas de resolução de sistemas não lineares ainda se fazem necessários. Em vista disso, dada $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$, o objetivo central do nosso trabalho é resolver o sistema

$$F(x) = 0, \tag{1.1}$$

o que não nos impede de assumir objetivos específicos no decorrer do texto, a fim de conciliar as características dos métodos com o propósito de cada um dos problemas considerados. Ao longo do trabalho vamos assumir que F está bem definida e que possui derivadas parciais contínuas em um conjunto aberto $D \subset \mathbb{R}^n$.

Métodos práticos para resolver (1.1) consistem em um processo iterativo. Assim, dada uma aproximação inicial x^0 gera-se uma sequência de iterados x^1, x^2, \dots , com o intuito de aproximar a mesma, a cada nova iteração, de uma solução do problema. Dentre os processos iterativos, o método de Newton é o mais conhecido e, assim como a maioria dos métodos frequentemente utilizados, apresenta um bom comportamento apenas em uma região próxima da solução, isto é, possui convergência local. Para contornar esse problema é usual incorporarmos alguma estratégia de globalização ao algoritmo, dentre as técnicas mais utilizadas destacamos a busca linear [54]. Desta forma, o método passa a desfrutar de convergência global, ou seja, temos a garantia de que ao menos um ponto de acumulação da sequência gerada pelo algoritmo é solução, ou um ponto estacionário da função de mérito associada.

Outra característica importante dos métodos iterativos consiste na taxa de convergência, isto é, a velocidade com a qual garantimos que o algoritmo converge para uma solução, ou ponto estacionário, x^* do problema. Dizemos que um método possui taxa de convergência linear se existem $k_0 \in \mathbb{N}$ e $r \in (0, 1)$ tais que a sequência de iterados $\{x^k\}$ gerada por ele satisfaz

$$\|x^{k+1} - x^*\| \leq r \|x^k - x^*\|$$

sempre que $k \geq k_0$. De maneira equivalente, dizemos que a convergência é superlinear se existe uma sequência $\{r_k\}$ de números positivos que convergem para zero, tal que

$$\|x^{k+1} - x^*\| \leq r_k \|x^k - x^*\|$$

para todo $k = 0, 1, 2, \dots$. Agora, se x^k converge para x^* e existem $k_0 \in \mathbb{N}$, $c > 0$ e $p > 0$ tais que

$$\|x^{k+1} - x^*\| \leq c \|x^k - x^*\|^{p+1}$$

para todo $k \geq k_0$, dizemos que a taxa de convergência é de, pelo menos, $p + 1$. Em particular, quando $p = 1$ temos convergência quadrática.

Desse modo, entendemos que há grande interesse em métodos globais com convergência local quadrática. Assim, no Capítulo 2 estudamos uma implementação computacional para o método de Newton globalizado, levando em consideração um tratamento de mau condicionamento ou singularidade, com o objetivo de resolver sistemas não lineares subdeterminados ou sobredeterminados esparsos. Aplicamos o método estudado na solução de problemas de programação matemática com restrições de complementaridade (MPCC) [43, 57, 59].

Devido à frequência que aparecem em modelos de otimização, o estudo de MPCC tem recebido grande atenção da comunidade de Pesquisa Operacional nos últimos anos. Podemos citar, por exemplo, modelos de rede de transporte [22, 71, 75], otimização bilevel [37], problemas multiobjetivo com restrições de complementaridade [42, 77], mercado de eletricidade [16, 25, 27], programação quadrática com restrições de complementaridade [60], condições de otimalidade [58], aplicações de valor ordenado [2], equilíbrio de oligopólio [76], entre outros.

Neste trabalho consideramos o problema de encontrar uma solução viável simples para o MPCC, isto é, uma solução para o problema de complementaridade horizontal [23] não linear (HNCP). Além disso, adicionando variáveis auxiliares, resolvemos também o problema de encontrar um ponto viável para o MPCC em que a função objetivo atinge uma determinada meta. Para tanto, definimos o algoritmo de Newton com Gradiente Projetado para sistemas subdeterminados (PGUN) que combina iterações rápidas de pontos interiores com passos de gradientes projetados. Para garantir a convergência global do método, empregamos o procedimento de busca linear de Li-Fukushima [41]. Além disso, sob hipóteses razoáveis, estabelecemos as propriedades de convergência local quadrática do algoritmo.

O método PGUN é então aplicado para resolver o HNCP associado à busca de soluções viáveis simples de problemas testes retirados de uma coleção bem conhecida [40]. Os resultados mostram que, para a maioria dos casos, iterações de gradientes projetados quase nunca foram necessárias, permitindo que o método convirja de maneira muito rápida para uma solução do HNCP. Para outros casos, o método PGUN converge lentamente usando direções de gradientes projetados para um ponto estacionário da função de mérito que não parece ser uma solução do problema. A fim de evitar esses casos, introduzimos um critério prático para parar o algoritmo prematuramente e evitar muitas iterações de gradientes projetados. Como a função de mérito não é convexa, a escolha do ponto inicial é muito importante para o sucesso do algoritmo PGUN. Por este motivo, sempre que o algoritmo parar prematuramente devido ao critério mencionado acima, vamos reiniciá-lo com um ponto inicial diferente. Os resultados numéricos obtidos após incorporarmos esses dois critérios (parar prematuramente e mudar o ponto inicial) mostram que, em geral, o método é eficiente para resolver o HNCP e parece ter um desempenho melhor do que o método de Levenberg-Marquardt Projetado [36].

Além desse experimento, também testamos o algoritmo PGUN para resolver o problema de encontrar uma solução viável com meta. Como discutido em [19], acrescentar essa restrição no

valor da função torna o problema mais difícil de resolver, e o algoritmo PGUN apresenta mais dificuldades nesse experimento. Apesar disso, o método proposto foi capaz de fornecer uma solução viável com meta para a grande maioria dos casos testados.

No Capítulo 3 entramos em uma segunda classe de métodos para resolução de sistemas não lineares, os métodos quase-Newton. Estudamos alguns dos métodos quase-Newton que satisfazem a equação secante ou a equação secante inversa. A proposta desses métodos é resolver o sistema não linear (1.1) com menor esforço computacional do que o utilizado pelo método de Newton, mas sem perder as boas propriedades de convergência do mesmo. Desse modo, uma iteração quase-Newton pode ser vista como uma generalização do método de Newton, em que trocamos a matriz Jacobiana $J(x)$ por alguma aproximação $B(x)$, que pode ou não envolver o cálculo de derivadas. A princípio a justificativa mais utilizada para a existência dos métodos quase-Newton baseava-se nos possíveis erros cometidos ao calcularmos as derivadas [45]. No entanto, com o surgimento de técnicas eficientes de diferenciação algorítmica [24], o uso dos métodos quase-Newton simplesmente como forma de eliminar o cálculo do Jacobiano parece menos importante do que naquela época. Nesse contexto, discutimos a utilização dos métodos quase-Newton como aceleradores de processos iterativos lineares.

O método proposto consiste em aplicar um dos métodos quase-Newton estudados ao sistema não linear definido a partir do problema de ponto fixo $x = G(x)$. Problemas de ponto-fixo aparecem de modo natural durante a modelagem de aplicações práticas, como no cálculo de estruturas eletrônicas [34], modelos complexos de transporte [7], modelagem estatística [68], entre outros. De acordo com [70], uma das grandes preocupações acerca da iteração de ponto fixo, está no fato de que os iterados podem convergir com taxa linear e extremamente lenta. Por este motivo, vários autores vêm se dedicando a acelerar de alguma maneira a convergência desses iterados [7, 17, 18, 33, 34, 39, 52, 62, 64, 67, 69, 70, 79].

Métodos quase-Newton que satisfazem a equação secante, como por exemplo, o primeiro método de Broyden [9] e o método de atualização de uma coluna por iteração [44], apresentam maiores dificuldades a medida que o número de iterações aumenta. Com o intuito de reduzir o esforço necessário para atualizar a aproximação da Jacobiana para esses métodos, utilizamos a técnica de atualização da direção quase-Newton exposta em [14]. Adicionalmente, para evitar possíveis instabilidades na definição da nova matriz $B(x)$ mantemos a aproximação anterior sempre que uma condição apropriada não for satisfeita. Visando evitar passos extremamente longos, realizamos um controle de passo simples. Com o objetivo de validar a técnica estudada, realizamos testes em um conjunto de problemas treino. De modo geral, a estratégia quase-Newton é eficiente na aceleração algorítmica, apresentando um desempenho superior ao método original.

A técnica de aceleração quase-Newton é então aplicada ao problema de reconstruir o comportamento cinético de um marcador radioativo durante o exame de perfusão cardíaca com dados reais oriundos de exames de Tomografia por Emissão de Pósitrons (PET), seguindo a modelagem proposta em [61].

O procedimento PET surgiu como uma ferramenta de pesquisa para estudar o cérebro e, apesar de ser um exame de alto custo, fornece uma melhor imagem de diagnóstico de maneira minimamente invasiva, o que permitiu sua transformação em uma ferramenta clínica. Nosso foco é o exame de perfusão cardíaca do miocárdio, mas a técnica PET também é aplicada em uma vasta gama de procedimentos, incluindo detecção e monitoramento da atividade de tumores malignos e de distúrbios cerebrais, como por exemplo, diagnóstico precoce da doença de Alzheimer [50, 55, 61].

O modelo proposto por Reips [61] separa a atividade do marcador radioativo em artéria, tecido (capilares) e veias. Assim, a imagem u associada ao problema é dada pela soma das concentrações C do marcador na artéria, tecido e veias, a saber,

$$u(x, t) = C_{\mathcal{A}}(x, t) + C_{\mathcal{T}}(x, t) + C_{\mathcal{V}}(x, t).$$

Para cada uma das concentrações acima está associada uma velocidade V e um coeficiente de difusão D . Dessa forma, Reips [61] define um sistema de equações diferenciais que descreve a troca de fluidos das artérias para o tecido (capilares), do tecido para as veias e das veias para as artérias.

Após a avaliação dos testes computacionais, Reips [61] conclui que o modelo proposto prevê com boa precisão a atividade do marcador radioativo. No entanto, o elevado número de iterações com um alto custo computacional torna o método pouco competitivo. Nesse contexto, torna-se necessário reduzir o esforço computacional de alguma maneira, e a técnica de aceleração quase-Newton parece suprir essa necessidade.

Visto que os métodos de Newton e quase-Newton estudados nos Capítulos 2 e 3 são, essencialmente, métodos locais, no Capítulo 5 abordamos uma estratégia diferente de globalização. Já vimos que uma possibilidade de globalizar o método de Newton consiste em incluir estratégias de busca linear, o que garante convergência para um ponto estacionário da função de mérito, que pode não ser uma solução de (1.1). Por outro lado, quando o método local converge, ele encontra uma solução do sistema. Isto posto, no Capítulo 5 descrevemos uma estratégia de globalização baseada em homotopias que prioriza o uso da iteração de Newton.

O algoritmo proposto combina o método de Newton com iterações de homotopia. Iniciamos com o método de Newton e, quando este apresenta alguma das dificuldades consideradas, o algoritmo realiza uma iteração homotópica. Com isso esperamos que o algoritmo híbrido proposto apresente um desempenho superior ao método homotópico puro. Consideramos a homotopia regularizante e, seguindo as ideias de [74], resolvemos o problema de valor inicial associado usando o método de Euler.

O algoritmo híbrido é então aplicado em problemas onde o método de Newton apresenta insucesso, mas que podem ser resolvidos utilizando a homotopia regularizante. Experimentos numéricos são considerados para validar a estratégia apresentada.

Diante da breve exposição acima, pode-se notar que buscamos manter as particularidades de cada um dos métodos estudados. Em razão destas diferenças, dividimos o texto em quatro capítulos, acrescido das considerações finais, sendo que os mesmos se estruturam da seguinte maneira: exposição do método ou problema, aplicação numérica e análise dos resultados. Visto que os ensaios computacionais constituem uma parte importante deste trabalho, experimentos numéricos são expostos no decorrer de cada um dos capítulos. Vale ressaltar que tal estrutura facilita a leitura de um tema específico, uma vez que não é necessário consultar todo o trabalho para se inteirar de um dos assuntos abordados.

Assim, no Capítulo 2 estudamos uma implementação computacional esparsa para o método de Newton globalizado e apresentamos o algoritmo PGUN para a solução de problemas com restrições de complementaridade. Os resultados aqui expostos compõem o artigo [3]. Já no Capítulo 3 apresentamos e analisamos a estratégia quase-Newton de aceleração algorítmica para exemplos treino. Na sequência, no Capítulo 4, aplicamos a técnica quase-Newton para o problema PET, realizamos experimentos numéricos com base em imagens reais. No Capítulo 5 colocamos o algoritmo híbrido de Newton com homotopia e apresentamos os resultados iniciais. Finalmente, no último capítulo as considerações finais sobre o trabalho são colocadas.

Notação:

- Denotamos por \mathbb{N} o conjunto dos números naturais e por \mathbb{R} o conjunto dos números reais.
- Dado $x \in \mathbb{R}^n$, $\|x\|$ representa a norma Euclidiana de x , isto é, $\|x\| = \sqrt{x^\top x}$.
- Vamos denotar por X a matriz diagonal cujas entradas são os elementos de $x \in \mathbb{R}^n$.
- Dada $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, denotamos a matriz Jacobiana de F por

$$J(x) \equiv F'(x) \equiv \begin{bmatrix} f'_1(x) \\ \vdots \\ f'_m(x) \end{bmatrix} \equiv \begin{bmatrix} \nabla f_1(x)^\top \\ \vdots \\ \nabla f_m(x)^\top \end{bmatrix} \equiv \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(x) & \cdots & \frac{\partial f_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1}(x) & \cdots & \frac{\partial f_m}{\partial x_n}(x) \end{bmatrix}$$

e, $B(x)$ representa uma aproximação para $J(x)$.

- Para uma matriz $A \in \mathbb{R}^{m \times n}$ a pseudoinversa de Moore-Penrose será denotada por A^\dagger .
- Dada uma matriz $A \in \mathbb{R}^{m \times n}$, $\|A\|_F$ denota a norma de Frobenius de A , isto é, $\|A\|_F = \sqrt{\text{traço}(A^\top A)}$.

Capítulo 2

O Método de Newton

2.1 Introdução

Considere o sistema não linear definido por

$$F(x) = 0 \tag{2.1}$$

com $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuamente diferenciável.

O método de Newton, mais conhecido no contexto de busca por zeros de funções, foi generalizado pela primeira vez para a resolução de sistemas não lineares por Simpson [78]. Tal método visa resolver o sistema não linear (2.1) através de uma sequência de aproximações lineares do mesmo. Para tanto, consideramos a aproximação de Taylor de primeira ordem de $F(x)$ em uma vizinhança do iterado atual x^k , isto é,

$$F(x) \approx F(x^k) + J(x^k)(x - x^k). \tag{2.2}$$

Desta forma, quando $J(x^k)$ é não singular, o processo iterativo que define o método de Newton puro, a saber,

$$x^{k+1} = x^k - J(x^k)^{-1}F(x^k). \tag{2.3}$$

pode ser visto como a sequência de soluções do problema (2.2). Logo, a cada nova iteração, busca-se melhorar a aproximação de uma solução do problema (2.1). Destacamos que essa formulação do método de Newton, utilizando a notação do cálculo moderno, deve-se a Lagrange [78].

A ampla utilização do método de Newton decorre de suas características, entre elas a convergência local quadrática. Além disso, o método é, até então, o único invariante tanto por transformações lineares invertíveis no domínio quanto no contradomínio. Vale ressaltar ainda outra propriedade importante deste método: a direção de Newton pura $x^{k+1} - x^k$ é uma direção de descida para todos os módulos das componentes que não se anulam em x^k , ou seja, a direção de Newton melhora localmente não apenas a norma do sistema, mas todas as componentes que ainda podem ser aprimoradas [45].

Por outro lado, para cada iteração é necessário calcular a matriz Jacobiana $J(x^k)$, resolver o sistema linear

$$J(x^k)s^k = -F(x^k) \tag{2.4}$$

e atualizar o iterado $x^{k+1} = x^k + s^k$. O custo computacional deste processo pode tornar o método de Newton inviável para problemas de grande porte. Nesse contexto, se faz necessário aproveitar, sempre que possível, a estrutura esparsa do Jacobiano.

No decorrer deste capítulo comentamos os detalhes da implementação do método de Newton puro, expomos as variações estudadas e, por fim, utilizamos o algoritmo implementado para resolver um problema prático, a saber, o problema de programação matemática com restrições de complementaridade.

2.2 Implementação Computacional

Essa seção está dividida em cinco subseções que descrevem os detalhes da implementação numérica do método de Newton estudado. Consideramos a resolução de sistemas subdeterminados e sobredeterminados, tratamento de mau condicionamento ou singularidade, controle de passo, análise de progresso e convergência, levando em consideração a possível estrutura esparsa do Jacobiano.

2.2.1 Sistemas subdeterminados e sobredeterminados

Considere agora $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \neq n$. Neste caso $J(x) \in \mathbb{R}^{m \times n}$ e o sistema linear em (2.4) pode ser do tipo subdeterminado, quando $m < n$, ou sobredeterminado, caso $m > n$. Para estes casos usaremos a direção de Newton generalizada [6], a saber,

$$s^k = -J(x^k)^\dagger F(x^k) \quad (2.5)$$

em que $J(x^k)^\dagger$ denota a pseudoinversa de Moore-Penrose de $J(x^k)$.

Consequentemente, o cálculo da direção de Newton sofre pequenas modificações, conforme segue:

- (i) Para sistemas subdeterminados, $J(x^k)^\dagger = J(x^k)^\top (J(x^k)J(x^k)^\top)^{-1}$ e a direção de Newton fica definida por

$$s^k = -J(x^k)^\top (J(x^k)J(x^k)^\top)^{-1} F(x^k);$$

- (ii) Para sistemas sobredeterminados, $J(x^k)^\dagger = (J(x^k)^\top J(x^k))^{-1} J(x^k)^\top$ e a direção de Newton fica definida por

$$s^k = -(J(x^k)^\top J(x^k))^{-1} J(x^k)^\top F(x^k).$$

2.2.2 Mau condicionamento ou singularidade

Seja $A \in \mathbb{R}^{m \times n}$ uma matriz de posto completo. Dizemos que A é mal condicionada quando seu número de condição

$$\kappa(A) = \|A\| \|A^\dagger\|$$

é “grande”. Em particular, considerando a norma Euclidiana, temos

$$\kappa_2(A) = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)},$$

em que $\sigma_{\min}(A)$ e $\sigma_{\max}(A)$ denotam o menor e o maior valor singular de A , respectivamente.

Da análise de erros de sistemas lineares (ver, por exemplo, [13]), sabemos que há a seguinte relação entre o número de condição do Jacobiano e a solução do sistema Newtoniano (2.4): se $\sigma_{\min}(J(x^k))$ é “muito pequeno” então a direção de Newton s^k é mal condicionada e possivelmente “muito grande”.

As sub-rotinas de álgebra linear utilizadas fornecem informação sobre mau condicionamento ou quase singularidade de $J(x^k)$. Quando isso ocorre adotamos a conhecida estratégia de regularização de Tikhonov [66] e, para $m = n$, resolvemos o sistema

$$\begin{aligned} (J(x^k)J(x^k)^\top + \mu_k I)y &= -F(x^k) \\ s^k &= J(x^k)^\top y \end{aligned}$$

com $\mu_k > 0$.

Analogamente, para os sistemas subdeterminado e sobredeterminado encontramos a direção de Newton por

$$s^k = -J(x^k)^\top (J(x^k)J(x^k)^\top + \mu_k I)^{-1} F(x^k)$$

e

$$s^k = -(J(x^k)^\top J(x^k) + \mu_k I)^{-1} J(x^k)^\top F(x^k)$$

respectivamente.

2.2.3 Controle de passo

Como vimos na subseção anterior, quando a matriz $J(x^k)$ do sistema (2.4) é quase singular o passo s^k pode ser extremamente grande. Nesse contexto, controles eficazes de passo são importantes para o bom desempenho de algoritmos.

Moré & Cosnard [49] propuseram interessantes sistemas de controle de passo para métodos de resolução de sistemas não lineares quando não se possui um mecanismo de redução de alguma função objetivo. Assim, vamos normalizar o passo s^k exigindo que

$$\|s^k\| \leq c_1 \|x^k\|$$

em que $c_1 > 0$, ou então, $\|s^k\| \leq c_1$ quando $\|x^k\| = 0$. Se as desigualdades acima não forem satisfeitas, atualizamos o passo s^k por

$$s^k = \frac{c_1 \|x^k\|}{\|s^k\|} s^k \tag{2.6}$$

ou $s^k = \frac{c_1}{\|s^k\|} s^k$, respectivamente.

2.2.4 Progresso e convergência

Com o objetivo de resolver $F(x) = 0$, em geral, se espera que o valor de $\|F(x^{k+1})\|$ seja suficientemente menor que $\|F(x^k)\|$. O método de Newton puro não satisfaz necessariamente esse critério. Para contornar esse problema costuma-se utilizar alguma estratégia de globalização [54], a mais conhecida é a busca linear. Assim, em cada iteração, dada uma direção decidimos o quanto seguir através dela, com o propósito de satisfazer uma condição de decréscimo suficiente. O próximo teorema apresenta uma condição bem conhecida na literatura, a saber, a Condição de Armijo.

Teorema 2.1. Sejam $f(x) = \frac{1}{2}\|F(x)\|^2$, $x, s \in \mathbb{R}^n$ tais que $\nabla f(x) \neq 0$, $\nabla f(x)^\top s < 0$ e $\gamma \in (0, 1)$. Então, existe $\varepsilon > 0$ de forma que

$$f(x + ts) \leq f(x) + \gamma t \nabla f(x)^\top s,$$

para todo $t \in (0, \varepsilon)$.

Demonstração. Ver Lema 3.1, página 35, de Nocedal & Wright [54]. □

Quando incorporamos a Condição de Armijo na busca linear, a sequência dos valores da função de mérito $f(x) = \frac{1}{2}\|F(x)\|^2$ torna-se monótona decrescente. No entanto, exigir monotonicidade pode tornar o algoritmo muito lento, no sentido de serem necessários muitos passos da busca linear e, conseqüentemente, muitas avaliações da função. Nesse contexto, utilizamos uma técnica de busca linear não monótona proposta em [41] e apresentada na definição a seguir.

Definição 2.2. Seja $\{\eta_k\}$ uma sequência de números positivos tal que

$$\sum_{k=0}^{\infty} \eta_k < \infty.$$

Sejam $\{x^k\}$ uma sequência definida por $x^{k+1} = x^k + t_k s^k$, $\sigma > 0$ e $t_k \in (0, 1)$. Diremos que x^{k+1} satisfaz a condição de Li-Fukushima se,

$$\|F(x^k + t_k s^k)\| \leq (1 + \eta_k) \|F(x^k)\| - \sigma \|t_k s^k\|^2.$$

Em todas as alternativas supomos que s^k foi obtida por alguma das possibilidades anteriormente mencionadas.

2.2.5 Esparsidade

Tendo em vista o objetivo de resolver sistemas esparsos, utilizamos a estrutura usual para definir uma matriz esparsa, a saber:

`nz`: número de possíveis elementos não nulos de $J(x)$;

`j`: vetor com os elementos não nulos do Jacobiano $J(x)$;

`lin(i)`: índice da linha do elemento armazenado em $j(i)$, $i = 1, \dots, \text{nz}$;

`col(i)`: índice da coluna do elemento armazenado em $j(i)$, $i = 1, \dots, \text{nz}$;

`comj(i)`: índice do primeiro elemento da coluna i de $J(x)$;

`numj(i)`: número de elementos não nulos da coluna i de $J(x)$.

Exemplo 2.1. Estrutura esparsa:

Para reescrever a matriz

$$\begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{pmatrix}$$

na forma esparsa apresentada acima, basta definirmos

$$\text{nz} = 5, \text{ j} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \\ -1 \end{pmatrix}, \text{ lin} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 1 \\ 3 \end{pmatrix}, \text{ col} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 3 \\ 4 \end{pmatrix}, \text{ comj} = \begin{pmatrix} 1 \\ 2 \\ 4 \\ 5 \end{pmatrix} \text{ e } \text{ numj} = \begin{pmatrix} 1 \\ 2 \\ 1 \\ 1 \end{pmatrix}.$$

A fim de resolver o sistema (2.4) com a matriz $J(x^k)$ na forma esparsa, utilizamos a sub-rotina `ma48` [26]. Essa sub-rotina resolve sistemas lineares esparsos e não simétricos via eliminação gaussiana e permite algumas escolhas particulares, entre elas, possibilita escolher a ordem dos pivôs, visando um equilíbrio entre esparsidade e precisão.

Podemos agora estabelecer o método de Newton esparsa, descrito no algoritmo abaixo.

Algoritmo 2.1. Método de Newton esparsa

Inicialização: Sejam $c_1 > 0$ e $x^0 \in \mathbb{R}^n$. Faça $k = 0$.

Passo 1. *Critério de parada:* Se $\|F(x^k)\| = 0$, pare.

Passo 2. *Cálculo da direção de Newton:* Encontre s^k resolvendo (2.4) ou (2.5). Caso não tenha sido possível completar a decomposição da matriz, tome $\mu_k > 0$ e resolva o sistema correspondente descrito na subseção 2.2.2.

Passo 3. *Controle de passo:* Se $\|s^k\| > c_1\|x^k\|$, atualize s^k como em (2.6).

Passo 4. *Cálculo do próximo iterado:* Faça $x^{k+1} = x^k + t_k s^k$, com $t_k = 1$ para o método de Newton puro, ou t_k satisfazendo a Definição 2.2. Faça $k = k + 1$ e volte ao Passo 1.

Na próxima seção, aplicaremos o Algoritmo 2.1 na resolução de problemas com restrições de complementaridade. Este estudo foi publicado na revista *European Journal of Operational Research* [3].

2.3 Estratégia de pontos interiores para MPCC: Uma aplicação do Algoritmo 2.1

Problemas de programação matemática com restrições de complementaridade (MPCC) [43, 57, 59] aparecem frequentemente em otimização e uma vez que estão relacionados com a noção de equilíbrio de um sistema, têm importantes aplicações em engenharia, economia e ciências [20]. Tais problemas podem ser escritos na forma

$$\begin{aligned} &\text{Minimizar} && \varphi(x, y, w) \\ &s.a. && H(x, y, w) = 0 \\ &&& \min\{x, w\} = 0 \end{aligned} \tag{2.7}$$

em que $x, w \in \mathbb{R}^n$, $y \in \mathbb{R}^m$ e $\varphi : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}$, $H : \mathbb{R}^{2n+m} \rightarrow \mathbb{R}^r$ são funções continuamente diferenciáveis. Denotamos por $\min\{x, w\}$ o vetor cujas componentes são $\min\{x_i, w_i\}$, $i = 1, \dots, n$. As variáveis de cada par (x_i, w_i) são ditas complementares e satisfazem

$$x_i \geq 0, w_i \geq 0, x_i w_i = 0, i = 1, \dots, n. \quad (2.8)$$

Note que (2.8) é equivalente a

$$x^\top w = 0, x_i \geq 0, w_i \geq 0, i = 1, \dots, n, \quad (2.9)$$

sendo esta a condição de complementaridade que normalmente é utilizada na definição de MPCC [43].

Um MPCC pode ser visto como um problema de programação não linear (PNL) quando as n restrições de complementaridade $\min\{x_i, w_i\} = 0$ são substituídas por (2.8) ou (2.9). Porém, resolver (2.7) utilizando as técnicas tradicionais de PNL pode apresentar algumas dificuldades, entre elas soluções que não satisfazem as Condições KKT. Esse empecilho decorre dos pontos chamados duplo zeros ou índices biativos, aqueles em que para uma determinada restrição $x_i w_i = 0$ resulta que $x_i = w_i = 0$.

Neste trabalho estamos interessados em encontrar uma solução viável simples para o MPCC, isto é, uma solução para o seguinte problema de complementaridade horizontal [23] não linear (HNCP)

$$\begin{pmatrix} H(x, y, w) \\ x_1 w_1 \\ \vdots \\ x_n w_n \end{pmatrix} = 0, x \geq 0, w \geq 0. \quad (2.10)$$

Vamos assumir que $r \leq n + m$. Assim o número de equações em (2.10) é menor ou igual ao número de variáveis. O caso em que $r = n + m$ foi estudado em [4]. O caso em que H é uma função afim também já foi muito estudado na literatura (para um trabalho recente, veja [35]), e é NP-hard [51]. No entanto, existem muitos MPCCs em que encontrar uma solução viável simples pode ser considerada uma tarefa fácil [35].

Vale ressaltar que em muitos casos práticos estamos mais interessados em obter uma solução viável próxima de um minimizador global do que em minimizar a função objetivo de (2.7). Dessa forma, podemos procurar por soluções viáveis que satisfazem uma determinada meta, isto é, vamos resolver o seguinte HNCP

$$\varphi(x, y, w) \leq c_t, H(x, y, w) = 0, x \geq 0, w \geq 0 \text{ and } x^\top w = 0, \quad (2.11)$$

em que c_t representa uma meta a ser atingida. Note que (2.11) é equivalente a

$$\varphi(x, y, w) + v_1 = c_t, H(x, y, w) = 0, x \geq 0, w \geq 0, x^\top w = 0, \text{ e } v_1 v_2 = 0 \quad (2.12)$$

em que v_1 é uma variável de folga e v_2 é uma variável auxiliar. Além disso, o HNCP (2.12) é subdeterminado se $r < n + m$, a hipótese mencionada anteriormente.

Note que o sistema (2.12) pode ser reescrito de maneira semelhante ao problema (2.10), para tal, basta incluirmos a restrição do valor da função como uma componente de H . Assim, os dois problemas de viabilidade, (2.10) e (2.12), podem ser escritos como um sistema de equações não lineares com restrições de não negatividade, a saber

$$F(x, y, w) = 0, x \geq 0, w \geq 0, \quad (2.13)$$

em que $F : \mathbb{R}^{n+m+n} \rightarrow \mathbb{R}^{r+n}$ é dada por

$$F(x, y, w) = \begin{pmatrix} H(x, y, w) \\ x_1 w_1 \\ \vdots \\ x_n w_n \end{pmatrix}, \quad (2.14)$$

e, com um pequeno abuso de notação, $H : \mathbb{R}^{n+m+n} \rightarrow \mathbb{R}^r$.

Definimos a função de mérito

$$f(x, y, w) = \frac{1}{2} \|F(x, y, w)\|^2 \quad (2.15)$$

e consideramos o problema

$$\text{Minimizar } f(x, y, w) \text{ sujeito a } (x, y, w) \in \Omega, \quad (2.16)$$

em que $\Omega = \{(x, y, w) \in \mathbb{R}^{n+m+n} : x \geq 0 \text{ e } w \geq 0\}$.

De agora em diante, vamos denotar $z = \begin{pmatrix} x \\ y \\ w \end{pmatrix}$.

Sabe-se que, se z^* é um ponto estacionário de um problema irrestrito $\|\Phi(z)\|^2$ e o resíduo $\Phi(z^*)$ não é nulo, então as linhas da matriz $\Phi'(z^*)$ são linearmente dependentes. Esta propriedade, em geral, não é válida para o caso restrito. Na sequência, generalizamos um resultado de [5] e provamos que a propriedade de posto incompleto mencionada acima é válida para o problema (2.16). Este resultado será utilizado para provar as propriedades de convergência do algoritmo proposto para resolver (2.13).

Teorema 2.3. Suponha que $\bar{z} = (\bar{x}^\top, \bar{y}^\top, \bar{w}^\top)^\top \in \Omega$ é um ponto estacionário de (2.16). Então,

- (i) se $H(\bar{z}) = 0$ ou $H'_y(\bar{z})$ é uma matriz de posto completo, então \bar{z} é solução de (2.13);
- (ii) se $\|F(\bar{z})\| \neq 0$, então as linhas do Jacobiano $F'(\bar{z})$ são linearmente dependentes.

Demonstração. Por hipótese \bar{z} é ponto estacionário de (2.16) então, das condições KKT, temos que existem $\gamma, \alpha \in \mathbb{R}^n$ tais que

$$\begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & \bar{X} \end{bmatrix} \begin{bmatrix} H(\bar{z}) \\ \bar{x}_1 \bar{w}_1 \\ \vdots \\ \bar{x}_n \bar{w}_n \end{bmatrix} = \begin{bmatrix} \gamma \\ 0 \\ \alpha \end{bmatrix} \quad (2.17)$$

e

$$\begin{aligned} \bar{x}_i \gamma_i &= 0, \quad i = 1, \dots, n, \\ \bar{w}_i \alpha_i &= 0, \quad i = 1, \dots, n, \\ \bar{x} &\geq 0, \quad \bar{w} \geq 0, \quad \gamma \geq 0, \quad \alpha \geq 0, \end{aligned} \quad (2.18)$$

em que $\bar{X}, \bar{W} \in \mathbb{R}^{n \times n}$ denotam a matriz diagonal cujas entradas são $\bar{x}_1, \dots, \bar{x}_n$ e $\bar{w}_1, \dots, \bar{w}_n$, respectivamente.

Para provar o item (i) vamos primeiro considerar $H(\bar{z}) = 0$, assim

$$\begin{bmatrix} \gamma \\ 0 \\ \alpha \end{bmatrix} = \begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & \bar{X} \end{bmatrix} \begin{bmatrix} 0 \\ \bar{x}_1\bar{w}_1 \\ \vdots \\ \bar{x}_n\bar{w}_n \end{bmatrix} = \begin{bmatrix} \bar{x}_1\bar{w}_1^2 \\ \vdots \\ \bar{x}_n\bar{w}_n^2 \\ 0 \\ \bar{x}_1^2\bar{w}_1 \\ \vdots \\ \bar{x}_n^2\bar{w}_n \end{bmatrix}.$$

Segue que $\bar{x}_i\bar{w}_i^2 = \gamma_i$, $\bar{x}_i^2\bar{w}_i = \alpha_i$, para todo $i = 1, \dots, n$, e por (2.18)

$$\bar{x}_i^2\bar{w}_i^2 = \bar{x}_i\gamma_i = 0, \quad i = 1, \dots, n$$

$$\bar{x}_i^2\bar{w}_i^2 = \bar{w}_i\alpha_i = 0, \quad i = 1, \dots, n.$$

Ou seja, $\bar{x}_i\bar{w}_i = 0$ para todo $i = 1, \dots, n$ e \bar{z} é solução de (2.13).

Considere agora que $H'_y(\bar{z})$ tem posto completo. Por (2.17) segue que $H'_y(\bar{z})^\top H(\bar{z}) = 0$, ou seja, temos uma combinação linear nula das linhas de $H'_y(\bar{z})$. Logo, $H(\bar{z}) = 0$ e, portanto, \bar{z} é solução de (2.13).

Agora, para provar o item (ii), suponha que $F(\bar{z}) \neq 0$.

Se $\bar{x}_i = \bar{w}_i = 0$ para algum $i \in \{1, \dots, n\}$, temos que a coluna $r + i$ da matriz

$$F'(z)^\top = \begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & \bar{X} \end{bmatrix}$$

é nula. Isto é, $F'(\bar{z})$ tem linhas linearmente dependentes.

Vamos assumir que $\bar{x}_{i_k} > 0$ e $\bar{w}_{i_k} > 0$ para q índices i_k , $k = 1, \dots, q$, $k \in \{1, \dots, n\}$. Temos três possibilidades para analisar:

Caso 1: $q = n$;

Caso 2: $q = 0$;

Caso 3: $1 \leq q < n$.

Para o Caso 1, note que (2.18) implica que $\gamma = \alpha = 0$, isto é

$$\begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & \bar{X} \end{bmatrix} \begin{bmatrix} H(\bar{z}) \\ \bar{x}_1\bar{w}_1 \\ \vdots \\ \bar{x}_n\bar{w}_n \end{bmatrix} = 0,$$

com $\bar{x}_i, \bar{w}_i > 0$ para todo $i = 1, \dots, n$. Ou seja, temos uma combinação linear nula das linhas de $F'(\bar{z})$ com coeficientes não nulos. Logo, $F'(\bar{z})$ tem linhas linearmente dependentes.

Considere agora o Caso 2. Uma vez que já consideramos o caso em que $\bar{x}_i = \bar{w}_i = 0$ para algum $i \in \{1, \dots, n\}$, temos que $\bar{x}_i + \bar{w}_i > 0$ para todo $i = 1, \dots, n$. Vamos assumir, sem perda de generalidade, que $\bar{x}_i = 0$ e $\bar{w}_i > 0$ para todo $i = 1, \dots, n$. Então, por (2.17), temos que

$$\begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\bar{z}) \\ 0 \end{bmatrix} = \begin{bmatrix} \gamma \\ 0 \\ 0 \end{bmatrix}. \quad (2.19)$$

Portanto,

$$\begin{bmatrix} H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\bar{z}) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

o que implica que a matriz

$$\begin{bmatrix} H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & 0 \end{bmatrix}$$

possui, no máximo, $r - 1$ colunas linearmente independentes, pois, caso contrário, teríamos $H(\bar{z}) = 0$. Logo, a matriz

$$\begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & 0 \end{bmatrix}$$

possui, no máximo, $n + r - 1$ colunas linearmente independentes. Lembrando que $\bar{X} = 0$, a matriz

$$\begin{bmatrix} H'_x(\bar{z})^\top & \bar{W} \\ H'_y(\bar{z})^\top & 0 \\ H'_w(\bar{z})^\top & \bar{X} \end{bmatrix}$$

também contém, no máximo, $n + r - 1$ colunas linearmente independentes.

Logo, $F'(\bar{z})$ possui, no máximo, $n + r - 1$ linhas linearmente independentes. Como $F'(\bar{z}) \in \mathbb{R}^{(n+r) \times (n+m+n)}$ segue que este Jacobiano não tem posto linha completo.

Por fim, considere o Caso 3. Vamos supor, sem perda de generalidade, que

$$\bar{x}_i, \bar{w}_i > 0 \text{ para } i = 1, \dots, q < n \quad (2.20)$$

e

$$\bar{x}_i = 0, \bar{w}_i > 0 \text{ para } i = q + 1, \dots, n. \quad (2.21)$$

De (2.17) e considerando (2.18), (2.20) e (2.21) segue que

$$H'_x(\bar{z})^\top H(\bar{z}) + \bar{W} \begin{bmatrix} \bar{x}_1 \bar{w}_1 \\ \vdots \\ \bar{x}_q \bar{w}_q \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \gamma_{q+1} \\ \vdots \\ \gamma_n \end{bmatrix}$$

e

$$H'_w(\bar{z})^\top H(\bar{z}) + \bar{X} \begin{bmatrix} \bar{x}_1 \bar{w}_1 \\ \vdots \\ \bar{x}_q \bar{w}_q \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Denotando por $\widehat{H}'_x(\bar{z})^\top$ a matriz formada pelas primeiras q linhas de $H'_x(\bar{z})^\top$, $\tilde{H}'_x(\bar{z})^\top$ a matriz contendo as últimas $n - q$ linhas, \widehat{W} a matriz diagonal $q \times q$ cujas entradas são $\bar{w}_1, \dots, \bar{w}_q$

e \tilde{W} a matriz diagonal com entradas $\bar{w}_{q+1}, \dots, \bar{w}_n$, obtemos

$$\begin{bmatrix} \widehat{H}'_x(\bar{z})^\top & \widehat{W} \\ \widehat{H}'_y(\bar{z})^\top & 0 \\ \widehat{H}'_w(\bar{z})^\top & \widehat{X} \\ \tilde{H}'_w(\bar{z})^\top & 0 \end{bmatrix} \begin{bmatrix} H(\bar{z}) \\ \bar{x}_1 \bar{w}_1 \\ \vdots \\ \bar{x}_q \bar{w}_q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.22)$$

Logo, a matriz

$$A = \begin{bmatrix} \widehat{H}'_x(\bar{z})^\top & \widehat{W} \\ \widehat{H}'_y(\bar{z})^\top & 0 \\ \widehat{H}'_w(\bar{z})^\top & \widehat{X} \\ \tilde{H}'_w(\bar{z})^\top & 0 \end{bmatrix}$$

possui, no máximo, $r + q - 1$ colunas linearmente independentes, uma vez que (2.22) representa uma combinação linear nula das colunas de A .

Note que $F'(\bar{z})$ pode ser construída adicionando as $n - q$ linhas e colunas faltantes na matriz A . Logo, $F'(\bar{z})$ possui, no máximo, $n + r - 1$ colunas linearmente independentes. Portanto, $F'(\bar{z})$ não possui posto linha completo, o que conclui nossa análise. \square

Para resolver (2.13) podemos usar uma abordagem Newtoniana, e portanto de convergência quadrática, que envolve apenas as primeiras derivadas de φ e H , conforme descrito na próxima seção.

2.3.1 Algoritmo PGUN

Nesta seção apresentamos um algoritmo para resolver (2.13) que gera pontos viáveis e combina os métodos de Newton e Gradiente Projetado. Este algoritmo é uma extensão do método introduzido em [4] para a resolução do mesmo sistema não linear com $r = n + m$, isto é, quando o número de restrições é igual ao número de variáveis. A implementação baseou-se no método de Newton exposto na seção anterior.

Algoritmo 2.2. Algoritmo de Newton com Gradiente Projetado para sistemas sub-determinados - PGUN

Inicialização: Considere $\gamma_k > 0$, para todo $k \in \mathbb{N}$, tal que

$$\sum_{k=0}^{\infty} \gamma_k < \infty.$$

Seja $\tau \in (0, 1)$, $\sigma \in (0, 1)$, $0 < \bar{\eta}_1 < \bar{\eta}_2$, $\rho > 0$, $\beta \in (0, \frac{1}{2})$, $c_{\text{big}} > c_{\text{small}} > 0$, $c_{\text{small}} < 1$.
Seja $z^0 = ((x^0)^\top, (y^0)^\top, (w^0)^\top)^\top \in \text{int}(\Omega)$. Assuma que

$$z^k = ((x^k)^\top, (y^k)^\top, (w^k)^\top)^\top \in \text{int}(\Omega),$$

$\sigma_k \in [0, \frac{1}{6}]$, $\tau_k \in [\tau, 1)$, e $\eta_k \in [\bar{\eta}_1, \bar{\eta}_2]$.

Faça $k = 0$.

Passo 1. Critério de parada: Calcule

$$g(z^k, \eta_k) = P_\Omega(z^k - \eta_k \nabla f(z^k)) - z^k,$$

em que P_Ω é a projeção ortogonal sobre Ω . Se $g(z^k, \eta_k) = 0$, pare.

Passo 2. Cálculo da direção de Newton: Calcule

$$d^k = ((d_x^k)^\top, (d_y^k)^\top, (d_w^k)^\top)^\top \in \mathbb{R}^{n+m+n}$$

satisfazendo

$$H'(z^k)d^k + H(z^k) = 0 \quad (2.23)$$

e

$$x_i^k w_i^k + x_i^k (d_w^k)_i + w_i^k (d_x^k)_i = \mu_i^k, \quad (2.24)$$

em que

$$\|\mu^k\|_\infty \leq \sigma_k \frac{(x^k)^\top w^k}{n}. \quad (2.25)$$

Se tal direção d^k não existe ou se $\|d^k\| > c_{\text{big}}$, vá para o Passo 4.

Passo 3. Cálculo do passo máximo: Escolha $\tau_k \in [\tau, 1)$. Calcule

$$\alpha_k^{\text{break}} = \max\{\alpha \geq 0 : z^k + \alpha d^k \in \Omega\} \quad (2.26)$$

e

$$\alpha_k^{\text{max}} = \min\{1, \tau_k \alpha_k^{\text{break}}\}. \quad (2.27)$$

Se $\alpha_k^{\text{max}} \leq c_{\text{small}} \min\{1, \|d^k\|\}$, vá para o Passo 4. Caso contrário, vá para o Passo 5.

Passo 4. Direção do gradiente projetado: Calcule (ou redefina) $d^k = g(z^k, \eta_k)$, e faça $\alpha_k^{\text{max}} = \tau_k$.

Passo 5. Busca linear: Defina $\alpha = \alpha_k^{\text{max}}$.

Passo 5.1. Se

$$\|F(z^k + \alpha d^k)\| \leq \|F(z^k)\| - \rho \|\alpha d^k\|^2 + \gamma_k \quad (2.28)$$

defina $\alpha_k = \alpha$ e vá para o Passo 6.

Passo 5.2 Escolha $\alpha_{\text{new}} \in [\beta\alpha, (1 - \beta)\alpha]$, defina $\alpha = \alpha_{\text{new}}$ e retorne ao Passo 5.1.

Passo 6. Cálculo do novo iterado: Escolha $z^{k+1} \in \Omega$ tal que

$$\|F(z^{k+1})\| \leq \|F(z^k + \alpha_k d^k)\|. \quad (2.29)$$

Fim.

Note que, quando $\sigma_k = 0$, então $\mu^k = 0$ e a direção de Newton do Passo 2 do Algoritmo PGUN é exatamente a mesma direção calculada no Passo 2 do Algoritmo 2.1.

Lembrando que $\Omega = \{(x, y, w) \in \mathbb{R}^{n+m+n} : x \geq 0 \text{ e } w \geq 0\}$, temos que $z^k + \alpha d^k \in \Omega$ sempre que $x^k + \alpha d_x^k \geq 0$ e $w^k + \alpha d_w^k \geq 0$. Assim, no Passo 3 do Algoritmo PGUN, só precisamos nos preocupar com o valor de α quando $[d_x^k]_i < 0$ ou $[d_w^k]_i < 0$ para algum $i \in \{1, \dots, n\}$. Quando isso ocorre, definimos

$$\alpha = \min_{[d_x^k]_i < 0} \left\{ \frac{-x_i^k}{[d_x^k]_i} \right\} \text{ ou } \alpha = \min_{[d_w^k]_i < 0} \left\{ \frac{-w_i^k}{[d_w^k]_i} \right\}$$

respectivamente.

Observe ainda que o algoritmo PGUN está bem definido, uma vez que para um ponto z^k que não satisfaz o critério de parada $g(z^k, \eta_k) = 0$, o Passo 5 do algoritmo é trivialmente satisfeito usando $\gamma_k > 0$. Além disso, o Algoritmo PGUN desfruta de convergência global, conforme estabelece o teorema a seguir.

Teorema 2.4. Dado $z^k = ((x^k)^\top, (y^k)^\top, (w^k)^\top)^\top$ tal que $x^k > 0$, $w^k > 0$ e $g(z^k, \eta_k) \neq 0$, o ponto $(x^{k+1}, y^{k+1}, w^{k+1}) \in \text{int}(\Omega)$ está sempre bem definido. Além disso, se $\{z^k\}$ é uma sequência gerada pelo Algoritmo PGUN e z^* é um ponto de acumulação tal que

$$\lim_{k \in K_1} z^k = z^*,$$

em que $K_1 \subset \mathbb{N}$ é uma subsequência infinita de índices, então:

- (i) z^* é ponto estacionário de Minimizar $f(z)$ sujeito a $z \in \Omega$;
- (ii) Se $F'(z^*)$ é matriz de posto linha completo, então $F(z^*) = 0$;
- (iii) Se K_1 contém infinitos índices k tais que d^k foi calculada como uma direção de Newton, então $F(z^*) = 0$.

Demonstração. A demonstração dos itens (i) e (iii) é exatamente igual à apresentada em [4], onde foi considerado $r = n + m$. Resta provarmos o item (ii).

Suponha, por contradição, que $F(z^*) \neq 0$. Pelo item (i) z^* é ponto estacionário de (2.16). Então, segue do Teorema 2.3 que as linhas de $F'(z^*)$ são linearmente dependentes, contradizendo a hipótese em (ii). Portanto, $F(z^*) = 0$. \square

2.3.2 Convergência local

Nesta seção vamos discutir a convergência local da sequência gerada pelo algoritmo PGUN. Para tanto, assumiremos que sempre que o sistema linear dado por (2.23) e (2.24) é compatível, a direção d^k calculada no Passo 2 do algoritmo PGUN será a solução de mínima norma desse sistema. Isto implica que d^k pertence ao espaço coluna de $F'(z^k)^\top$. Assim,

$$d^k = F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix}, \quad (2.30)$$

em que $\mu_k \geq 0$ satisfaz (2.25), $e = (1, \dots, 1)^\top$ e X_k e W_k denotam as matrizes diagonais cujas entradas são os elementos de x^k e w^k , respectivamente.

Na sequência, provamos que em uma vizinhança de uma solução z^* de (2.13) tal que $F'(z^*)$ é uma matriz de posto linha completo, podemos considerar o passo máximo α_k^{max} tão próximo de 1 quanto desejarmos. Esse resultado foi provado para o caso $n + m = r$ em [4]. Com o próximo teorema, apresentamos a prova para o caso retangular estudado, visto que a solução do sistema linear Newtoniano não é única.

Teorema 2.5. Assuma que o algoritmo PGUN é aplicado ao problema (2.13) e que z^* é uma solução tal que o Jacobiano $F'(z^*)$ tem posto completo. Dado $\delta \in (0, 1)$, existe $\varepsilon > 0$ tal que, d^k está bem definida por (2.23) e (2.24), e $\alpha_k^{break} \geq \delta$, sempre que $\|z^k - z^*\| \leq \varepsilon$.

Demonstração. Assuma que $F'(z^*)$ tem posto completo e $F(z^*) = 0$. Note que,

$$F'(z) = \begin{bmatrix} H'_x(z) & H'_y(z) & H'_w(z) \\ W & 0 & X \end{bmatrix} \in \mathbb{R}^{(r+n) \times (n+m+n)}.$$

Como $F'(z^*)$ tem posto completo, x_i^* e w_i^* não podem ser ambos nulos. Vamos assumir, sem perda de generalidade, que $x_i^* = 0$ e $w_i^* > 0$ para todo $i = 1, \dots, n$. Assim,

$$F'(z^*) = \begin{bmatrix} H'_x(z^*) & H'_y(z^*) & H'_w(z^*) \\ W^* & 0 & 0 \end{bmatrix}.$$

Da independência linear das linhas de $F'(z^*)$ segue que a matriz

$$\begin{bmatrix} H'_y(z^*) & H'_w(z^*) \end{bmatrix}$$

tem posto linha completo.

Como H e F possuem primeira derivada contínua, existe $\varepsilon > 0$ tal que, para todo z satisfazendo $\|z - z^*\| \leq \varepsilon$,

$$F'(z) \text{ e } H'_{yw}(z) \equiv \begin{bmatrix} H'_y(z) & H'_w(z) \end{bmatrix} \text{ tem posto completo.} \quad (2.31)$$

Além disso, (2.31) implica que $\|F'(z)^\dagger\|$ e $\|H'_{yw}(z)^\dagger\|$ são limitadas para todo z tal que $\|z - z^*\| \leq \varepsilon$.

Para um ponto $z = (a^\top, b^\top, c^\top)^\top$ arbitrário, com $a > 0$, $c > 0$ tal que $\|z - z^*\| \leq \varepsilon$ e $\mu \geq 0$, $\mu \in \mathbb{R}^n$, definimos x , y e w de forma que $(x - a, y - b, w - c)$ é a solução de mínima norma do sistema:

$$\begin{cases} H'_x(a, b, c)(x - a) + H'_y(a, b, c)(y - b) + H'_w(a, b, c)(w - c) = -H(a, b, c), \\ C(x - a) + A(w - c) = -Ca + \mu, \end{cases} \quad (2.32)$$

em que A e C denotam as matrizes diagonais cujas entradas são os elementos de a e c , respectivamente. Note que $\|a\| = \|a - x^*\| \leq \varepsilon$ e $\|c - w^*\| \leq \varepsilon$.

Para simplificar a notação, apesar de x, y, w serem funções de a, b, c , e μ , não deixaremos esta dependência explícita.

De (2.32) e da limitação de $\|F'(a, b, c)^\dagger\|$, segue que

$$\lim_{(a,b,c,\mu) \rightarrow (x^*, y^*, w^*, 0)} \|x - a\| = \lim_{(a,b,c,\mu) \rightarrow (x^*, y^*, w^*, 0)} \|w - c\| = \lim_{(a,b,c,\mu) \rightarrow (x^*, y^*, w^*, 0)} \|y - b\| = 0. \quad (2.33)$$

Então,

$$\lim_{(a,b,c,\mu) \rightarrow (x^*,y^*,w^*,0)} (x, w) = (x^*, w^*) = (0, w^*). \quad (2.34)$$

Por (2.32), simplificando a notação, temos que

$$\begin{bmatrix} H'_x & H'_y & H'_w \\ C & 0 & A \end{bmatrix} \begin{bmatrix} x - a \\ y - b \\ w - c \end{bmatrix} = \begin{bmatrix} -H \\ -Ca + \mu \end{bmatrix} \in \mathbb{R}^{r+n}. \quad (2.35)$$

Tomando a solução de mínima norma do sistema (2.35), temos que o vetor $(x-a, y-b, w-c)^\top$ pertence ao espaço coluna de $F'(z)^\top$. Assim, existem $q \in \mathbb{R}^r$ e $t \in \mathbb{R}^n$ tais que

$$\begin{bmatrix} x - a \\ y - b \\ w - c \end{bmatrix} = \begin{bmatrix} (H'_x)^\top & C \\ (H'_y)^\top & 0 \\ (H'_w)^\top & A \end{bmatrix} \begin{bmatrix} q \\ t \end{bmatrix} \in \mathbb{R}^{n+m+n}. \quad (2.36)$$

Logo,

$$\begin{cases} x - a = (H'_x)^\top q + Ct \\ y - b = (H'_y)^\top q \\ w - c = (H'_w)^\top q + At \end{cases} \quad (2.37)$$

Então, por (2.35) e (2.37),

$$\begin{bmatrix} H'_x & H'_y & H'_w \\ C & 0 & A \end{bmatrix} \begin{bmatrix} (H'_x)^\top q + Ct \\ (H'_y)^\top q \\ (H'_w)^\top q + At \end{bmatrix} = \begin{bmatrix} -H \\ -Ca + \mu \end{bmatrix}$$

o que implica em

$$\begin{bmatrix} H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top & H'_xC + H'_wA \\ C(H'_x)^\top + A(H'_w)^\top & C^2 + A^2 \end{bmatrix} \begin{bmatrix} q \\ t \end{bmatrix} = \begin{bmatrix} -H \\ -Ca + \mu \end{bmatrix}. \quad (2.38)$$

Portanto,

$$t = -(C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top)q - (C^2 + A^2)^{-1}(Ca - \mu). \quad (2.39)$$

Pela primeira equação de (2.38) e (2.39) temos que

$$\begin{aligned} & (H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top \\ & - (H'_xC + H'_wA)(C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top))q \\ & = -H + (H'_xC + H'_wA)(C^2 + A^2)^{-1}(Ca - \mu). \end{aligned} \quad (2.40)$$

Note que,

$$(C^2 + A^2)^{-1} = C^{-1}(I + C^{-1}A^2C^{-1})^{-1}C^{-1} \quad (2.41)$$

e defina

$$\tilde{H}' = H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top - (H'_xC + H'_wA)(C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top).$$

Então, por (2.41),

$$\begin{aligned}\tilde{H}' &= H'_x(H'_x)^\top + H'_y(H'_y)^\top + H'_w(H'_w)^\top \\ &\quad - (H'_x + H'_wAC^{-1})(I + C^{-1}A^2C^{-1})^{-1}((H'_x)^\top + C^{-1}A(H'_w)^\top).\end{aligned}\quad (2.42)$$

De (2.42), como $A \rightarrow 0$, temos que $\tilde{H}' \rightarrow H'_y(z^*)(H'_y(z^*))^\top + H'_w(z^*)(H'_w(z^*))^\top$.

Como a matriz $\begin{bmatrix} H'_y & H'_w \end{bmatrix}$ tem posto linha completo, temos que, se (a, b, c) estão suficientemente próximos de z^* , \tilde{H}' é não singular e sua inversa é limitada. Então, lembrando que por (2.40),

$$q = (\tilde{H}')^{-1}(-H + (H'_xC + H'_wA)(C^2 + A^2)^{-1}(Ca - \mu)), \quad (2.43)$$

obtemos que q é limitada desde que (a, b, c) esteja suficientemente próximo de uma solução e μ esteja próximo de 0. Além disso, como $Ca - \mu \rightarrow 0$, temos que $q(a, b, c, \mu)$ tende a zero quando (a, b, c) tende a z^* e μ tende a zero.

Em outras palavras,

$$\lim_{(a,b,c,\mu) \rightarrow (x^*,y^*,w^*,0)} q(a, b, c, \mu) = 0. \quad (2.44)$$

Analogamente, por (2.39),

$$\lim_{(a,b,c,\mu) \rightarrow (x^*,y^*,w^*,0)} t(a, b, c, \mu) = 0. \quad (2.45)$$

Lembrando que $x - a = (H'_x)^\top q + Ct$, por (2.39), temos que

$$\begin{aligned}Ct &= -C((C^2 + A^2)^{-1}(C(H'_x)^\top + A(H'_w)^\top)q - (C^2 + A^2)^{-1}(Ca - \mu)) \\ &= -CC^{-1}(I + C^{-1}A^2C^{-1})^{-1}C^{-1}C((H'_x)^\top + C^{-1}A(H'_w)^\top)q \\ &\quad -CC^{-1}(I + C^{-1}A^2C^{-1})^{-1}C^{-1}C(a - C^{-1}\mu) \\ &= -(I + C^{-1}A^2C^{-1})^{-1}((H'_x)^\top + C^{-1}A(H'_w)^\top)q \\ &\quad - (I + C^{-1}A^2C^{-1})^{-1}(a - C^{-1}\mu)\end{aligned}\quad (2.46)$$

e

$$\begin{aligned}x &= (H'_x)^\top q + Ct + a \\ &= (I - (I + C^{-1}A^2C^{-1})^{-1})a + (I - (I + C^{-1}A^2C^{-1})^{-1})(H'_x)^\top q \\ &\quad - (I + C^{-1}A^2C^{-1})^{-1}(C^{-1}A(H'_w)^\top)q + (I + C^{-1}A^2C^{-1})^{-1}C^{-1}\mu.\end{aligned}\quad (2.47)$$

Observe que

$$\begin{aligned}I - (I + C^{-1}A^2C^{-1})^{-1} &= I - I - \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j \\ &= C^{-1}A^2C^{-1}(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j).\end{aligned}$$

Então por (2.47),

$$\begin{aligned}x &= C^{-1}A^2C^{-1}(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a \\ &\quad + C^{-1}A^2C^{-1}(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H'_x)^\top q) \\ &\quad - AC^{-1}(I + C^{-1}A^2C^{-1})^{-1}((H'_w)^\top)q + (I + C^{-1}A^2C^{-1})^{-1}C^{-1}\mu.\end{aligned}$$

Segue que, para todo $i = 1, \dots, n$, temos

$$\begin{aligned} x_i \geq & (c_i)^{-2}(a_i)^2[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\ & + (c_i)^{-2}(a_i)^2[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H'_x)^\top q)]_i \\ & - (c_i)^{-1}a_i[(I + C^{-1}A^2C^{-1})^{-1}((H'_w)^\top q)]_i. \end{aligned} \quad (2.48)$$

Vamos agora analisar o comportamento do passo α_k^{break} . No Passo 3 do algoritmo PGUN buscamos o maior valor de α tal que $a + \alpha(x - a) \geq 0$ e $c + \alpha(w - c) \geq 0$. Para definir esse valor, vamos investigar os possíveis valores de $\alpha \in [0, 1]$ tais que

$$\alpha x_i + (1 - \alpha)a_i = 0 \quad (2.49)$$

ou

$$\alpha w_i + (1 - \alpha)c_i = 0. \quad (2.50)$$

Se ocorre (2.50), então

$$\alpha = \frac{c_i}{c_i - w_i}. \quad (2.51)$$

No entanto, de (2.33) e $w_i^* > 0$, se ε é suficientemente pequeno, não existe $\alpha \in [0, 1]$ satisfazendo (2.51).

Logo, precisamos analisar apenas os valores de α que satisfazem (2.49), isto é,

$$\alpha = 1 + \alpha \frac{x_i}{a_i}.$$

Então, por (2.48),

$$\begin{aligned} \alpha \geq & 1 + \alpha \frac{(c_i)^{-2}(a_i)^2}{a_i} [(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\ & + \frac{(c_i)^{-2}(a_i)^2}{a_i} [(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H'_x)^\top q)]_i \\ & - \frac{(c_i)^{-1}a_i}{a_i} [(I + C^{-1}A^2C^{-1})^{-1}(H'_w)^\top q]_i. \end{aligned}$$

Segue que,

$$\begin{aligned} \alpha \geq & 1 + \alpha(c_i)^{-2}a_i[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)a]_i \\ & + \alpha(c_i)^{-2}a_i[(I + \sum_{j=1}^{\infty}(-1)^j(C^{-1}A^2C^{-1})^j)((H'_x)^\top q)]_i \\ & - \alpha c_i[(I + C^{-1}A^2C^{-1})^{-1}(H'_w)^\top q]_i. \end{aligned} \quad (2.52)$$

Logo, por (2.44) dado qualquer $\delta \in [0, 1)$, tomando ε suficientemente pequeno, temos $\alpha = 1$. Consequentemente $\alpha \geq \delta$, como queríamos provar. \square

No próximo teorema vamos mostrar que se o Jacobiano de F tem posto completo em um ponto de acumulação z^* , então toda a sequência $\{z^k\}$, $k \in \mathbb{N}$, converge para este ponto z^* . Como consequência mostramos que, eventualmente, $\alpha_k = \alpha_k^{\text{max}}$, o que significa que o primeiro ponto testado no Passo 5 do algoritmo PGUN é aceito, uma vez que a norma de F decresce conforme requerido por (2.28). Antes de enunciar o teorema apresentamos a hipótese que será assumida:

Hipótese L Seja Ω' um conjunto aberto contendo Ω . Para todo $z, z' \in \Omega'$,

$$\|F'(z) - F'(z')\| \leq L\|z' - z\| \text{ para todos } z, z' \in \Omega' \subset \mathbb{R}^{n+m+n}. \quad (2.53)$$

Da hipótese acima segue que,

$$\begin{aligned} \|F(z') - F(z) - F'(z)(z' - z)\| &= \left\| \int_0^1 [F'(z + t(z' - z)) - F'(z)](z' - z) dt \right\| \\ &\leq \int_0^1 \|F'(z + t(z' - z)) - F'(z)\| \|z' - z\| dt \\ &\leq \|z' - z\|^2 \int_0^1 L|t| dt = \frac{L}{2} \|z' - z\|^2. \end{aligned} \quad (2.54)$$

Teorema 2.6. Assuma que vale a Hipótese L, que $z^* \in \Omega$ é um ponto de acumulação da sequência gerada pelo algoritmo PGUN, que $F'(z^*)$ tem posto completo e que, para k suficientemente grande, escolhemos

$$z^{k+1} = z^k + \alpha_k d^k \quad (2.55)$$

no Passo 6 do algoritmo. Além disso, assumamos que c_{big} (usado no Passo 2 do algoritmo PGUN) é maior do que $4\|F'(z^*)^\dagger\|$ e $\lim_{k \rightarrow \infty} \tau_k = 1$. Então,

$$\lim_{k \rightarrow \infty} z^k = z^*$$

e

$$\alpha_k = \alpha_k^{max} \quad (2.56)$$

para k suficientemente grande.

Demonstração. Seja K_1 uma subsequência infinita de índices tal que $\lim_{k \in K_1} z^k = z^*$. Pelo Teorema 2.4, z^* é um ponto estacionário de f restrita ao conjunto Ω .

Da escolha de d^k no Passo 2 do algoritmo, segue que

$$H'(z^k)d^k + H(z^k) = 0 \quad (2.57)$$

e

$$(x_i^k [d_w^k]_i + w_i^k [d_x^k]_i + x_i^k w_i^k)^2 = \sigma_k^2 \frac{\langle x^k, w^k \rangle^2}{n^2} \leq \sigma_k^2 \frac{\sum_{i=1}^n (x_i^k w_i^k)^2}{n}.$$

Então,

$$\sum_{i=1}^n (x_i^k [d_w^k]_i + w_i^k [d_x^k]_i + x_i^k w_i^k)^2 \leq \sigma_k^2 \sum_{i=1}^n (x_i^k w_i^k)^2 \leq \sigma_k^2 \|F(z^k)\|^2.$$

Note que,

$$\begin{aligned} F'(z^k)d^k + F(z^k) &= \begin{bmatrix} H'_x(z^k) & H'_y(z^k) & H'_w(z^k) \\ W^k & 0 & X^k \end{bmatrix} \begin{bmatrix} d_x^k \\ d_y^k \\ d_w^k \end{bmatrix} + \begin{bmatrix} H(z^k) \\ x_1^k w_1^k \\ \vdots \\ x_n^k w_n^k \end{bmatrix} \\ &= \begin{bmatrix} H'(z^k)d^k + H(z^k) \\ w_1^k [d_k^k]_1 + x_1^k [d_w^k]_1 + x_1^k w_1^k \\ \vdots \\ w_n^k [d_k^k]_n + x_n^k [d_w^k]_n + x_n^k w_n^k \end{bmatrix}. \end{aligned}$$

Logo, por (2.57),

$$\|F'(z^k)d^k + F(z^k)\| \leq \sigma_k \|F(z^k)\|. \quad (2.58)$$

Como $F'(z^*)$ tem posto linha completo, existe $\varepsilon_1 > 0$ tal que $\|F'(z)^\dagger\| \leq M_1 \equiv 2\|F'(z^*)^\dagger\|$ e $F'(z)$ tem posto linha completo sempre que $\|z - z^*\| \leq \varepsilon_1$.

Além disso, $F'(z)^\dagger F'(z) F'(z)^\dagger = F'(z)^\dagger$. Portanto, por (2.58), para $k \in K_1$ suficientemente grande e $\|z^k - z^*\| \leq \varepsilon_1$,

$$\begin{aligned} \|d^k\| &= \left\| F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix} \right\| \\ &= \left\| F'(z^k)^\dagger F'(z^k) F'(z^k)^\dagger \begin{bmatrix} -H(z^k) \\ -X_k W_k e + \mu^k \end{bmatrix} \right\| \\ &= \|F'(z^k)^\dagger F'(z^k) d^k\| \leq \|F'(z^k)^\dagger\| \|F'(z^k) d^k + F(z^k) - F(z^k)\| \\ &\leq \|F'(z^k)^\dagger\| (\|F'(z^k) d^k + F(z^k)\| + \|F(z^k)\|) \leq M_1(1 + \sigma_k) \|F(z^k)\|. \end{aligned} \quad (2.59)$$

Pelo Teorema 2.4, temos que $F(z^*) = 0$. Além disso, como $c_{big} \geq 4\|F'(z^*)^\dagger\|$, e se $\|z^k - z^*\| \leq \varepsilon_1$, $k \in K_1$ suficientemente grande, temos que $\|F(z^k)\| \leq 1$, segue que

$$\begin{aligned} \|d^k\| &\leq 2\|F'(z^*)^\dagger\|(1 + \sigma_k)\|F(z^k)\| \\ &\leq 4\|F'(z^*)^\dagger\|\|F(z^k)\| \leq c_{big}\|F(z^k)\| \leq c_{big} \end{aligned}$$

ou seja, d^k é calculada no Passo 2.

Defina $M_2 = 2\|F'(z^*)\|$. Então, como $F(z^*) = 0$, por (2.59) e pelo Teorema 2.5 existe $\varepsilon_2 \in (0, \varepsilon_1]$ tal que, para todo $k \in \mathcal{N}$ satisfazendo $\|z^k - z^*\| \leq \varepsilon_2$, temos

- (i) $\|d^k\| \leq M_1(1 + \sigma_k)\|F(z^k)\|$;
- (ii) $\alpha_k^{max} \geq \max\{1 - \frac{1}{12M_1M_2}, \frac{11}{12}\}$;
- (iii) $\|F'(z^k)\| \leq M_2$;
- (iv) $\|F(z^k)\| \leq \frac{1}{12LM_1^2}$;
- (v) $\rho\|\alpha_k^{max}d^k\|^2 \leq \frac{1}{2}\|F(z^k)\|$.

Então, para todo $k \in \mathcal{N}$ tal que $\|z^k - z^*\| \leq \varepsilon_2$,

$$\begin{aligned} &\|F(z^k + \alpha_k^{max}d^k)\| \\ &\leq \|F(z^k + \alpha_k^{max}d^k) - F(z^k) - \alpha_k^{max}F'(z^k)d^k\| + \|F(z^k) + \alpha_k^{max}F'(z^k)d^k\| \\ &\leq \frac{L}{2}(\alpha_k^{max})^2\|d^k\|^2 + \|F(z^k) + F'(z^k)d^k\| + (1 - \alpha_k^{max})\|F'(z^k)d^k\| \\ &\leq \frac{L}{2}(\alpha_k^{max})^2\|d^k\|^2 + \sigma_k\|F(z^k)\| + (1 - \alpha_k^{max})\|F'(z^k)d^k\| \\ &\leq \frac{L}{2}(\alpha_k^{max})^2M_1^2(1 + \sigma_k)^2\|F(z^k)\|^2 + \sigma_k\|F(z^k)\| \\ &\quad + (1 - \alpha_k^{max})\|F'(z^k)\|M_1(1 + \sigma_k)\|F(z^k)\| \\ &\leq \left(\frac{L}{2}(\alpha_k^{max})^2M_1^2\|F(z^k)\| + \sigma_k + (1 - \alpha_k^{max})(1 + \sigma_k)M_2M_1\right)\|F(z^k)\| \\ &\leq \frac{1}{2}\|F(z^k)\| \leq \|F(z^k)\| - \rho\|\alpha_k^{max}d^k\|^2 + \gamma_k. \end{aligned} \quad (2.60)$$

Logo, por (2.28), para todo $k \in \mathbb{N}$ tal que $\|z^k - z^*\| \leq \varepsilon_2$, temos que $\alpha_k = \alpha_k^{max}$ (o que prova (2.56)),

$$z^{k+1} = z^k + \alpha_k^{max} d^k \quad \text{e} \quad \|F(z^{k+1})\| \leq \frac{1}{2} \|F(z^k)\|. \quad (2.61)$$

Como $\lim_{k \in K_1} F(z^k) = F(z^*) = 0$, existe $k_0 \in K_1$ tal que $\|z^{k_0} - z^*\| \leq \frac{\varepsilon_2}{4}$ e $\|F(z^{k_0})\| \leq \frac{\varepsilon_2}{4(2M_1+1)}$.

Vamos provar, por indução, que $\|z^k - z^*\| \leq \varepsilon_2$ para todo $k \geq k_0$.

Para $k = k_0$ o resultado é trivial. Assuma, como hipótese de indução, que $\|z^k - z^*\| \leq \varepsilon_2$ para todo $k = k_0, k_0 + 1, \dots, k_0 + j - 1$. Então, por (2.61), $\|F(z^{k+1})\| \leq \frac{1}{2} \|F(z^k)\|$ para $k = k_0 + 1, \dots, k_0 + j - 1$.

De (2.61) e (i)-(v), segue que

$$\begin{aligned} \|z^{k_0+j} - z^{k_0}\| &= \left\| \sum_{i=0}^{j-1} \alpha_{k_0+i}^{max} d^{k_0+i} \right\| \leq 2M_1 \sum_{i=0}^{j-1} \left(\frac{1}{2}\right)^i \|F(z^{k_0})\| \\ &\leq 4M_1 \|F(z^{k_0})\| \leq \frac{\varepsilon_2}{4}. \end{aligned}$$

Logo, $\|z^{k_0+j} - z^*\| \leq \|z^{k_0+j} - z^{k_0}\| + \|z^{k_0} - z^*\| \leq \frac{\varepsilon_2}{2}$.

Portanto $\|z^{k_0+j} - z^*\| \leq \varepsilon_2$ e, pelo Princípio de Indução, o resultado segue.

Para concluir a demonstração, vamos mostrar que $\{z^k\}$ é sequência de Cauchy.

Sejam $j \geq k_0$ e $\ell \geq 1$. Então,

$$\begin{aligned} \|z^{j+\ell} - z^j\| &\leq \sum_{i=0}^{\ell-1} \alpha_{j+i}^{max} \|d^{j+i}\| \\ &\leq 2M_1 \sum_{i=0}^{\ell-1} \left(\frac{1}{2}\right)^{i+1} \|F(z^j)\| \\ &\leq 2M_1 \sum_{i=0}^{\ell-1} \left(\frac{1}{2}\right)^{i+1} \|F(z^j)\| \leq 2M_1 \|F(z^j)\|. \end{aligned} \quad (2.62)$$

Como $\lim_{j \rightarrow \infty} \|F(z^j)\| = 0$, (2.62) implica que $\{z^k\}$ é sequência de Cauchy. Portanto, como z^* é ponto de acumulação, temos que $\lim_{k \rightarrow \infty} z^k = z^*$. \square

Na sequência, vamos mostrar que, sob as mesmas hipóteses do Teorema 2.6, e com escolhas adequadas dos parâmetros σ_k , o algoritmo PGUN exhibe convergência superlinear ou quadrática.

Considere a seguinte hipótese sobre os parâmetros σ_k :

Hipótese S. Escolha σ_k tal que

$$\lim_{k \rightarrow \infty} \sigma_k = 0. \quad (2.63)$$

Teorema 2.7. Assuma que $\{z^k\}$ é sequência gerada pelo algoritmo PGUN e converge para um ponto z^* tal que $F(z^*) = 0$, $F'(z^*)$ tem posto completo e para k suficientemente grande escolhemos

$$z^{k+1} = z^k + \alpha_k d^k \quad (2.64)$$

no Passo 6 do algoritmo. Assuma que as hipóteses do Teorema 2.6 e as Hipóteses L e S são satisfeitas. Então, z^k converge superlinearmente para z^* . Além disso, se existem $c_1, c_2 > 0$ tais que, para todo k suficientemente grande,

$$\sigma_k \leq c_1 \|F(z^k)\| \quad \text{e} \quad |\tau_k - 1| \leq c_2 \|F(z^k)\|, \quad (2.65)$$

z^k converge quadraticamente para z^* .

Demonstração. Como $\tau_k \rightarrow 1$ temos que $\lim_{k \rightarrow \infty} \alpha_k^{max} = 1$.

Pelo Teorema 2.6, para todo k suficientemente grande, existe $M > 0$ tal que $\|d^k\| \leq M\|F(z^k)\|$, $\|F'(z^k)\| \leq M$ e

$$\begin{aligned}
\|F(z^{k+1})\| &\leq \|F(z^{k+1}) - F(z^k) - \alpha_k^{max} F'(z^k) d^k\| + \|F(z^k) + \alpha_k^{max} F'(z^k) d^k\| \\
&\leq \frac{L}{2} (\alpha_k^{max})^2 \|d^k\|^2 + \|F(z^k) + F'(z^k) d^k\| + (1 - \alpha_k^{max}) \|F'(z^k) d^k\| \\
&\leq \frac{L}{2} M^2 \|F(z^k)\|^2 + \sigma_k \|F(z^k)\| + (1 - \alpha_k^{max}) M^2 \|F(z^k)\| \\
&\leq \left(\frac{L}{2} M^2 \|F(z^k)\| + \sigma_k + (1 - \alpha_k^{max}) M^2 \right) \|F(z^k)\| = R_k \|F(z^k)\|
\end{aligned} \tag{2.66}$$

em que

$$R_k = \frac{L}{2} M^2 \|F(z^k)\| + \sigma_k + M^2 (1 - \alpha_k^{max}).$$

Além disso,

$$\|z^{k+1} - z^*\| \leq \sum_{j=k+1}^{\infty} \alpha_j^{max} \|d^j\| \leq 2M \sum_{j=1}^{\infty} \left(\frac{1}{2}\right)^j \|F(z^{k+1})\|.$$

Por (2.66) e (2.54) temos

$$\begin{aligned}
\|z^{k+1} - z^*\| &\leq 2MR_k \|F(z^k)\| \\
&= 2MR_k \|F(z^k) - F(z^*) - F'(z^k)(z^k - z^*) + F'(z^k)(z^k - z^*)\| \\
&\leq 2MR_k \|F(z^k) - F(z^*) - F'(z^k)(z^k - z^*)\| + \|F'(z^k)(z^k - z^*)\| \\
&\leq 2MR_k \left(\frac{L}{2} \|z^k - z^*\| + M \right) \|z^k - z^*\| \\
&\leq 2MR_k L \left(\frac{L}{2} + M \right) \|z^k - z^*\|.
\end{aligned}$$

Como $\lim_{k \rightarrow \infty} R_k = 0$, segue que z^k converge superlinearmente para z^* .

Agora, tomando $c = \max\{c_1, c_2\}$, como $\max\{\sigma_k, 1 - \alpha_k^{max}\} \leq \max\{\sigma_k, 1 - \tau_k\} \leq c\|F(z_k)\|$, para k suficientemente grande, temos

$$\begin{aligned}
\|z^{k+1} - z^*\| &\leq 2M \left(\frac{L}{2} M^2 \|F(z^k)\| + \sigma_k + (1 - \alpha_k^{max}) M^2 \right) \|F(z^k)\| \\
&\leq 2M \left(\frac{L}{2} M^2 + (1 + M^2)c \right) \|F(z^k)\|^2 \\
&\leq 2M \left(\frac{L}{2} + M \right)^2 \left(\frac{L}{2} M^2 + (1 + M^2)c \right) \|z^k - z^*\|^2.
\end{aligned}$$

O que demonstra a convergência quadrática. □

2.4 Experimentos computacionais

Nesta seção, vamos relatar algumas experiências com o algoritmo PGUN para o cálculo de uma solução viável simples (2.10) e uma solução viável com meta c_t (2.12). Para termos uma ideia melhor da eficiência do algoritmo PGUN na prática, fizemos comparações do método PGUN com o algoritmo denominado Levenberg-Marquardt Projetado (PLM) [36]. Uma breve descrição deste último método é apresentada antes de relatarmos os experimentos numéricos.

2.4.1 O método de Levenberg-Marquardt Projetado

O método de Levenberg-Marquardt Projetado (PLM) é um algoritmo para obter a solução de sistemas não lineares com restrições $F(z) = 0$, $z \in Z$, com $Z \subseteq \mathbb{R}^n$ um conjunto não vazio, fechado e convexo. Para resolver este problema o método é aplicado a um sistema semelhante ao definido em (2.16) com a função de mérito definida por (2.15).

O algoritmo PLM gera uma sequência de pontos $\{z^k\}$ através da fórmula

$$z^{k+1} = P_Z(z^k + d_U^k) \quad k = 0, 1, \dots,$$

em que P_Z é a projeção ortogonal sobre Z , d_U^k é a única solução do sistema de equações lineares

$$(H_k^\top H_k + \mu_k I) d_U = -H_k^\top F(z^k), \quad (2.67)$$

e $H_k \in \mathbb{R}^{m \times n}$ é uma aproximação para o Jacobiano de $F(z^k)$.

Apresentamos a seguir, em termos gerais, o Algoritmo 3.12 de [36] que, ao longo deste trabalho, será referenciado como PLM. Mais detalhes sobre o método, incluindo os resultados de convergência podem ser encontrados em [36].

Algoritmo 2.3. Levenberg-Marquardt Projetado - PLM

Inicialização: Escolha $z^0 \in Z$, $\mu > 0$, $\beta, \sigma, \gamma \in (0, 1)$, $\rho > 0$ e $p > 1$.

Passo 1. *Critério de parada:* Se $F(z^k) = 0$, pare.

Passo 2. *Direção:* Escolha $H_k \in \mathbb{R}^{m \times n}$, faça $\mu_k = \mu \|F(z^k)\|^2$ e encontre d_U^k a solução de (2.67).

Passo 3. *Passo de Levenberg-Marquardt:* Se

$$\|F(P_Z(z^k + d_U^k))\| \leq \gamma \|F(z^k)\|, \quad (2.68)$$

então defina $z^{k+1} = P_Z(z^k + d_U^k)$ e vá para o Passo 1.

Passo 4. *Passo de busca linear:* Se a direção $s^k = P_Z(z^k + d_U^k) - z^k$ é uma direção de descida de f , isto é, se $\nabla f(z^k)^\top s^k \leq -\rho \|s^k\|^p$, defina $\alpha = 1$ e

Passo 4.1. Se

$$\|F(z^k + t s^k)\|^2 \leq \|F(z^k)\|^2 + \gamma \alpha \nabla f(z^k)^\top s^k$$

defina $z^{k+1} = z^k + \alpha s^k$ e vá para o Passo 1.

Passo 4.2. Escolha $\alpha_{new} \in (0, \alpha)$, faça $\alpha = \alpha_{new}$ e retorne ao Passo 4.1.

Passo 5. *Passo de Gradiente Projetado:* Calcule um tamanho de passo $\alpha_k = \max\{\beta^l : l = 0, 1, 2, \dots\}$ tal que

$$f(z^k(\alpha_k)) \leq f(z^k) + \sigma \nabla f(z^k)^\top (z^k(\alpha_k) - z^k),$$

em que $z^k(\alpha) = P_Z(z^k - \alpha \nabla f(z^k))$. Defina $z^{k+1} = z^k(\alpha_k)$ e vá para o Passo 1.

2.4.2 Implementação e experimentos numéricos

Os códigos para os algoritmos PGUN e PLM foram escritos em Fortran 77 com dupla precisão. Os experimentos foram realizados usando gfortran-4.6 em uma máquina Intel CORE I3-2310M @ 2.10 GHz com 100 Gb de HD e 4Gb de Ram. Para resolver os sistemas lineares esparsos utilizamos a rotina `ma48` disponibilizada pela Harwell Subroutine Library [26]. Os critérios de parada considerados estão colocados a seguir.

SC1: Parar com z^k se $\|g(z^k, \eta)\| < 10^{-5}$.

Em nossos experimentos percebemos que se o algoritmo termina em uma iteração de pontos interiores do tipo Newton (IP), é possível continuar o processo para conseguir uma solução mais precisa. Isso nos levou a implementar um segundo critério de parada, a saber:

SC2: Parar com z^k quando SC1 foi satisfeito e $\|F(z^k)\| < 10^{-6}$.

No entanto, se durante esse processo for necessário realizar uma iteração de gradiente projetado, retornamos com o ponto que satisfaz **SC1**.

Em alguns casos o algoritmo PGUN converge lentamente, usando iterações de gradientes projetados, para um ponto estacionário cujo valor de função de mérito (2.15) é positivo. Neste caso, o algoritmo PGUN não está convergindo para uma solução do problema e não há motivos para continuar o processo iterativo. Para evitar esse fato, decidimos parar o algoritmo prematuramente através do terceiro critério de parada:

SC3: Parar na iteração k se $\|F(z^k)\| > 10^{-3}$ e $\|F(z^{k-1})\| - \|F(z^k)\| < 10^{-4}$.

Além disso, o algoritmo PGUN é reiniciado com um ponto inicial diferente quando **SC3** for satisfeito.

O algoritmo PLM utiliza iterações de Levenberg-Marquardt (LM) e gradientes projetados (PG) e usa os mesmos critérios de parada SCi, $i = 1, 2, 3$, utilizados por PGUN com as iterações LM substituindo as iterações IP.

Limitamos o número de iterações, para ambos os algoritmos, em $\max\{100, \min\{p + 1, 2n + m\}^3\}$ e o tempo computacional em 600 segundos. O iterado inicial para ambos os métodos foi:

$$x^0 = e, \quad y^0 = 0, \quad w^0 = e \quad (2.69)$$

em que e é o vetor unitário, isto é, $e = (1, \dots, 1)^\top$. Os seguintes valores foram utilizados para as constantes do algoritmo PGUN: $\alpha_{min} = 10^{-8}$, $\beta = 0.25$, $c_{big} = 10^4$, $c_{small} = 10^{-10}$, $\eta_k = \eta = 1.0$, $\gamma_k = \frac{1}{k^2}$, $\rho = 10^{-3}$, $\sigma_k = \sigma = \frac{1}{\sqrt{2n+m}}$, $\tau_k = \tau = 0.9995$ e $\theta = 0.5$. Para o método PLM utilizamos os valores citados em [36]: $\alpha_{min} = 10^{-12}$, $\beta = 0.9$, $\mu = 10^{-5}$, $\sigma = 10^{-4}$, $\gamma = 0.99995$, $p = 2.1$ e $\rho = 10^{-8}$.

Para os experimentos numéricos utilizamos 48 MPCCs da coleção MacMPEC [40]. Esses problemas são apresentados na Tabela 2.1. Nesta tabela, m é a dimensão de y , n é a dimensão de x e w , \mathbf{p} é a dimensão de $(\varphi(x, y, w), H(x, y, w))^\top$, \mathbf{nz} é o número de possíveis elementos não nulos da matriz Jacobiana de F , $\mathbf{densidade}$ é a densidade da matriz Jacobiana e \mathbf{min} é o menor valor de função conhecido.

Tabela 2.1: Problemas selecionados da coleção MacMPEC.

Problema	m	n	p	nz	densidade	min	Problema	m	n	p	nz	densidade	min
bard1	0	6	5	29	22%	17.0000	liswet1	52	102	104	760	1%	0.01399
bard2	0	22	18	98	6%	-6598.00	nash1	2	7	7	35	16%	7.8e-30
bard3	0	8	6	38	17%	-12.6787	outrata31	0	7	6	37	20%	3.20770
bilevel1	2	12	12	62	10%	-60.0000	outrata32	0	7	6	38	21%	3.44940
bilevel3	2	8	8	44	15%	-12.6787	outrata33	0	7	6	38	21%	4.60425
bilin	0	10	8	56	16%	18.4000	outrata34	0	7	6	40	22%	6.59268
dempe	2	2	3	12	40%	28.2500	portfl1	1	75	14	1149	9%	1.5e-05
design_cent1	9	7	13	60	13%	1.86065	qpec1	10	21	21	113	5%	80.0000
desilva	2	7	7	33	15%	-1.00000	qpecgen1	5	103	103	11124	26%	0.09900
df1	1	6	6	27	17%	0.00000	ralph2	0	2	1	7	58%	0.00000
ex911	2	7	8	42	18%	-13.0000	ralphmod	0	109	105	10831	23%	-683.033
ex921	0	7	6	34	19%	17.0000	scale1	0	2	1	7	58%	1.00000
ex922	0	9	7	38	13%	100.000	scale2	0	2	1	7	58%	1.00000
ex925	1	6	6	30	19%	5.00000	scale3	0	2	1	7	58%	1.00000
ex928	0	6	5	24	18%	1.50000	scale4	0	2	1	7	58%	1.00000
flp2	0	7	5	33	20%	0.00000	scale5	0	2	1	7	58%	100.000
gauvin	0	5	4	22	24%	20.0000	scholtes1	1	3	2	14	40%	2.00000
gnash1	1	11	11	57	11%	-230.823	scholtes2	1	3	2	14	40%	15.0000
hakonsen	0	9	7	46	16%	24.3668	scholtes3	0	2	1	7	58%	0.50000
jr1	1	2	2	10	50%	0.50000	scholtes4	1	4	3	18	29%	-3.0e-07
jr2	1	2	2	10	50%	0.50000	scholtes5	0	3	2	12	40%	1.00000
kth1	0	2	1	7	58%	0.00000	sl1	2	11	10	49	10%	0.00010
kth2	0	2	1	7	58%	0.00000	stackelberg1	0	4	3	16	29%	-3266.67
kth3	0	2	1	7	58%	0.50000	traffic1	0	739	737	3679	0.17%	45.1500

Experimento 1: Calculando uma solução viável simples

Para calcular uma solução viável simples para MPCC consideramos o problema horizontal (2.10). A Tabela 2.2 mostra o número de pares complementares de cada problema. Nesta tabela, NCP representa o número original de pares complementares e NNG o número de pares adicionados ao reescrevermos os problemas na forma (2.10).

Neste primeiro experimento o método PGUN resolveu 46 dos 48 problemas testados, alcançando 96% de sucesso, enquanto que o método PLM resolveu 39 problemas, atingindo 75% de sucesso. Tal comportamento pode ser observado na Figura 2.1, que expõe o número de problemas resolvidos por iteração para cada um dos dois métodos considerados.

Na Tabela 2.3 apresentamos os resultados obtidos com o algoritmo PGUN para o Experimento 1. Nesta tabela, utilizamos a notação:

TERM: a terminação do algoritmo, conforme segue:

IP-1: o algoritmo terminou com uma iteração de Newton (IP) satisfazendo o critério SC1;

IP-2: o algoritmo terminou com uma iteração IP satisfazendo o critério SC2;

PG-1: o algoritmo terminou com uma iteração de gradiente projetado satisfazendo o critério SC1.

IP: número de iterações de Newton;

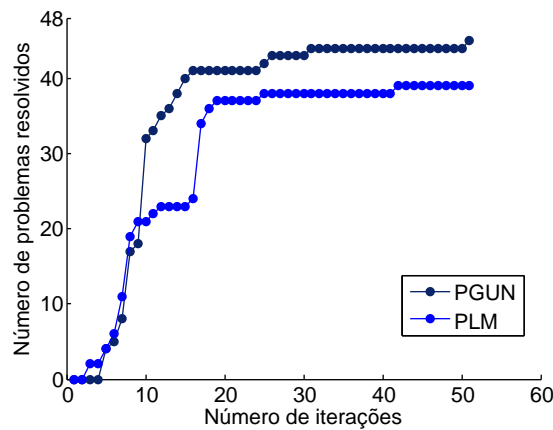


Figura 2.1: Número de problemas resolvidos por iteração para o Experimento 1.

Tabela 2.2: Número de pares complementares para o Experimento 1.

Problema	NCP	NNG	Problema	NCP	NNG	Problema	NCP	NNG
bard1	3	2	gauvin	2	2	qpecgen	100	2
bard2	4	17	gnash1	8	2	ralph2	1	0
bard3	2	5	hakonsen	4	4	ralphmod	100	8
bilevel1	6	5	jr1	1	0	scale1	1	0
bilevel3	4	3	jr2	1	0	scale2	1	0
bilin	6	3	kth1	1	0	scale3	1	0
dempe	1	0	kth2	1	0	scale4	1	0
design-cent1	3	3	kth3	1	0	scale5	1	0
desilva	2	4	liswet1-inv50	50	51	scholtes1	1	1
df1	1	4	nash1	2	4	scholtes2	1	1
ex911	5	1	outrata31	4	2	scholtes3	1	0
ex921	4	2	outrata32	4	2	scholtes4	1	2
ex922	4	4	outrata33	4	2	scholtes5	2	0
ex925	3	2	outrata34	4	2	sl1	3	7
ex928	2	3	portfl1	12	62	stackelberg1	1	2
flp2	2	4	qpec1	10	10	traffic1	244	494

PG: número de iterações de gradientes projetados;

CG: número de vezes que o algoritmo mudou de uma iteração de Newton para gradiente projetado e vice-versa;

NE: número de avaliações de função;

TEMPO: tempo computacional (em segundos) medido pela função `etime`;

RES: o resíduo encontrado, isto é, o valor de $\|F(\bar{z})\|$ em que \bar{z} é a solução calculada pelo algoritmo;

Tabela 2.3: Desempenho do método PGUN para o Experimento 1.

Problema	TERM	IP	PG	CG	NE	TEMPO	RES	SPG	norm	FEAS	COMP
bard1	IP-1	6	0	0	7	0.0000	1.17e-08	3.08e-08	1.16e-08	1.19e-09	
bard2	IP-2	12	0	0	13	0.0040	5.86e-14	4.23e-13	1.20e-14	5.73e-14	
bard3	IP-1	5	0	0	6	0.0000	4.72e-07	1.06e-06	2.88e-07	3.12e-07	
bilevel1	IP-2	25	0	0	26	0.0040	9.90e-07	2.97e-06	9.90e-07	1.36e-20	
bilevel3	IP-2	51	0	0	52	0.0040	9.64e-07	4.09e-06	9.64e-07	4.19e-23	
bilin	IP-1	7	0	0	8	0.0000	2.68e-08	8.96e-09	1.84e-09	2.48e-08	
dempe	IP-1	5	0	0	6	0.0000	1.25e-07	5.07e-07	1.25e-07	1.98e-12	
design-cent1	IP-2*	8	0	0	9	0.0000	6.80e-08	1.05e-08	4.51e-09	6.79e-08	
desilva	IP-1	5	0	0	6	0.0000	2.21e-07	6.11e-07	2.17e-07	2.94e-08	
df1	IP-2	10	0	0	11	0.0000	8.97e-08	1.26e-07	8.97e-08	2.72e-23	
ex911	IP-1	8	0	0	9	0.0000	6.08e-08	3.80e-07	5.08e-08	3.34e-08	
ex921	IP-2	31	0	0	32	0.0000	6.08e-07	1.74e-06	6.08e-07	1.97e-22	
ex922	IP-2	14	0	0	15	0.0000	4.84e-07	4.77e-10	1.13e-15	4.84e-07	
ex925	IP-1	8	0	0	9	0.0000	4.57e-07	7.93e-08	6.38e-16	4.57e-07	
ex928	IP-1	5	0	0	6	0.0000	2.16e-08	6.12e-09	9.91e-17	1.86e-08	
flp2	IP-1	7	0	0	8	0.0000	4.55e-13	1.06e-12	1.14e-15	4.54e-13	
gauvin	IP-1	8	0	0	9	0.0000	5.13e-07	2.43e-07	4.43e-15	5.13e-07	
gnash1	IP-1*	14	0	0	15	0.0000	4.42e-11	4.60e-11	4.42e-011	1.86e-17	
hakonsen	IP-1	9	0	0	10	0.0000	3.54e-11	4.26e-09	3.54e-11	3.45e-15	
jr1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	0.0000	9.53e-07	
jr2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	0.0000	9.53e-07	
kth1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
kth2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
kth3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
liswet1-inv50	IP-1	26	0	0	45	0.1400	3.21e-08	3.96e-08	2.82e-08	1.18e-08	
nash1	IP-1	8	0	0	9	0.0000	6.52e-07	7.94e-08	2.25e-15	6.52e-07	
outrata31	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09	
outrata32	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09	
outrata33	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09	
outrata34	IP-1	8	0	0	9	0.0000	1.06e-08	2.03e-08	3.79e-09	9.93e-09	
portfl1	IP-2*	2098	0	0	2100	5.6403	2.21e-09	3.19e-09	2.21e-09	4.98e-17	
qpec1	IP-2	12	0	0	13	0.0040	2.66e-07	9.20e-11	0.00000	5.96e-08	
qpecgen	**										
ralph2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
ralphmod	IP-1	16	0	0	17	0.8080	7.28e-09	1.64e-07	6.40e-11	6.95e-09	
scale1	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scale2	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scale3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scale4	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scale5	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scholtes1	PG-1	13	2	3	16	0.0000	6.40e-09	6.40e-09	6.40e-09	9.31e-15	
scholtes2	PG-1	13	2	3	16	0.0000	6.40e-09	6.40e-09	6.40e-09	9.31e-15	
scholtes3	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	-	9.53e-07	
scholtes4	IP-2	10	0	0	11	0.0000	9.53e-07	1.31e-09	1.61e-17	9.53e-07	
scholtes5	IP-2	11	0	0	12	0.0000	3.37e-07	2.32e-10	0.00000	2.38e-07	
sl1	IP-2	13	0	0	14	0.0000	4.09e-07	4.14e-10	1.94e-14	4.09e-07	
stackelberg1	IP-1	7	0	0	8	0.0000	3.49e-07	8.48e-06	3.40e-15	3.49e-07	
traffic1	**										

SPG_norm: norma do gradiente projetado escalado avaliado na solução calculada pelo algoritmo;

FEAS: medida de viabilidade, isto é, $FEAS = \|H(\bar{z})\|$;

COMP: medida de complementaridade, isto é, $COMP = \max_{i=1,n} \{x_i w_i\}$.

* o algoritmo calculou uma solução viável para MPCC usando uma aproximação inicial diferente da usual (2.69);

** o algoritmo não foi capaz de calcular uma solução para MPCC após 10 tentativas com diferentes aproximações iniciais.

Os resultados obtidos indicam que, em geral, o algoritmo PGUN convergiu rapidamente para uma solução do problema (2.10), uma vez que foram necessárias poucas iterações de Newton (IP). Note que, apenas para o problema *portfl1* o algoritmo PGUN precisou de um número maior de iterações. Além disso, iterações lentas de gradientes projetados só foram necessárias para dois dos problemas testados. Em três problemas o critério de parada SC3 foi utilizado para evitar que o algoritmo convergisse, de maneira lenta, para um ponto estacionário da função de mérito que não era uma solução do problema. Mas, para dois destes problemas, o algoritmo PGUN convergiu rapidamente para uma solução quando tentamos outro ponto inicial. Por fim, o método não conseguiu resolver dois dos problemas testados.

Vale ressaltar que os baixos valores de FEAS e COMP indicam que o algoritmo PGUN é, em geral, capaz de calcular uma solução viável e precisa para o problema com restrições de complementaridade. Além disso, o uso do critério de parada SC2 se mostrou apropriado.

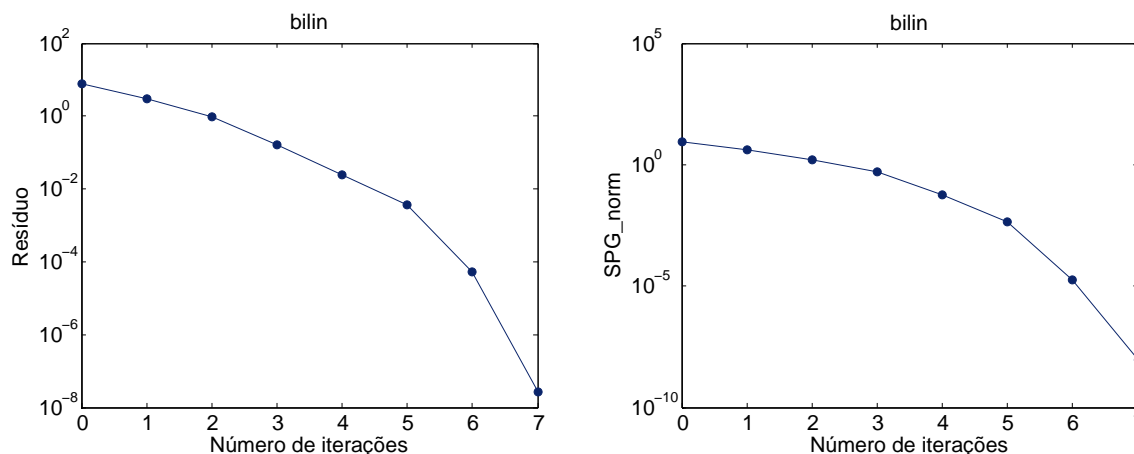


Figura 2.2: Gráficos de convergência do método PGUN para o problema *bilin*.

As Figuras 2.2, 2.3 e 2.4 exibem os gráficos de convergência do método PGUN para os problemas *bilin*, *qpec1* e *ralphmod*, respectivamente.

Com o objetivo de conhecer melhor o desempenho do algoritmo PGUN na prática, resolvemos os mesmos problemas com o método PLM. Os resultados alcançados por este último estão dispostos na Tabela 2.4, onde acrescentamos a notação:

TERM: a terminação do algoritmo, conforme segue:

LM-1: o algoritmo terminou com uma iteração de Levenberg-Marquardt (LM) satisfazendo o critério SC1;

LM-2: o algoritmo terminou com uma iteração LM satisfazendo o critério SC2;

PG-1: o algoritmo terminou com uma iteração de gradiente projetado (PG) satisfazendo o critério SC1.

LM: número de iterações de Levenberg-Marquardt (associado aos passos 2, 3 e 4).

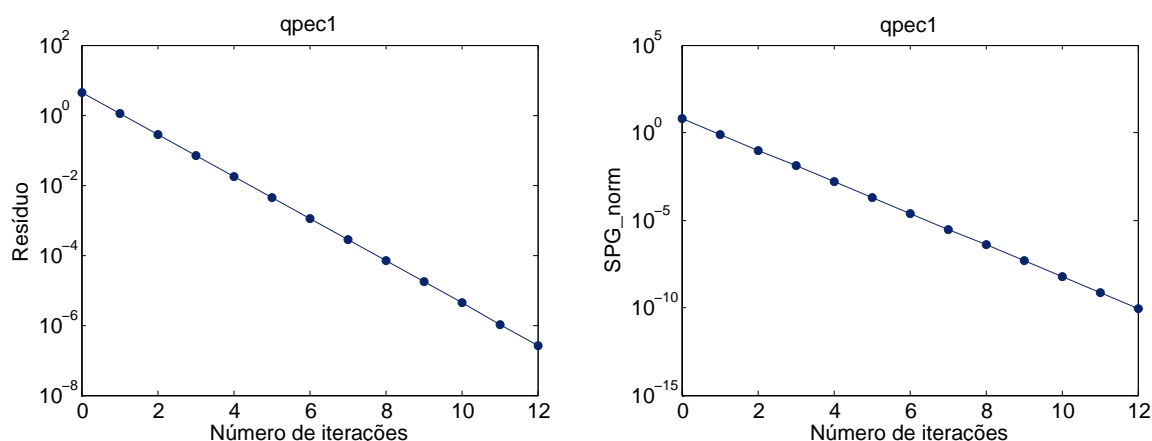


Figura 2.3: Gráficos de convergência do método PGUN para o problema *qpec1*.

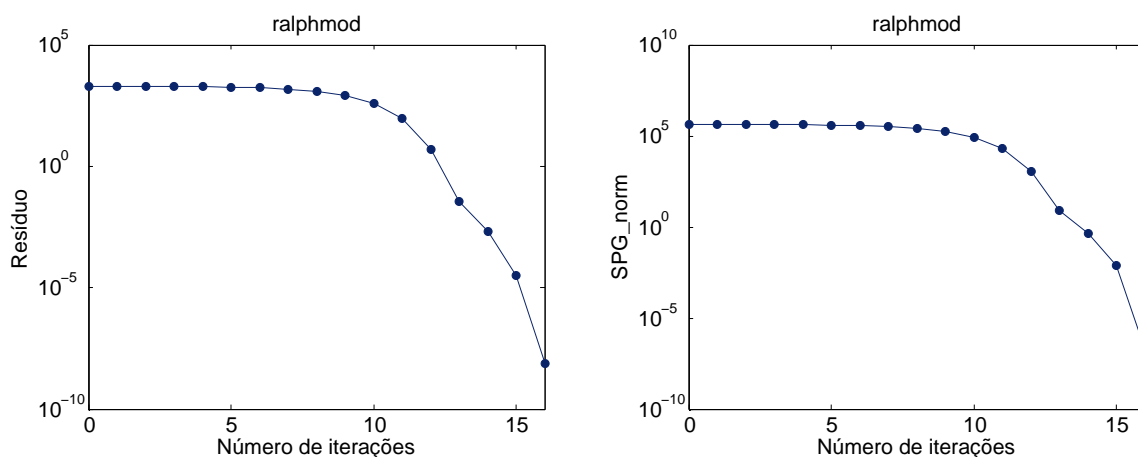


Figura 2.4: Gráficos de convergência do método PGUN para o problema *ralphmod*.

PG: número de iterações de gradientes projetados (passo 5).

Os resultados obtidos indicam que o método PLM utilizou poucas iterações de Levenberg-Marquardt para convergir e, além disso, direções lentas de gradientes projetados raramente foram necessárias. Bem como no caso anterior, o critério de parada SC3 foi empregado para evitar convergência para pontos que não são uma solução viável do MPCC. Assim como para o algoritmo PGUN, o critério SC2 permitiu encontrar soluções viáveis e precisas para o problema com restrições de complementaridade. Por fim, o método PLM aparenta ter mais dificuldades, devido ao maior número de insucessos.

As Figuras 2.5 e 2.6 apresentam os gráficos de convergência do método PLM para os problemas *bilin* e *qpec1*, respectivamente. Omitimos os gráficos de convergência para o problema *ralphmod* visto que o mesmo não convergiu. Em comparação aos gráficos de convergência do método PGUN (Figuras 2.2 e 2.3) percebemos que os dois algoritmos apresentaram um comportamento muito semelhante para os casos onde ambos obtiveram sucesso.

Tabela 2.4: Desempenho do método PLM para o Experimento 1.

Problema	TERM	LM	PG	CG	NE	TEMPO	RES	SPG	norm	FEAS	COMP
bard1	LM-1	3	0	0	4	0.0000	3.35e-09	1.02e-08	8.84e-16	3.35e-09	
bard2	LM-1*	11	1	2	23	0.0000	5.76e-08	2.09e-07	2.17e-11	4.44e-08	
bard3	LM-2	7	0	0	8	0.0000	7.25e-09	2.44e-08	5.67e-09	4.45e-09	
bilevel1	**										
bilevel3	LM-2	8	0	0	9	0.0000	6.74e-09	5.46e-09	1.63e-09	6.54e-09	
bilin	LM-1	7	0	0	8	0.0000	8.81e-12	4.40e-12	1.51e-16	8.81e-12	
dempe	LM-2	42	0	0	43	0.0000	6.13e-07	1.71e-06	4.33e-07	4.33e-07	
designcent1	LM-2*	9	0	0	10	0.0000	1.06e-10	3.25e-10	1.06e-10	7.45e-14	
desilva	LM-2	8	0	0	9	0.0000	1.91e-10	3.07e-10	1.83e-10	3.95e-11	
df1	LM-1	9	0	0	10	0.0000	2.02e-08	6.36e-08	1.93e-08	5.52e-09	
ex911	LM-2	6	0	0	126	0.0000	9.56e-10	7.93e-10	9.44e-16	7.91e-10	
ex921	LM-2	5	0	0	113	0.0000	1.61e-10	3.20e-10	6.76e-16	1.57e-10	
ex922	LM-2*	7	0	0	8	0.0000	2.81e-10	5.95e-07	4.39e-15	2.17e-10	
ex925	LM-2	11	0	0	12	0.0000	5.16e-08	6.35e-09	1.05e-15	5.16e-08	
ex928	LM-1	7	0	0	8	0.0000	1.78e-07	4.48e-08	1.00e-16	1.78e-07	
flp2	LM-1	5	0	0	6	0.0000	6.75e-10	2.69e-10	5.93e-16	6.73e-10	
gauvin	LM-1*	6	0	0	7	0.00000	5.18e-09	7.95e-10	4.97e-13	5.18e-09	
gnash1	**										
hakonsen	**										
jr1	LM-2	18	0	0	19	0.0000	7.68e-08	1.37e-10	1.08e-19	7.68e-08	
jr2	LM-2	18	0	0	19	0.0000	7.68e-08	1.37e-10	1.08e-19	7.68e-08	
kth1	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07	
kth2	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07	
kth3	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07	
liswet1-inv50	**										
nash1	LM-2	7	0	0	8	0.0000	4.93e-08	2.21e-08	1.36e-15	4.93e-08	
outrata31	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07	
outrata32	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07	
outrata33	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07	
outrata34	LM-1	8	0	0	18	0.0000	2.54e-07	7.06e-07	1.13e-07	2.27e-07	
portfl1	**										
qpec1	LM-2	19	0	0	20	0.0000	5.68e-07	4.49e-10	0.00000	1.32e-07	
qpecgen	**										
ralph2	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07	
ralphmod	**										
scale1	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07	
scale2	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07	
scale3	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07	
scale4	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07	
scale5	LM-2	17	0	0	18	0.0000	6.71e-07	1.15e-09	-	6.71e-07	
scholtes1	LM-1	8	0	0	9	0.0000	3.60e-07	4.32e-08	2.49e-08	3.59e-07	
scholtes2	LM-1	8	0	0	9	0.0000	3.60e-07	4.32e-08	2.49e-08	3.59e-07	
scholtes3	LM-2	17	0	0	18	0.0000	6.71e-07	1.16e-09	-	6.71e-07	
scholtes4	LM-1	16	0	0	11	0.0000	2.02e-04	8.90e-06	3.04e-14	1.96e-04	
scholtes5	LM-1	3	0	0	98	0.0000	9.99e-19	9.99e-19	0.00000	9.99e-19	
sl1	**										
stackelberg1	LM-1*	25	0	0	1767	0.0000	2.29e-09	2.61e-09	3.12e-15	2.29e-09	
traffic1	**										

Experimento 2: Cálculo de uma solução viável para MPCC com meta c_t

Nesta seção reportamos os experimentos com os algoritmos PGUN e PLM para o cálculo de uma solução viável para MPCC com meta c_t , isto é, uma solução para o HNCP (2.12). Realizamos os experimentos com os mesmos problemas da seção anterior e, para a meta c_t , utilizamos o menor valor de função conhecido.

Neste segundo experimento o método PGUN resolveu 39 dos 48 problemas testados, alcançando 81% de sucesso, enquanto que o método PLM resolveu 25 problemas, atingindo 52% de sucesso. Tal comportamento pode ser também observado na Figura 2.7, que expõe o número

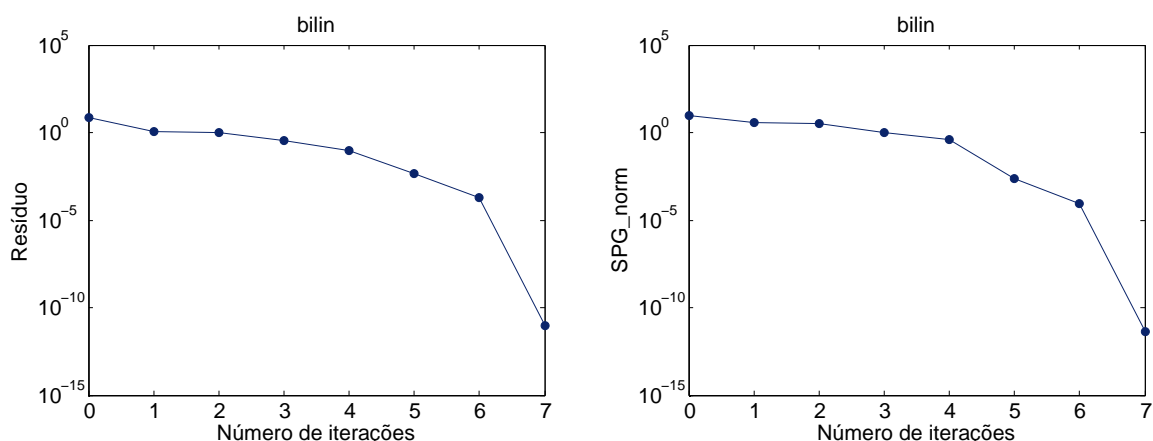


Figura 2.5: Gráficos de convergência do método PLM para o problema *bilin*.

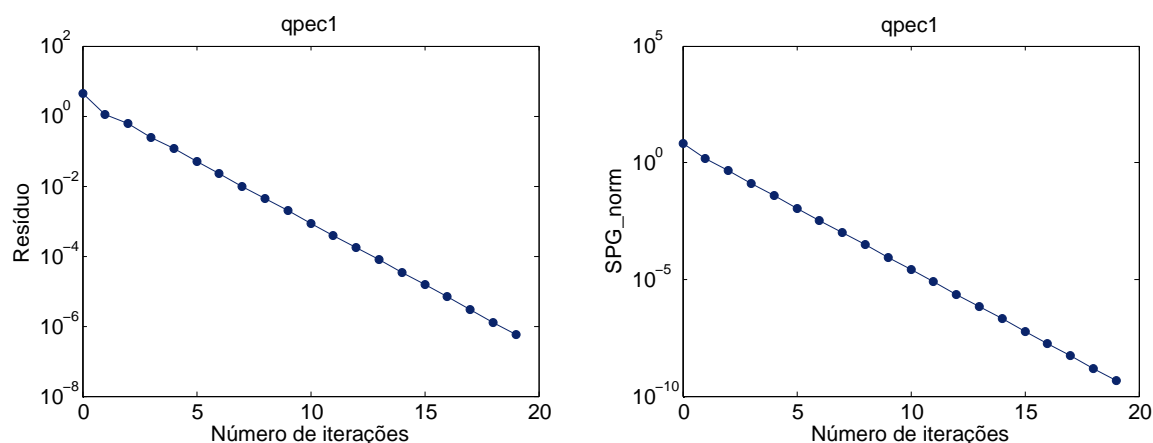


Figura 2.6: Gráficos de convergência do método PLM para o problema *qpec1*.

de problemas resolvidos por iteração para cada um dos dois métodos considerados.

Nas Tabelas 2.5 e 2.6 definimos os problemas testados e apresentamos os resultados obtidos. Nestas tabelas usamos a mesma notação do experimento anterior. Para complementar a análise da solução encontrada, na Tabela 2.6 acrescentamos a coluna:

SLACK: representa o valor da variável de folga associada à restrição no valor da função. Se SLACK é maior do que a tolerância 10^{-6} , então o algoritmo foi capaz de calcular uma solução viável melhor do que a dada pela coleção.

As Figuras 2.8 e 2.9 exibem os gráficos de convergência do método PGUN para os problemas *bard2* e *hakonsen*, respectivamente.

A Tabela 2.7 exhibe o desempenho do método PLM para o Experimento 2 e a Figura 2.10 os gráficos de convergência para o problema *hakonsen*.

Os resultados obtidos, para ambos os métodos, indicam um desempenho semelhante ao obtido para o Experimento 1. No entanto, houve um aumento no número de insucessos quando

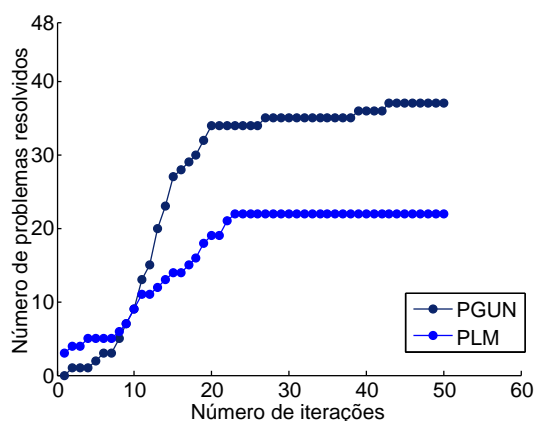


Figura 2.7: Número de problemas resolvidos por iteração para o Experimento 2.

Tabela 2.5: Número de pares complementares para o Experimento 2.

Problema	NCP	NNG	Problema	NCP	NNG	Problema	NCP	NNG
bard1	3	3	gauvin	2	3	qpecgen	100	3
bard2	4	18	gnash1	8	3	ralph2	1	1
bard3	2	6	hakonsen	4	5	ralphmod	100	9
bilevel1	6	6	jr1	1	1	scale1	1	1
bilevel3	4	4	jr2	1	1	scale2	1	1
bilin	6	4	kth1	1	1	scale3	1	1
dempe	1	1	kth2	1	1	scale4	1	1
design-cent1	3	4	kth3	1	1	scale5	1	1
desilva	2	5	liswet1-inv50	50	52	scholtes1	1	2
df1	1	5	nash1	2	5	scholtes2	1	2
ex911	5	2	outrata31	4	3	scholtes3	1	1
ex921	4	3	outrata32	4	3	scholtes4	1	3
ex922	4	5	outrata33	4	3	scholtes5	2	1
ex925	3	3	outrata34	4	3	sl1	3	8
ex928	2	4	portff1	12	63	stackelberg1	1	3
flp2	2	5	qpec1	10	11	traffic1	244	495

acrescentamos a restrição do valor da função. Além disso, os valores da variável de folga indicam que os algoritmos PGUN e PLM sempre calcularam a solução dada pela coleção. Esses resultados não são surpreendentes e confirmam as conclusões de [19] de que calcular um solução viável com meta é, em geral, mais difícil do que calcular uma solução viável simples.

Novamente, as Figuras 2.9 e 2.10 mostram que ambos os métodos apresentam um comportamento muito semelhante para os casos de sucesso.

Tabela 2.6: Desempenho do método PGUN para o Experimento 2.

Problema	TERM	IP	PG	CG	NE	TEMPO	RES	SPG	norm	FEAS	COMP	SLACK
bard1	IP-1	9	0	0	10	0.0000	7.93e-10	6.21e-09	7.72e-10	1.81e-10	-6.68e-18	
bard2	IP-1	18	0	0	19	0.0080	1.06e-12	9.27e-07	1.06e-12	8.36e-14	-1.72e-23	
bard3	IP-1	13	0	0	14	0.0000	1.54e-08	8.22e-08	1.48e-08	3.98e-09	-1.55e-17	
bilevel1	**											
bilvel3	IP-1	13	0	0	14	0.0000	2.74e-08	1.49e-07	2.70e-08	4.91e-09	3.12e-17	
bilin	**											
dempe	**											
design-cent1	**											
desilva	IP-2	13	0	0	14	0.0000	5.01e-07	8.52e-07	5.01e-07	1.41e-11	-1.03e-16	
df1	IP-2	13	0	0	14	0.0000	7.90e-07	2.02e-06	7.89e-07	4.69e-08	-1.53e-16	
ex911	IP-1	10	0	0	11	0.0000	6.26e-13	7.65e-13	2.11e-15	4.52e-13	1.29e-17	
ex921	IP-1	9	0	0	10	0.0000	7.74e-09	5.92e-08	7.37e-09	1.68e-09	-2.22e-18	
ex922	IP-2	17	0	0	18	0.0040	6.18e-07	3.93e-07	1.96e-08	6.12e-07	-3.15e-22	
ex925	IP-2	16	0	0	17	0.0000	4.48e-07	2.00e-06	4.48e-07	1.12e-15	1.04e-16	
ex928	IP-2	8	0	0	9	0.0000	2.13e-09	6.00e-09	7.22e-10	2.01e-09	1.15e-13	
fp2	IP-2	19	0	0	20	0.0000	3.59e-07	6.72e-10	3.59e-07	2.16e-17	-1.11e-16	
gauvin	IP-1	20	0	0	21	0.0000	3.24e-07	2.90e-06	3.24e-07	3.54e-16	-1.06e-16	
gnash1	IP-1*	43	0	0	57	0.0040	2.36e-07	8.88e-07	2.36e-07	3.09e-14	-5.87e-17	
hakonsen	IP-1*	10	0	0	11	0.0000	8.37e-15	9.94e-13	8.37e-15	5.07e-22	1.44e-05	
jr1	IP-2	11	0	0	12	0.0000	3.45e-07	4.88e-07	3.45e-07	3.00e-17	-1.52e-18	
jr2	IP-2	12	0	0	13	0.0000	3.65e-07	5.17e-07	3.65e-07	1.86e-17	2.60e-18	
kth1	IP-1*	5	0	0	6	0.0000	6.87e-07	4.88e-07	6.87e-07	2.84e-10	1.56e-10	
kth2	IP-1	2	0	0	3	0.0000	6.12e-07	3.53e-07	4.99e-07	2.49e-07	2.49e-07	
kth3	IP-2	12	0	0	13	0.0000	6.38e-07	6.38e-07	6.38e-07	7.83e-17	-1.93e-17	
liswet1-inv50	**											
nash1	IP-2*	14	0	0	15	0.0000	6.86e-07	1.13e-09	6.86e-07	5.00e-19	-1.43e-17	
outrata31	IP-1	11	0	0	12	0.0000	5.05e-08	1.23e-07	5.05e-08	1.29e-09	1.24e-14	
outrata32	II-IP*	2197	0	0	5607	0.2280	3.70e-06	1.07e-05	3.70e-06	9.13e-15	1.16e-14	
outrata33	IP-1	15	0	0	16	0.0000	2.54e-06	7.39e-06	2.54e-06	5.88e-16	-5.96e-15	
outrata34	IP-1	14	0	0	15	0.0000	2.61e-06	7.29e-06	2.61e-06	1.63e-16	8.24e-16	
portfl1	**											
qpec1	IP-2*	20	0	0	22	0.0080	5.61e-07	8.49e-09	5.61e-07	2.92e-16	1.40e-17	
qpecgen	**											
ralph2	IP-2	13	0	0	14	0.0000	4.01e-07	3.58e-07	3.58e-07	1.79e-07	-6.20e-25	
ralphmod	**											
scale1	IP-2	11	0	0	12	0.0000	7.88e-07	1.57e-06	7.88e-07	8.23e-17	-4.34e-19	
scale2	IP-2	39	0	0	85	0.0000	3.31e-07	6.62e-07	3.31e-07	2.75e-19	3.00e-19	
scale3	IP-2	15	0	0	16	0.0000	3.16e-07	6.33e-07	3.16e-07	4.58e-19	1.30e-19	
scale4	IP-1*	86	0	0	275	0.0000	4.86e-06	9.35e-06	4.60e-05	8.36e-07	1.04e-06	
scale5	IP-1*	14	0	0	15	0.0000	2.01e-08	4.03e-06	2.01e-08	2.35e-17	2.96e-19	
scholtes1	IP-2	15	0	0	16	0.0000	7.98e-07	1.42e-09	7.98e-07	4.42e-16	-1.75e-17	
scholtes2	IP-2*	6	0	0	7	0.0000	4.52e-08	1.80e-07	4.52e-08	1.49e-10	2.65e-10	
scholtes3	IP-2*	8	0	0	9	0.0000	4.13e-07	4.13e-07	4.13e-07	1.04e-16	-1.68e-17	
scholtes4	IP-2	11	0	0	12	0.0000	3.52e-07	3.05e-10	1.84e-18	2.39e-07	-2.84e-20	
scholtes5	PG-1	19	1	1	32	0.0000	2.22e-05	2.10e-07	2.22e-05	1.09e-19	-2.21e-20	
sl1	IP-2	15	0	0	16	0.0000	6.44e-07	5.81e-09	2.88e-07	5.76e-07	-1.10e-16	
stackelberg1	IP-1	27	0	0	28	0.0000	7.76e-08	3.73e-06	7.76e-08	2.99e-14	-7.33e-18	
traffic1	**											

2.5 Conclusões

Neste capítulo estudamos uma implementação esparsa para o método de Newton e aplicamos na solução de problemas com restrições de complementaridade. O algoritmo resultante utiliza iterações rápidas de Newton e direções lentas de gradientes projetados, é globalmente convergente e, sob certas hipóteses, apresenta convergência local quadrática. Além disso, o algoritmo estudado demonstra ser, em geral, mais eficiente do que o método PLM, algoritmo estabelecido anteriormente para o mesmo objetivo.

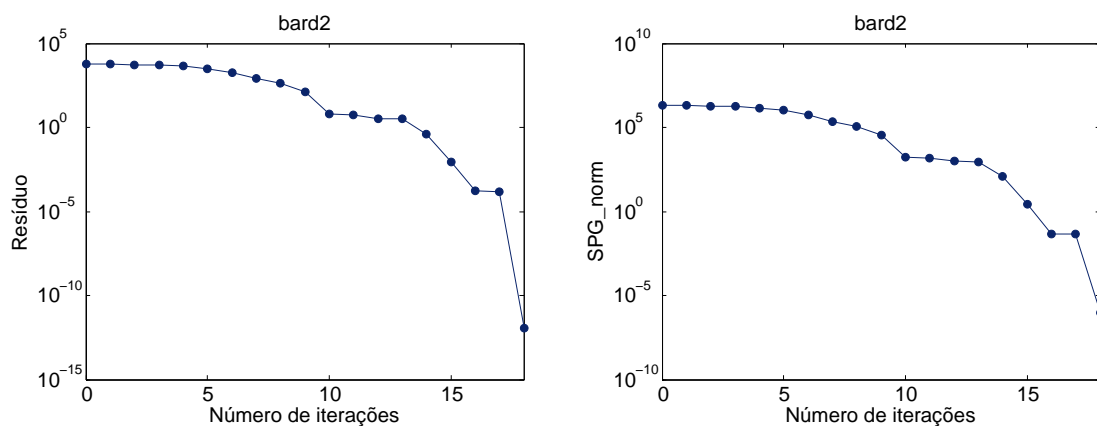


Figura 2.8: Gráficos de convergência do método PGUN para o problema *bard2*.

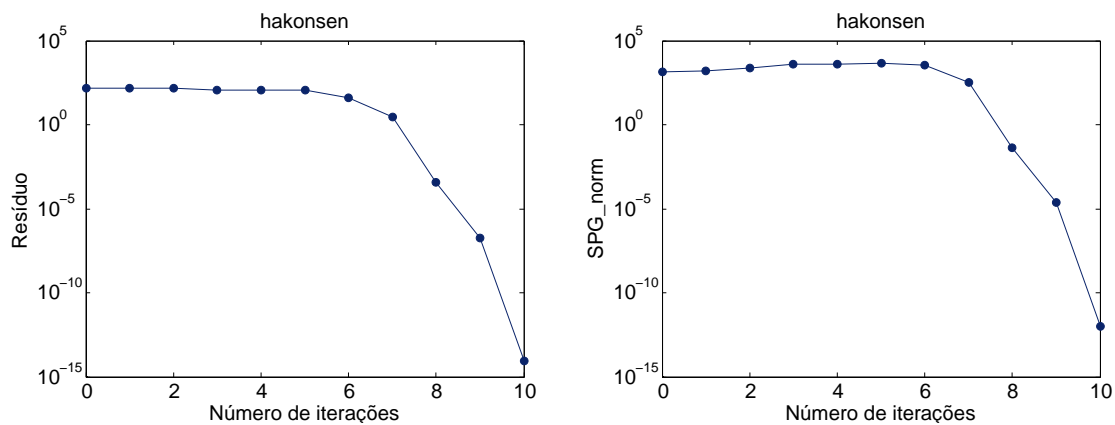


Figura 2.9: Gráficos de convergência do método PGUN para o problema *hakonsen*.

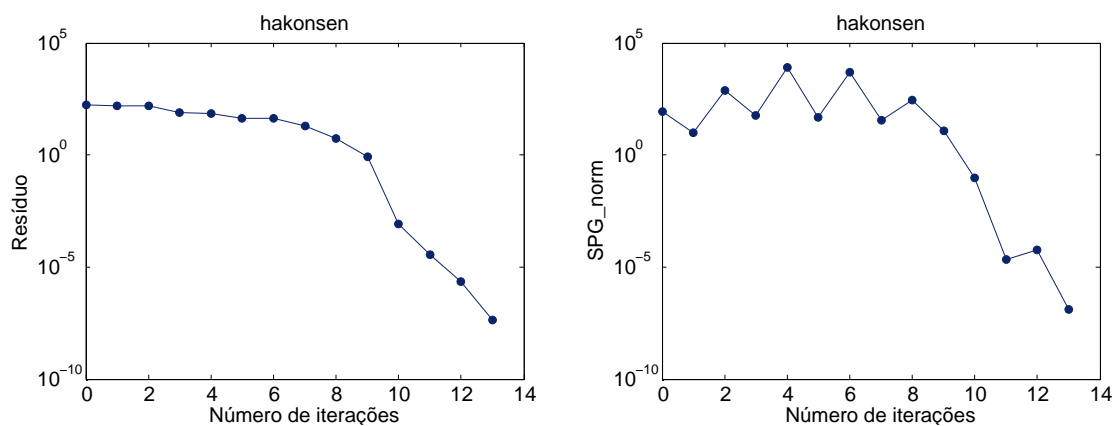


Figura 2.10: Gráficos de convergência do método PLM para o problema *hakonsen*.

Capítulo 3

Avaliação da estratégia quase-Newton de aceleração algorítmica

3.1 Introdução

Dada uma função $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$, um problema de ponto fixo consiste em encontrar $x \in \mathbb{R}^n$ tal que

$$x = G(x). \quad (3.1)$$

Associado a este problema, definimos uma iteração de ponto fixo por $x^{k+1} = G(x^k)$.

Problemas de ponto fixo surgem naturalmente em aplicações como cálculo de estruturas eletrônicas [34], modelos complexos de transporte [7], modelagem estatística [68], entre outros. A grande preocupação acerca da iteração de ponto fixo, como comenta [70], está no fato de que os iterados podem convergir com taxa linear e extremamente lenta. Por este motivo, vários autores vêm se dedicando a acelerar de alguma maneira a convergência desses iterados, principalmente para os exemplos acima citados [7, 17, 18, 33, 34, 39, 52, 62, 64, 67, 69, 70, 79].

O problema de encontrar um ponto fixo pode ser visto como um sistema não linear. Para isto, basta definirmos a função $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ por

$$F(x) = x - G(x). \quad (3.2)$$

No entanto, apesar de os problemas serem equivalentes do ponto de vista teórico, Fang & Saad [18] ressaltam que existem diferenças práticas entre resolver um sistema não linear e acelerar uma sequência de vetores. Por exemplo, a possível dificuldade em calcular F pode inviabilizar o uso de técnicas não lineares que envolvem o uso de informações do Jacobiano e, além disso, pode não ser apropriado avaliar $F(x + d)$ para um d arbitrário.

A fim de escolhermos a classe de métodos mais adequada para acelerar a convergência de iterados de problemas de ponto fixo, é conveniente considerarmos as características típicas desse tipo de problema. De acordo com [7, 18] problemas práticos de ponto fixo possuem grande dimensão, sendo, portanto, recomendável o uso de aceleradores com memória limitada e que não requeiram inversão de matrizes. Outra questão importante destacada em [18] refere-se ao uso de técnicas de busca linear, uma vez que o custo de avaliar a função pode ser extremamente alto, tais técnicas devem ser evitadas. Por esses motivos, algoritmos baseados na equação secante se tornaram as opções mais utilizadas para lidar com tais problemas [18].

No decorrer do texto comentamos a relação dos métodos secantes com o método de Steffensen, apresentamos os métodos quase-Newton utilizados e, por fim, validamos a técnica em exemplos treino.

O método de Steffensen

Para o caso $n = 1$ um processo bem conhecido para acelerar a convergência de iterações de ponto fixo é o chamado método de Steffensen. Este método consegue alcançar convergência quadrática sem utilizar derivadas. Para o problema $\phi(x) = x$, $\phi : \mathbb{R} \rightarrow \mathbb{R}$, o algoritmo está definido a seguir.

Algoritmo 3.1. Método de Steffensen

Inicialização: Escolha $x^0 \in \mathbb{R}$.

Passo 1. Calcule x^1 e x^2 usando iterações de ponto fixo.

Passo 2. Use o processo de Aitken para calcular \hat{x}^0 a partir de x^0 , x^1 e x^2 , isto é,

$$\hat{x}^0 = \frac{x_0 x_2 - x_1^2}{x_0 - 2x_1 + x_2};$$

Passo 3. Repita os passos 1 e 2 com $x^0 = \hat{x}^0$.

Dado o sistema $f(x) = x - \phi(x) = 0$, a aceleração quase-Newton definida pelo Primeiro Método de Broyden com $B_0 = I$ gera os seguintes iterados:

$$\begin{aligned} x_1 &= x_0 - B_0^{-1} f(x_0) = x_0 - f(x_0) = \phi(x_0) \\ x_2 &= x_1 - B_1^{-1} f(x_1) \\ &= x_1 - f(x_1) - \frac{x_1 - f(x_1) - (x_0 - f(x_0))}{f(x_1) - f(x_0)} f(x_1) \\ &= \phi(x_1) - \frac{(\phi(x_1) - \phi(x_0))(x_1 - \phi(x_1))}{x_1 - \phi(x_1) - (x_0 - \phi(x_0))} \\ &= x_2 + \frac{(x_2 - x_1)(x_1 - x_2)}{x_0 - 2x_1 + x_2} \\ &= \frac{x_0 x_2 - x_1^2}{x_0 - 2x_1 + x_2}. \end{aligned}$$

Dessa forma, o processo de aceleração quase-Newton é uma generalização do método de Steffensen no seguinte sentido: quando consideramos o caso $n = 1$ e $B_k = I$ sempre que $k \equiv 0 \pmod{2}$, a sequência de iterados gerada pelo processo de aceleração quase-Newton é uma subsequência dos iterados gerados pelo método de Steffensen. O mesmo resultado é válido quando consideramos as atualizações quase-Newton definidas pelos demais métodos secantes.

A seguir introduzimos os métodos quase-Newton utilizados.

3.2 Implementação Computacional

Vamos considerar o sistema de equações não lineares

$$F(x) = 0, \quad (3.3)$$

em que $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é continuamente diferenciável.

Como visto anteriormente, métodos práticos para resolver (3.3) consistem em um processo iterativo. O método de Newton é o mais conhecido e pode ser aplicado sempre que o Jacobiano for não singular. Neste caso, o processo iterativo é definido por (2.3). No entanto, encontrar a direção de Newton pode ser uma tarefa de alto custo computacional, pois envolve o cálculo de derivadas parciais e a solução de um sistema linear que pode ser de grande porte.

Os métodos quase-Newton surgiram dessa necessidade de resolver um sistema não linear com menor esforço computacional do que o utilizado pelo método de Newton, mas sem perder as boas propriedades do mesmo. Dessa forma, define-se a iteração quase-Newton da seguinte maneira:

$$x^{k+1} = x^k - B_k^{-1}F(x^k), \quad (3.4)$$

em que $B_k \in \mathbb{R}^{n \times n}$ é uma aproximação da matriz Jacobiana de F no ponto x^k . Neste contexto o próprio método de Newton pode ser visto como um método quase-Newton.

Os Métodos Secantes constituem uma classe de modelos de atualizações quase-Newton amplamente utilizadas. Para defini-los assumimos que, na iteração k , a função F pode ser aproximada pela fórmula linear $L_k(x) = F(x^k) + B_k(x - x^k)$, e impomos que a aproximação linear $L_{k+1}(x)$ interpole a função F em x^k e x^{k+1} , isto é,

$$L_{k+1}(x^k) = F(x^k), \quad (3.5)$$

e

$$L_{k+1}(x^{k+1}) = F(x^{k+1}). \quad (3.6)$$

Note que a igualdade (3.6) é verificada e (3.5) equivale a

$$B_{k+1}s_k = y_k, \quad (3.7)$$

em que $y_k = F(x^{k+1}) - F(x^k)$ e $s_k = x^{k+1} - x^k$. O sistema linear (3.7) é conhecido como equação secante e tem como incógnita a matriz $B_{k+1} \in \mathbb{R}^{n \times n}$. Quando $n > 1$ e $s_k \neq 0$, o sistema tem infinitas soluções e cada uma dessas soluções define um método diferente. As atualizações de B_{k+1} consideradas neste trabalho são:

$$(i) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^\top}{s_k^\top s_k};$$

$$(ii) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k) y_k^\top B_k}{y_k^\top B_k s_k};$$

$$(iii) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k) e_{j_k}^\top}{e_{j_k}^\top s_k};$$

$$(iv) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k) e_{j_k}^\top B_k}{e_{j_k}^\top B_k s_k};$$

$$(v) \quad B_{k+1} = B_k + \frac{(y_k - B_k s_k) w_k^\top}{w_k^\top s_k}, \text{ em que } w_k = \left(P_k + \frac{\|s_k\|}{2} I \right) s_k \text{ e}$$

$$P_{k+1} = (1 + \|s_k\|) \left(\|s_k\| I + P_k - \frac{w_k w_k^\top}{w_k^\top s_k} \right), \quad P_0 = \rho^2 I.$$

Tais atualizações definem o Primeiro e Segundo Métodos de Broyden [9], o Método de atualização de uma coluna por iteração [44], o Método de atualização de uma coluna por iteração inverso [47] e o Método de Thomas [65], respectivamente.

Sabemos que se partimos de um ponto x^0 suficientemente próximo da solução x^* e que se todas as matrizes B_k também estão próximas de $J(x^*)$, então a sequência gerada por (3.4) está bem definida e converge para x^* com taxa superlinear. Detalhes sobre as propriedades de convergência dos métodos quase-Newton foram omitidos, uma vez que consistem em resultados teóricos bem conhecidos e podem ser encontrados, por exemplo, em [46].

Na sequência comentamos os detalhes da implementação numérica dos métodos quase-Newton adotados.

Primeiro Método de Broyden - MB1

No Primeiro Método de Broyden (MB1) [9] a matriz B_{k+1} é escolhida de forma que a mudança prevista para F pela equação secante com B_{k+1} em uma direção q_k ortogonal a s_k seja a mesma que seria prevista com B_k , isto é,

$$B_{k+1} q_k = B_k q_k, \quad q_k^\top s_k = 0. \quad (3.8)$$

As equações (3.7) e (3.8) definem B_{k+1} de maneira única. Com efeito, por (3.8) temos que

$$B_{k+1} = B_k + uv^\top$$

como consequência, por (3.7) segue que

$$v = s_k \text{ e } u = \frac{y_k - B_k s_k}{v^\top s_k}.$$

Logo,

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) s_k^\top}{s_k^\top s_k}.$$

Vamos trabalhar com a inversa da matriz B_{k+1} , uma vez que esta pode ser facilmente obtida da inversa de B_k . Assim, da fórmula de Sherman-Morrison, se B_k^{-1} existe e $s_k^\top B_k^{-1} y_k \neq 0$, então B_{k+1}^{-1} existe e é definida por

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) s_k^\top B_k^{-1}}{s_k^\top B_k^{-1} y_k},$$

ou ainda,

$$B_{k+1}^{-1} = (I + u_k s_k^\top) B_k^{-1} = \prod_{i=0}^k (I + u_i s_i^\top) B_0^{-1} \quad (3.9)$$

em que $u_k = \frac{(s_k - B_k^{-1} y_k)}{s_k^\top B_k^{-1} y_k}$.

Atualizar a matriz B_k de acordo com (3.9) tem seu grau de dificuldade potencializado conforme o número de iterações cresce. Assim, para esse método utilizaremos a proposta de atualização proposta em [14] e exposta no lema a seguir.

Lema 3.1. Sejam $d_k = -B_k^{-1}F(x^k)$, $z_k = B_k^{-1}y_k$, $\tilde{z}_i = B_i^{-1}y_i$, $\gamma_i = \frac{s_i^\top \tilde{z}_i}{s_i^\top z_i}$ e $\tau_k = \frac{s_k^\top d_k}{s_k^\top z_k}$. Considere a iteração $x^{k+1} = x^k + t_k d_k$ e assumamos que $s_i^\top z_i \neq 0$, $i = 0, 1, \dots, k-1$. Então,

- (i) $d_{k+1} = d_k - s_k + \tau_k(s_k - z_k)$;
- (ii) $\tilde{z}_{i+1} = \tilde{z}_i + \frac{\gamma_i}{\tau_i}(d_{i+1} - (1 - t_i)d_i)$.

Demonstração. Com efeito,

$$\begin{aligned}
 d_{k+1} &= -B_{k+1}^{-1}F(x^{k+1}) = -\left(B_k^{-1} + \frac{(s_k - B_k^{-1}y_k)s_k^\top B_k^{-1}}{s_k^\top B_k^{-1}y_k}\right)F(x^{k+1}) \\
 &= -B_k^{-1}F(x^{k+1}) - \frac{(s_k - z_k)}{s_k^\top z_k} s_k^\top B_k^{-1}F(x^{k+1}) \\
 &= -B_k^{-1}(F(x^k) + y_k) - \frac{(s_k - z_k)}{s_k^\top z_k} s_k^\top B_k^{-1}[F(x^k) + y_k] \\
 &= d_k - B_k^{-1}y_k + \frac{(s_k - z_k)}{s_k^\top z_k} s_k^\top d_k - \frac{(s_k - z_k)}{s_k^\top z_k} s_k^\top B_k^{-1}y_k \\
 &= d_k - z_k + \tau_k(s_k - z_k) - \frac{(s_k - z_k)}{s_k^\top z_k} s_k^\top z_k \\
 &= d_k - z_k + \tau_k(s_k - z_k) - s_k + z_k \\
 &= d_k - s_k + \tau_k(s_k - z_k)
 \end{aligned}$$

o que prova o item (i).

Para demonstrar o item (ii), note que

$$\begin{aligned}
 \tilde{z}_{i+1} &= B_{i+1}^{-1}y_{i+1} = \left(B_i^{-1} + \frac{(s_i - B_i^{-1}y_i)s_i^\top B_i^{-1}}{s_i^\top B_i^{-1}y_i}\right)y_{i+1} \\
 &= B_i^{-1}y_{i+1} + \frac{(s_i - z_i)}{s_i^\top z_i} s_i^\top B_i^{-1}y_{i+1} \\
 &= \tilde{z}_i + \frac{(s_i - z_i)}{s_i^\top z_i} s_i^\top \tilde{z}_i \\
 &= \tilde{z}_i + \gamma_i(s_i - z_i) \\
 &= \tilde{z}_i + \gamma_i\left(\frac{1}{\tau_i}(d_{i+1} - d_i + s_i)\right) \\
 &= \tilde{z}_i + \frac{\gamma_i}{\tau_i}(d_{i+1} - (1 - t_i)d_i).
 \end{aligned}$$

□

O Lema 3.1 permite implementar o Primeiro Método de Broyden armazenando, até a iteração k , apenas os vetores d_i e os valores τ_i e t_i , $i = 0, \dots, k-1$.

Lembrando da condição imposta pela fórmula de Sherman-Morrison, sempre que $d_k^\top z_k \leq \text{tol} \|z_k\|$, $\text{tol} > 0$, vamos definir $B_{k+1} = B_k$. Para isso basta fazer $z_{k+1} = \tilde{z}_k$ e $\tau_k = 1$. Com efeito,

$$d_{k+1} = d_k - z_k = -B_k^{-1}F(x^k) - B_k^{-1}y_k = -B_k^{-1}F(x^{k+1})$$

e

$$z_{k+1} = \tilde{z}_k = B_k^{-1}y_{k+1}.$$

Algoritmo 3.2. Primeiro Método de Broyden - MB1

Inicialização: Sejam $\text{tol}, \text{BIG} > 0$. Escolha $x^0 \in \mathbb{R}^n$ e faça $B_0 = I$.

Passo 1. *Iteração de Ponto Fixo:* $x^1 = x^0 - F(x^0)$.

Faça $k = 1$.

Passo 2. *Critério de parada:* Se $F(x^k) = 0$, pare.

Passo 3. *Cálculo da direção:* Calcule

$$d_k = (1 - t_{k-1})d_{k-1} + \tau_{k-1}(t_{k-1}d_{k-1} - z_{k-1}).$$

Passo 4. *Escolha do passo:* Se $\|d_k\| < \text{BIG}$, faça $t_k = 1$. Senão, faça $t_k = \frac{\text{BIG}}{\|d_k\|}$.

Passo 5. *Cálculo do novo iterado:* Faça $x^{k+1} = x^k + t_k d_k$.

Passo 6. *Cálculo de z_k :* Defina $\tilde{z}_0 = B_0^{-1}y_k$. Para $i = 0, \dots, k-1$ faça

$$\tilde{z}_{i+1} = \tilde{z}_i + \frac{\gamma_i}{\tau_i}(d_{i+1} - (1 - t_i)d_i).$$

Defina $z_k = \tilde{z}_k$.

Passo 7. *Controle:* Se $|d_k^\top z_k| \leq \text{tol} \|d_k\| \|z_k\|$ ou $\|d_k\| \leq \text{tol}^2$, defina $\tau_k = 1$ e $z_k = \tilde{z}_{k-1}$.

Faça $k = k + 1$ e volte ao Passo 2.

Fim.

Segundo Método de Broyden - MB2

O Segundo Método de Broyden (MB2) [9] é definido de maneira semelhante ao primeiro, sendo a utilização da fórmula secante inversa a principal alteração. Dessa forma, chega-se à seguinte atualização:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1}y_k)y_k^\top}{y_k^\top y_k}.$$

Definindo

$$w_k = \frac{s_k - B_k^{-1}y_k}{y_k^\top y_k}, \quad (3.10)$$

podemos escrever

$$B_{k+1}^{-1} = B_k^{-1} + w_k y_k^\top = B_0^{-1} + \sum_{i=0}^k w_i y_i^\top. \quad (3.11)$$

Como consequência, o vetor w_k e a direção $d_k = -B_k^{-1}F(x^k)$ podem ser atualizados como segue

$$w_k = \frac{1}{y_k^\top y_k} \left(s_k - y_k - \sum_{i=0}^{k-1} (y_i^\top y_k) w_i \right)$$

e

$$d_k = -F(x^k) - \sum_{i=0}^{k-1} (y_i^\top F(x^k)) w_i,$$

respectivamente.

Dessa forma podemos implementar o algoritmo do Segundo Método de Broyden armazenando, até a iteração k , os vetores y_i e w_i , $i = 0, \dots, k-1$.

Algoritmo 3.3. Segundo Método de Broyden - MB2

Inicialização: Escolha $x^0 \in \mathbb{R}^n$ e faça $H_0 = I$.

Passo 1. *Iteração de Ponto Fixo:* $x^1 = x^0 - F(x^0)$.

Faça $k = 1$.

Passo 2. *Critério de parada:* Se $F(x^k) = 0$, pare.

Passo 3. *Cálculo da direção:* Calcule

$$d_k = -F(x^k) - \sum_{i=0}^{k-1} (y_i^\top F(x^k)) w_i.$$

Passo 4. *Escolha do passo:* Se $\|d_k\| < \text{BIG}$, faça $t_k = 1$. Senão, faça $t_k = \frac{\text{BIG}}{\|d_k\|}$.

Passo 5. *Cálculo do novo iterado:* Faça $x^{k+1} = x^k + t_k d_k$.

Passo 6. *Cálculo de w_k :* Se $\|y_k\| \leq \text{tol} \|F(x^k)\|$, faça $w_k = 0$. Senão, calcule

$$w_k = \frac{1}{y_k^\top y_k} \left(t_k d_k - y_k - \sum_{i=0}^{k-1} (y_i^\top y_k) w_i \right).$$

Faça $k = k + 1$ e volte ao Passo 2.

Fim.

Para evitar instabilidade na definição da matriz B_{k+1}^{-1} sempre que $\|y_k\| \leq \text{tol} \|F(x^k)\|$ faremos $B_{k+1}^{-1} = B_k^{-1}$. Para tanto, no Passo 6 do algoritmo basta definirmos $w_k = 0$.

O Primeiro e o Segundo Método de Broyden são também conhecidos como Método de Broyden Bom e Método de Broyden Ruim, respectivamente. As razões pelas quais os métodos receberam tais nomes não são claras. Além disso, na prática, nem sempre o Método de Broyden Bom tem melhor desempenho do que o Método de Broyden Ruim [45].

Método de atualização de uma coluna por iteração - MAC

O Método de atualização de uma coluna por iteração (MAC) foi introduzido em [44] com o intuito de reduzir o esforço computacional do Primeiro Método de Broyden. Este método consiste em, a cada iteração, modificar apenas a j -ésima coluna de B_k , com j definido por

$$|[s_k]_j| = \|s_k\|_\infty.$$

O método fica então definido por

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) e_{j_k}^\top}{e_{j_k}^\top s_k}, \quad (3.12)$$

em que e_i é o i -ésimo vetor canônico de \mathbb{R}^n .

De (3.12) e da fórmula de Sherman-Morrison segue que, se B_k^{-1} existe e $e_{j_k}^\top B_k^{-1} y_k \neq 0$, então B_{k+1}^{-1} existe e é dada por

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) e_{j_k}^\top B_k^{-1}}{e_{j_k}^\top B_k^{-1} y_k}. \quad (3.13)$$

Repetindo o processo feito para o Primeiro Método de Broyden concluímos que:

$$d_{k+1} = -B_{k+1}^{-1} F(x^{k+1}) = (1 - t_k) d_k + \frac{d_{j_k}}{z_{j_k}} (t_k d_k - z_k) \quad (3.14)$$

e

$$\tilde{z}_{i+1} = B_{i+1}^{-1} y_k = \tilde{z}_i + \frac{e_{j_i}^\top \tilde{z}}{e_{j_i}^\top d_i} (d_{i+1} - (1 - t_i) d_i). \quad (3.15)$$

Assim, podemos estabelecer o seguinte algoritmo.

Algoritmo 3.4. Método de atualização de uma coluna por iteração - MAC

Inicialização: Escolha $x^0 \in \mathbb{R}^n$, $\text{tol} > 0$, $\text{BIG} > 0$ e faça $B_0 = I$.

Passo 1. *Iteração de Ponto Fixo:* $x^1 = x^0 - F(x^0)$.

Faça $k = 1$.

Passo 2. *Critério de parada:* Se $F(x^k) = 0$, pare.

Passo 3. *Cálculo da direção:* Calcule

$$d_k = (1 - t_{k-1}) d_{k-1} + \tau_{k-1} (t_{k-1} d_{k-1} - z_{k-1}).$$

Passo 4. *Escolha do passo:* Se $\|d_k\| \leq \text{BIG}$, faça $t_k = 1$. Senão, faça $t_k = \frac{\text{BIG}}{\|d_k\|}$.

Passo 5. *Cálculo do novo iterado:* Faça $x^{k+1} = x^k + t_k d_k$.

Passo 6. *Cálculo de z_k :* Defina $\tilde{z}_0 = B_0^{-1} y_k$. Para $i = 0, \dots, k-1$ faça

$$\tilde{z}_{i+1} = \tilde{z}_i + \frac{\gamma_i}{\tau_i} (d_{i+1} - (1 - t_i) d_i).$$

Faça $z_k = \tilde{z}_k$.

Passo 7. *Controle:* Se $|e_{j_k}^\top z_k| \leq \text{tol} \|z_k\|$ ou $|e_{j_k}^\top d_k| \leq \text{tol}^2$, defina $\tau_k = 1$ e $z_k = \tilde{z}_{k-1}$.

Faça $k = k + 1$ e volte ao Passo 2.

Fim.

Analogamente ao que foi feito para o MB1, quando $\|z_{j_k}\| \leq \text{tol} \|z_k\|$ definimos $z_k = \tilde{z}_{k-1}$ e $\tau_k = 1$. Dessa forma, $B_{k+1}^{-1} = B_k^{-1}$.

Método de atualização de uma coluna por iteração inverso - MACI

O Método de atualização de uma coluna por iteração inverso (MACI) foi descrito por Martínez e Zambaldi em [47]. Relacionado com o MAC, este método propõe a atualização da matriz B_{k+1}^{-1} modificando, também, apenas uma coluna da matriz anterior B_k^{-1} , ou seja,

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1}y_k)e_{j_k}^\top}{e_{j_k}^\top y_k} \quad (3.16)$$

com j_k escolhido de maneira que $|e_{j_k}^\top y_k| = \|y_k\|_\infty$.

Definindo $u_k = \frac{s_k - B_k^{-1}y_k}{e_{j_k}^\top y_k}$ temos que

$$B_{k+1}^{-1} = B_k^{-1} + u_k e_{j_k}^\top = B_0^{-1} + \sum_{i=0}^k u_i e_{j_i}^\top.$$

Assim, atualizamos a direção d_k por

$$d_{k+1} = -B_k^{-1}F(x^k) = -B_0^{-1}F(x^k) - \sum_{i=0}^{k-1} [F(x^k)]_{j_i} u_i$$

e, até a iteração k guardamos os vetores u_i e os índices j_i , $i = 0, \dots, k-1$.

O algoritmo MACI está colocado a seguir.

Algoritmo 3.5. Método de atualização de uma coluna por iteração inverso - MACI

Inicialização: Escolha $x^0 \in \mathbb{R}^n$, $\text{tol} > 0$, $BIG > 0$.

Passo 1. *Iteração de Ponto Fixo:* $x^1 = x^0 - F(x^0)$.

Faça $k = 1$.

Passo 2. *Critério de parada:* Se $F(x^k) = 0$, pare.

Passo 3. *Cálculo da direção:* Faça

$$d_k = -F(x^k) - \sum_{i=0}^{k-1} [F(x^k)]_{j_i} u_i.$$

Passo 4. *Escolha do passo:* Se $\|d_k\| \leq BIG$, faça $t_k = 1$. Senão, faça $t_k = \frac{BIG}{\|d_k\|}$.

Passo 5. *Cálculo do novo iterado:* Faça $x^{k+1} = x^k + t_k d_k$.

Passo 6. *Cálculo de u_k :* Se $\|y_k\| \leq \text{tol}\|F(x^k)\|$, faça $u_k = 0$. Senão, calcule

$$u_k = \frac{1}{[y_k]_{j_k}} \left(s_k - y_k - \sum_{i=0}^{k-1} [y_k]_{j_i} u_i \right).$$

Faça $k = k + 1$ e volte ao Passo 2.

Fim.

Para evitar instabilidade na definição da matriz B_{k+1}^{-1} , sempre que $\|y_k\| \leq \text{tol}\|F(x^k)\|$, definimos $u_k = 0$, isto é, $B_{k+1}^{-1} = B_k^{-1}$.

Método de Thomas - MTH

O Método de Thomas (MTH) foi definido em [65] e consiste na seguinte atualização

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k) w_k^\top}{w_k^\top s_k}, \quad (3.17)$$

em que

$$w_k = \left(P_k + \frac{\|s_k\|}{2} I \right) s_k$$

e

$$P_{k+1} = (1 + \|s_k\|) \left(\|s_k\| I + P_k - \frac{w_k w_k^\top}{w_k^\top s_k} \right), \quad P_0 = \rho^2 I.$$

De (3.17) e da Fórmula de Sherman-Morrison segue que, se B_k^{-1} existe, $w_k^\top s_k \neq 0$ e $w_k^\top B_k^{-1} y_k \neq 0$ então B_{k+1}^{-1} existe e é definida por

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k) w_k^\top B_k^{-1}}{w_k^\top B_k^{-1} y_k}.$$

Analogamente ao que foi feito para o Primeiro Método de Broyden, concluímos que:

$$d_{k+1} = (1 - t_k) d_k + \frac{w_k^\top d_k}{w_k^\top z_k} (t_k d_k - z_k)$$

e

$$\tilde{z}_{i+1} = \tilde{z}_i + \frac{w_i^\top \tilde{z}_i}{w_i^\top d_i} (d_{i+1} - (1 - t_i) d_i).$$

Dessa forma, até a iteração k basta guardarmos os vetores d_i , w_i , $i = 1, \dots, k$ e os valores $w_i^\top d_i$, t_i e $\|d_i\|$, $i = 1, \dots, k$.

Escrevendo,

$$\begin{aligned} w_k &= \left(P_k + \frac{\|s_k\|}{2} I \right) s_k = \frac{\|s_k\|}{2} s_k + P_k s_k \\ &= \frac{\|s_k\|}{2} s_k + \sum_{i=0}^{k-1} \prod_{j=i}^{k-1} (1 + \|s_j\|) \left(\|s_i\| s_k - \frac{w_i^\top s_k}{w_i^\top s_i} w_i \right) + \prod_{j=0}^{k-1} (1 + \|s_j\|) P_0 s_k \\ &= \frac{t^2 \|d_k\|}{2} d_k + \sum_{i=0}^{k-1} \prod_{j=i}^{k-1} (1 + \|t_j d_j\|) \left(t_k \|t_i d_i\| d_k - \frac{t_k w_i^\top d_k}{t_i w_i^\top d_i} w_i \right) + \prod_{j=0}^{k-1} (1 + \|t_j d_j\|) t_k P_0 d_k \end{aligned} \quad (3.18)$$

podemos definir o algoritmo do Método de Thomas como segue.

Algoritmo 3.6. Método de Thomas - MTH

Inicialização: Escolha $x^0 \in \mathbb{R}^n$, $\rho > 0$, $\text{tol} > 0$, $BIG > 0$, faça $P_0 = \rho^2 I$ e $B_0 = I$.

Passo 1. *Iteração de Ponto Fixo:* $x^1 = x^0 - F(x^0)$.

Faça $k = 1$.

Passo 2. *Critério de parada:* Se $F(x^k) = 0$, pare.

Passo 3. *Cálculo da direção:* Faça

$$d_k = (1 - t_k)d_k + \frac{w_k^\top d_k}{w_k^\top z_k}(t_k d_k - z_k).$$

Passo 4. *Escolha do passo:* Se $\|d_k\| \leq \text{BIG}$, faça $t_k = 1$. Senão, faça $t_k = \frac{\text{BIG}}{\|d_k\|}$.

Passo 5. *Cálculo do novo iterado:* Faça $x^{k+1} = x^k + t_k d_k$.

Passo 6. *Cálculo de z_k :* Defina $\tilde{z}_0 = B_0^{-1}y_k$. Para $i = 0, \dots, k - 1$ faça

$$\tilde{z}_{i+1} = \tilde{z}_i + \frac{w_i^\top \tilde{z}_i}{w_i^\top d_i}(d_{i+1} - (1 - t_i)d_i).$$

Faça $z_k = \tilde{z}_k$.

Passo 7. *Cálculo de w_k :* Calcule w_k como em (3.18).

Passo 8. *Controle:* Se $|w_k^\top z_k| \leq \text{tol} \|w_k\| \|z_k\|$ ou $|w_k^\top d_k| \leq \|w_k^\top\| \|d_k\|$, defina $\tau_k = 1$ e faça $z_k = \tilde{z}_{k-1}$.

Faça $k = k + 1$ e volte ao Passo 2.

Fim.

Os problemas testados, juntamente com os resultados obtidos, estão expostos na próxima seção.

3.3 Resultados Numéricos e Problemas

Os algoritmos descritos na seção anterior foram implementados usando gfortran-4.6 em um computador Intel CORE I3-2310M @ 2.10 GHz com 100 Gb de HD e 4Gb de Ram. Utilizamos o mesmo critério de parada para todos os métodos, a saber, o algoritmo termina na iteração k para a qual o resíduo $\|F(x^k)\|$ é menor do que uma certa tolerância. Utilizamos a tolerância igual a 10^{-6} significando que o seguinte critério de parada foi satisfeito:

Parar com x^k se $\|F(x^k)\| < 10^{-6}$.

Os métodos testados desfrutam de convergência local e, dependendo do iterado inicial, x^0 , podem não convergir para uma solução. Para evitar esse comportamento, decidimos parar o método prematuramente utilizando um segundo critério de parada:

Parar na iteração k se $\|F(x^k)\| \geq 10^6 \|F(x^0)\|$.

Além dos dois critérios colocados acima, limitamos o número de iterações em $\max\{100, n^3\}$ e o tempo computacional em 60 segundos.

Para as constantes definidas nos algoritmos apresentados na seção anterior utilizamos: $\text{tol} = 10^{-7}$, $\text{BIG} = \min\{10^6, 10^6 \|x\|\}$ e, para o Método de Thomas consideramos $\rho = 1$.

Nossos experimentos numéricos estão divididos em dois grupos. O primeiro engloba os testes que motivaram o estudo da aceleração quase-Newton e o segundo, consiste em uma aplicação da estratégia estudada.

3.3.1 Experimento 1: problemas treino

O Experimento 1 engloba os testes que motivaram o estudo da aceleração quase-Newton. Neste experimento consideramos a abordagem apresentada nas seções anteriores para acelerar a convergência dos seguintes métodos: Gauss-Seidel, Máxima Descida, Frank-Wolfe e Newton-Lagrange. Os resultados e detalhes da implementação estão colocados a seguir.

Exemplo 3.1. Aceleração do Método Gauss-Seidel (MGS):

O Método Gauss-Seidel (MGS) consiste em um processo iterativo para a resolução de sistemas lineares que pode ser descrito por

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=i}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right),$$

em que $A \in \mathbb{R}^{n \times n}$ é uma matriz diagonalmente dominante, $b \in \mathbb{R}^n$ e $x^k \in \mathbb{R}^n$.

Para a aceleração quase-Newton deste método consideramos a função

$$G(x) = -(L + D)^{-1}(Ux + b),$$

em que $L \in \mathbb{R}^{n \times n}$ é a parte triangular inferior estrita de A , $D \in \mathbb{R}^{n \times n}$ é a matriz formada pelos elementos da diagonal de A e $U \in \mathbb{R}^{n \times n}$ é a parte triangular superior estrita de A , isto é, escrevemos $A = L + D + U$.

Nos experimentos utilizamos a matriz de discretização abaixo

$$A = \begin{bmatrix} 2(1+h^2) & -1 & 0 & \dots & 0 \\ -1 & 2(1+h^2) & -1 & \dots & 0 \\ 0 & \ddots & \ddots & \ddots & \\ \vdots & & & & -1 \\ 0 & \dots & 0 & -1 & 2(1+h^2) \end{bmatrix} \quad \text{e} \quad b = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \quad (3.19)$$

Na Tabela 3.1 descrevemos o comportamento dos métodos Gauss-Seidel e aceleração quase-Newton aplicados ao sistema $Ax = b$ definido por (3.19). Nesta tabela, utilizamos a seguinte notação:

NI: número de iterações;

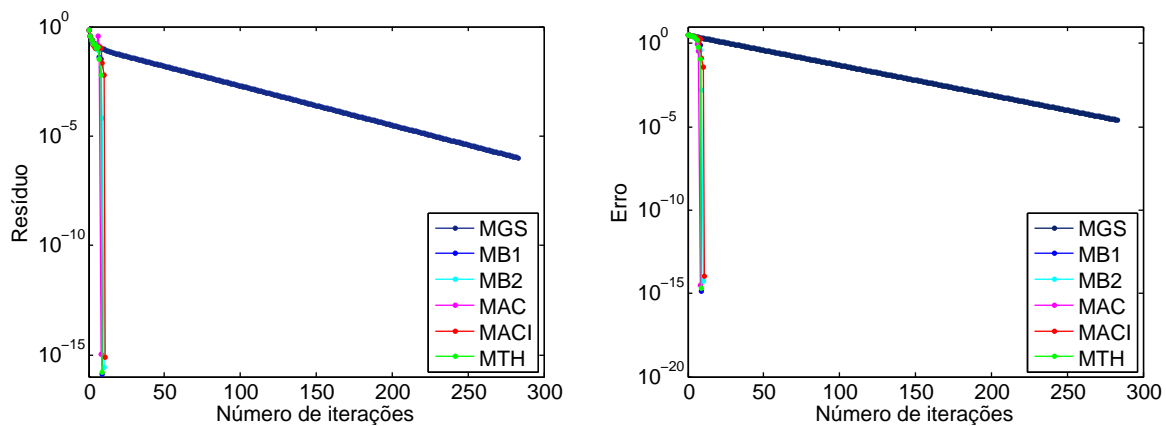
TEMPO: tempo computacional (em segundos) medido pela função `etime`;

RES: o resíduo encontrado, isto é, $\|F(\bar{x})\|$ em que \bar{x} é a solução encontrada pelo algoritmo;

ERRO: o erro com relação à solução exata x^* , isto é, $\|\bar{x} - x^*\|$.

Tabela 3.1: Comparativo entre os métodos testados para o Exemplo 3.1.

n	$n = 10$				$n = 50$			
	NI	TEMPO	RES	ERRO	NI	TEMPO	RES	ERRO
MGS	283	0.0000	9.94e-07	2.45e-05	4956	0.0200	9.99e-07	5.27e-04
MB1	9	0.0000	1.35e-16	1.36e-15	48	0.0040	8.00e-07	1.92e-04
MB2	10	0.0000	2.71e-16	5.10e-15	38	0.0000	1.85e-07	7.42e-05
MAC	8	0.0000	1.06e-15	3.09e-15	48	0.0000	2.74e-09	9.26e-07
MACI	11	0.0000	7.52e-16	1.00e-14	42	0.0000	5.35e-07	4.97e-05
MTH	9	0.0000	1.57e-16	2.12e-15	49	0.0000	7.73e-08	2.19e-05
n	$n = 100$				$n = 500$			
	NI	TEMPO	RES	ERRO	NI	TEMPO	RES	ERRO
MGS	17329	0.1160	9.99e-07	2.06e-03	304258	10.100	9.99e-07	5.08e-02
MB1	145	0.0160	5.13e-07	4.34e-04	1335	5.3843	8.38e-07	1.28e-03
MB2	67	0.0040	8.63e-07	5.36e-04	297	0.5040	7.20e-07	1.25e-02
MAC	110	0.0000	9.10e-07	6.52e-04	1302	1.4680	9.60e-07	1.21e-02
MACI	75	0.0000	3.61e-07	2.90e-04	330	0.1280	4.84e-07	4.37e-04
MTH	136	0.0360	7.41e-07	7.59e-04	1329	14.924	8.58e-07	2.30e-03

Figura 3.1: Gráfico de convergência dos métodos testados para o Exemplo 3.1 com $n = 10$.

Nas Figuras 3.1 e 3.2 expomos o gráfico de convergência dos métodos testados para o problema definido por (3.19) com $n = 10$.

A partir dos resultados expostos, percebemos que o algoritmo MGS apresentou um comportamento típico de convergência linear, sendo necessário um número elevado de iterações para satisfazer o critério de parada. Por outro lado, a estratégia de aceleração quase-Newton apresentou uma redução significativa tanto no número de iterações (redução média maior que 98%) quanto no tempo computacional. Tal comportamento pode ser observado para todos os métodos quase-Newton considerados, exceto o MTH para $n = 500$. Além disso, houve uma melhora na precisão da solução encontrada.

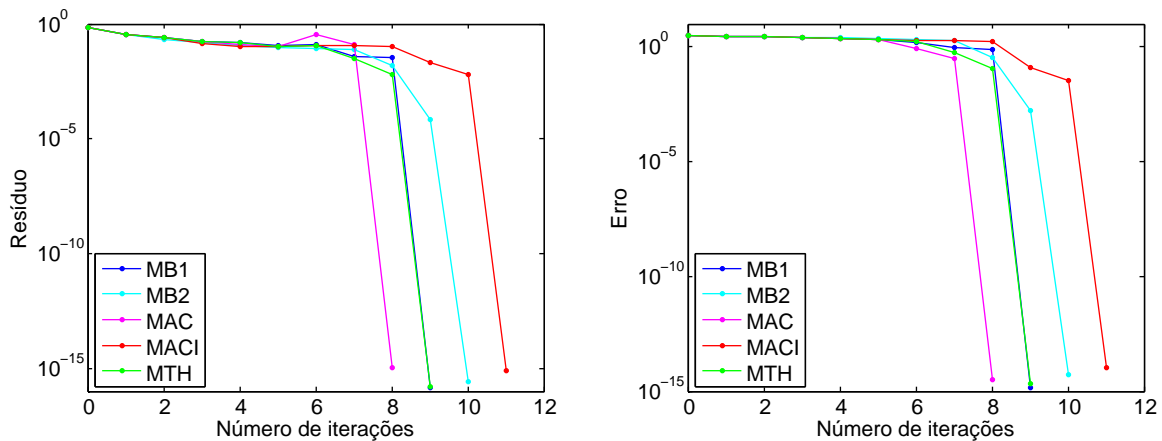


Figura 3.2: Gráfico de convergência dos métodos quase-Newton para o Exemplo 3.1 com $n = 10$.

Exemplo 3.2. Aceleração do Método de Máxima Descida (MMD):

O Método de Máxima Descida (MMD) é um método clássico para minimizar uma função. O processo iterativo consiste em mover-se na direção oposta ao gradiente da função objetivo utilizando uma busca linear exata. Para a aceleração quase-Newton deste método consideramos a função

$$G(x) = x - \alpha(x)g(x),$$

em que $g(x) = \nabla f(x)$, $f(x) = \frac{1}{2}x^\top Ax + b^\top x$, $\alpha(x) = \frac{g(x)^\top g(x)}{g(x)^\top Ag(x)}$, $A \in \mathbb{R}^{n \times n}$ é uma matriz simétrica definida positiva, $b \in \mathbb{R}^n$ e $x \in \mathbb{R}^n$.

A Tabela 3.2 contém os resultados obtidos com os diferentes métodos para o Exemplo 3.2 definido a partir de (3.19). Na sequência, a Figura 3.3 expõe o comportamento gráfico dos métodos para o mesmo experimento.

Novamente, os resultados obtidos indicam que o MMD apresentou convergência linear, enquanto que a estratégia quase-Newton alcançou convergência superlinear. A redução média no número de iterações obtida foi maior que 94% e, mais importante, sem afetar a qualidade da solução.

Exemplo 3.3. Aceleração do algoritmo Frank-Wolfe (AFW):

O algoritmo Frank-Wolfe (AFW) [21] aplica-se ao problema

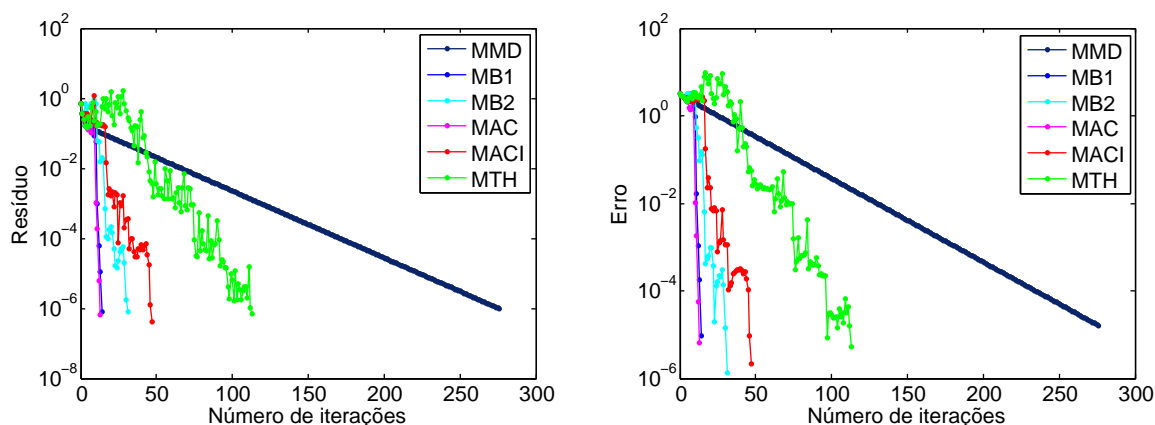
$$\begin{aligned} &\text{Minimizar } f(x) \\ &\text{s.a. } x \in D \end{aligned} \quad (3.20)$$

em que f é uma função convexa, diferenciável e D é um conjunto convexo e limitado.

O método consiste em, a cada iteração, minimizar uma aproximação linear da função objetivo, mantendo-se sempre no conjunto viável.

Tabela 3.2: Comparativo entre os métodos testados para o Exemplo 3.2.

n	$n = 10$				$n = 50$			
	NI	TEMPO	RES	ERRO	NI	TEMPO	RES	ERRO
MMD	276	0.0000	9.74e-07	1.59e-05	5126	0.0240	9.99e-07	3.71e-04
MB1	14	0.0000	7.96e-07	9.31e-06	148	0.0080	2.50e-07	2.91e-05
MB2	31	0.0000	7.95e-07	1.35e-06	56	0.0000	7.22e-07	5.22e-06
MAC	13	0.0000	6.41e-07	6.33e-06	125	0.0000	6.40e-07	2.22e-04
MACI	47	0.0000	4.00e-07	2.10e-06	69	0.0000	4.37e-07	5.25e-05
MTH	113	0.0040	1.05e-06	5.14e-06	255	0.1120	1.04e-05	2.58e-04
n	$n = 100$				$n = 500$			
	NI	TEMPO	RES	ERRO	NI	TEMPO	RES	ERRO
MMD	18034	0.1880	9.99e-07	1.46e-03	321879	16.069	9.99e-07	3.59e-02
MB1	231	0.0400	4.69e-07	2.66e-04	4455	60.003	6.44e-04	6.4577
MB2	102	0.0120	5.11e-07	7.34e-05	346	0.6840	7.43e-07	6.31e-03
MAC	2181	0.8320	8.65e-07	1.15e-03	8326	60.011	1.83e-05	0.1354
MACI	109	0.0000	4.48e-07	9.45e-05	352	0.1520	8.96e-07	6.19e-03
MTH	209	0.0840	9.76e-07	9.54e-05	1			

Figura 3.3: Gráfico de convergência dos métodos testados para o Exemplo 3.2 com $n = 10$.**Algoritmo 3.7. Algoritmo Frank-Wolfe - AFW**

Inicialização: Escolha $x^0 \in D$ e faça $k = 0$.

Passo 1. *Encontrar a direção do subproblema:* Encontre s_k solução de

$$\begin{aligned} & \text{Minimizar} \quad s^\top g(x^k) \\ & \text{s.a} \quad s \in D \end{aligned} \quad (3.21)$$

Passo 2. *Determinar o tamanho do passo:* Encontre γ_k solução de

$$\begin{aligned} & \text{Minimizar } f(x^k + \gamma(s_k - x^k)) \\ & \text{s.a. } 0 \leq \gamma \leq 1 \end{aligned} \quad (3.22)$$

Passo 3. Atualização: Faça $x^{k+1} = x^k + \gamma_k(s_k - x^k)$, $k = k + 1$ e volte ao Passo 1.

Note que o novo iterado é uma combinação convexa entre dois pontos do conjunto viável, a saber, x^k e s_k , assim, a sequência $\{x^k\}$ gerada pelo Algoritmo 3.7 está contida no conjunto viável. A Figura 3.4 exemplifica o processo iterativo que define o algoritmo AFW.

A aceleração quase-Newton do algoritmo é definida a partir da função

$$G(x) = x + \gamma(s_k - x)$$

em que $f(x) = \frac{1}{2}x^\top Ax + b^\top x$, $A \in \mathbb{R}^{n \times n}$ é uma matriz simétrica definida positiva, $b \in \mathbb{R}^n$ e $x \in \mathbb{R}^n$. Vamos considerar o mesmo problema definido por (3.19), restrito ao conjunto convexo $D = \{x \in \mathbb{R}^n : c \leq x \leq d\}$, isto é, D é uma caixa.

Note que, neste caso, a solução dos problemas que definem os passos 1 e 2 do Algoritmo 3.7 é trivial, a saber:

- Passo 1: A solução do problema (3.21) é dada por

$$[s_k]_i = \begin{cases} [c]_i, & \text{caso } [g(x^k)]_i \geq 0 \\ [d]_i, & \text{caso } [g(x^k)]_i < 0 \end{cases}$$

em que $g(x) = \nabla f(x)$.

- Passo 2: A solução de (3.22) é dada por

$$\gamma_k = \frac{g(x^k)^\top (x^k - s_k)}{(x^k - s_k)^\top A (x^k - s_k)}.$$

Na Tabela 3.3 comparamos o desempenho do algoritmo Frank-Wolfe com os métodos de aceleração quase-Newton para o Exemplo 3.3. Nesta tabela, acrescentamos a notação:

†: o algoritmo gerou pontos não viáveis;

‡: o algoritmo satisfaz o segundo critério de parada, isto é, $\|F(x^k)\| \geq 10^6 \|F(x^0)\|$;

‡: o algoritmo atingiu o tempo limite ou número máximo de iterações.

Note que o algoritmo Frank-Wolfe, em todos os casos, alcançou RES e ERRO nulos, isto ocorreu pois na última iteração o minimizador da função linearizada encontrado pelo algoritmo foi nulo e o valor de γ correspondente foi $\gamma = 1$, resultando em $x^{k+1} = x^k + s^k - x^k = s^k = 0$, que é a solução do problema. Ilustramos este fato na Figura 3.5, para o problema definido por (3.19), com $n = 2$, caixa definida por $0 \leq x_i \leq 2$ e $x^0 = (1, 1)^\top$. Além disso, os métodos de aceleração quase-Newton geraram pontos não viáveis em algumas iterações. Nestes casos o ponto foi projetado no conjunto viável e continuamos o processo. De fato, não há garantia de que a aceleração quase-Newton encontre pontos viáveis, indicando que a abordagem utilizada não é adequada para processos iterativos como o Algoritmo Frank-Wolfe.

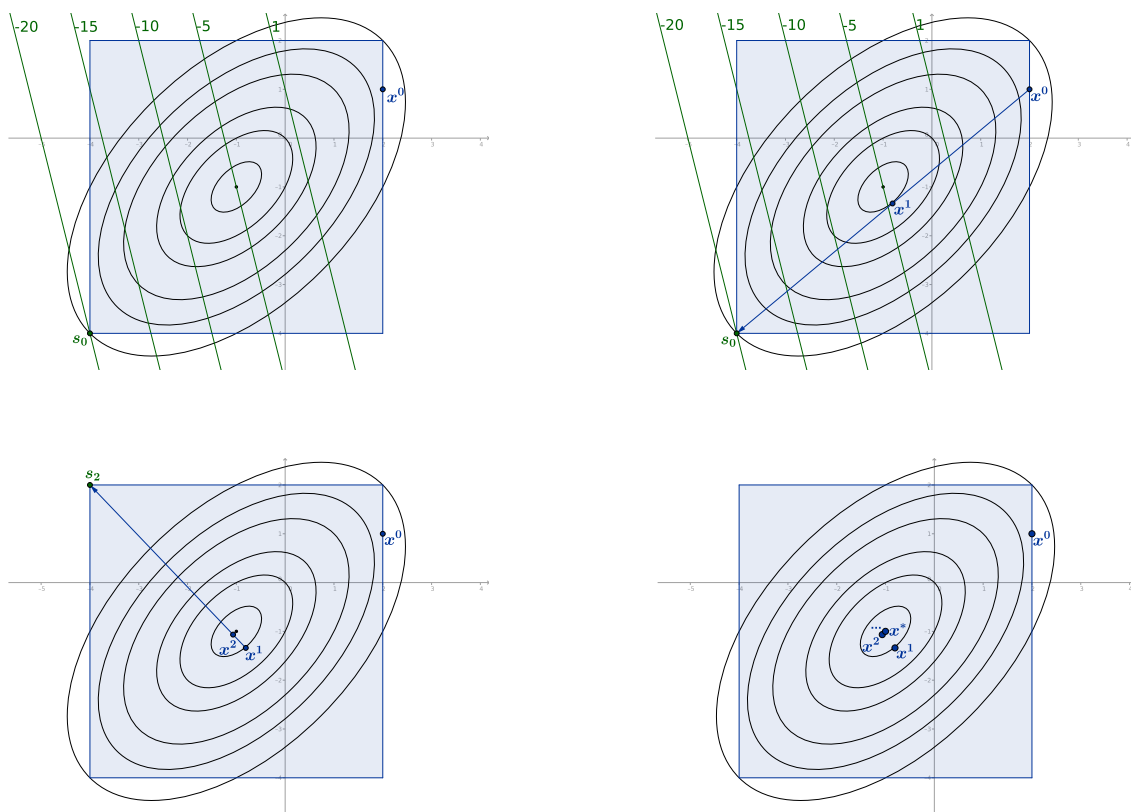


Figura 3.4: Pontos gerados pelo Algoritmo Frank-Wolfe com $n = 2$ e $x_0 = (2, 1)^\top$.

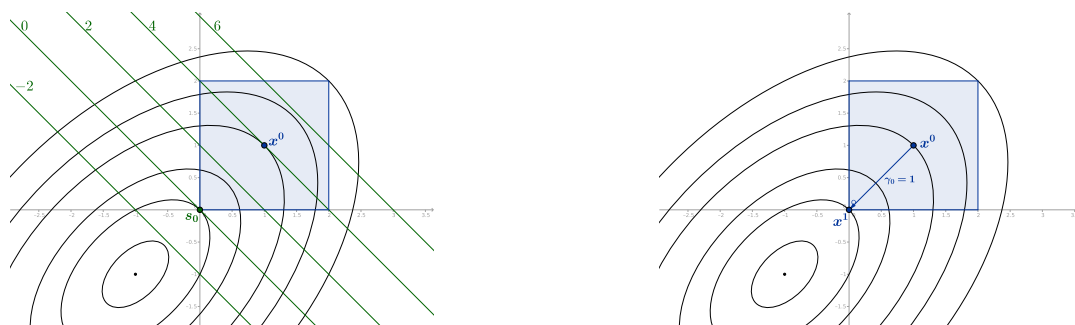


Figura 3.5: Uma iteração do Algoritmo Frank-Wolfe com $n = 2$ e $x_0 = (1, 1)^\top$.

Tabela 3.3: Comparativo entre os métodos testados para o Exemplo 3.3 com $h = 0$ e caixa definida por $0 \leq x_i \leq 10$.

Método	x_i^0	$n = 10$				$n = 50$			
		NI	TEMPO	RES	ERRO	NI	TEMPO	RES	ERRO
AFW	0.1*i	19	0.0000	0.0	0.0	899	0.0040	0.0	0.0
MB1	0.1*i	32 [†]	0.0000	0.0	0.0	8719 [†]	25.305	0.0	0.0
MB2	0.1*i	1000 ^{†b}	0.1360	4.3591	19.531	9890 [†]	60.003 ^b	18.807	53.177
MAC	0.1*i	422 [†]	0.0080	0.0	0.0	24120 [†]	60.003 ^b	10.693	34.897
MACI	0.1*i	1000 ^{†b}	0.0520	11.108	19.383	4851 ^{†‡}	3.2362	595543.7	41.2310
MTH	0.1*i	38 [†]	0.0000	0.0	0.0	34 ^{†‡}	0.0080	999990.4	34.641
		$n = 100$				$n = 500$			
AFW	0.01*i	9220	0.1240	0.0	0.0	575586	37.434	0.0	0.0
MB1	0.01*i	9645 [†]	60.007 ^b	0.2018	48.7509	4349 [†]	60.027 ^b	92.1824	153.986
MB2	0.01*i	7002 [†]	60.015 ^b	19.088	64.710	3167 [†]	60.027 ^b	8115.9	154.16
MAC	0.01*i	7940 ^{†‡}	11.848	39456.2	75.425	5345 ^{†‡}	25.641	564343.0	158.745
MACI	0.01*i	316 ^{†‡}	0.0360	24957.6	86.717	479 ^{†‡}	0.3280	193389.1	152.97
MTH	0.01*i	25 ^{†‡}	0.0200	999880.5	93.808	26 ^{†‡}	0.0680	999901.5	182.7566

Exemplo 3.4. Aceleração do método de Newton-Lagrange (MNL) na presença de multiplicadores críticos:

Considere o problema de otimização com restrições de igualdade definido por

$$\begin{aligned} &\text{Minimizar } f(x) \\ &\text{s.a. } h(x) = 0, \end{aligned}$$

em que $f : \mathbb{R}^n \rightarrow \mathbb{R}$ e $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ são suficientemente suaves. Este problema está associado ao sistema não linear

$$\nabla f(x) + \nabla h(x)\lambda = 0, \quad h(x) = 0 \quad (3.23)$$

cujas variáveis são $x \in \mathbb{R}^n$ e $\lambda \in \mathbb{R}^m$. Denotamos por $\nabla h(x) = (\nabla h_1(x), \dots, \nabla h_m(x)) \in \mathbb{R}^{n \times m}$. O método de Newton-Lagrange (MNL) consiste em aplicar o método de Newton ao sistema (3.23).

Se (x^*, λ^*) é uma solução deste sistema e as colunas de $\nabla h(x^*)$ são linearmente dependentes, então existem infinitas soluções da forma (x^*, λ) . Se existe $v \neq 0$ tal que $\nabla h(x^*)^\top v = 0$ e

$$\left[\nabla^2 f(x^*) + \sum_{i=1}^m \lambda_i \nabla^2 h_i(x^*) \right] v = 0,$$

dizemos que o vetor λ de multiplicadores de Lagrange é crítico.

Multiplicadores de Lagrange críticos estão associados com convergência lenta do método de Newton quando aplicado ao sistema (3.23). Em certos casos, quando o método converge para uma solução de (3.23), independentemente da aproximação inicial, a sequência $\{\lambda_k\}$ converge

para um multiplicador crítico [31]. Quando isto ocorre, a convergência não é superlinear. Esse fenômeno foi estudado em vários trabalhos por Izmailov, Solodov, e coautores, veja, por exemplo, [28, 29, 30, 31, 32] e as referências contidas nos mesmos.

Neste exemplo nos propomos a acelerar a convergência do método de Newton-Lagrange no caso de convergência para multiplicadores de Lagrange críticos. Para tal, aplicaremos a abordagem quase-Newton a partir da função

$$G(x) = x - J(x)^{-1}F(x),$$

em que $J(x) \in \mathbb{R}^{(n+m) \times (n+m)}$ é a matriz Jacobiana associada ao sistema (3.23).

Para os experimentos numéricos usamos os problemas de [31] colocados a seguir, nos quais observa-se convergência linear do método de Newton-Lagrange para uma solução primal-dual com multiplicadores críticos.

Exemplo 3.4.1 Considere

$$\begin{array}{ll} \text{Minimizar} & 0 \\ \text{s.a} & x^2 = 0. \end{array}$$

Neste caso o único multiplicador crítico é $\lambda = 0$. Utilizamos o vetor unitário como aproximação inicial, isto é, $(x^0, \lambda_0)^\top = (1, 1)^\top$.

Exemplo 3.4.2 Considere

$$\begin{array}{ll} \text{Minimizar} & x_1^2 \\ \text{s.a} & x_1^2 - x_2^2 = 0. \end{array}$$

Este exemplo apresenta dois multiplicadores críticos, a saber, -1 e 0 . Como aproximação inicial, utilizamos o vetor $(x^0, \lambda^0)^\top = (0.5, 0.5, 0.5)^\top$.

Exemplo 3.4.3 Considere

$$\begin{array}{ll} \text{Minimizar} & x_1^2 - x_2^2 + 2x_3^2 \\ \text{s.a} & -\frac{1}{2}x_1^2 + x_2^2 - \frac{1}{2}x_3^2 = 0 \\ & x_1x_3 = 0. \end{array}$$

O conjunto de multiplicadores críticos consiste em $\lambda \in \mathbb{R}^2$ tais que $\lambda_1 = 1$ ou $(\lambda_1 - 3)^2 - \lambda_2^2 = 1$. Utilizamos o vetor unitário como aproximação inicial.

Exemplo 3.4.4 Considere

$$\begin{array}{ll} \text{Minimizar} & x_1^2 + x_2^2 + x_3^2 \\ \text{s.a} & \text{sen}x_1 + \text{sen}x_2 + \text{sen}x_3 = 0 \\ & x_1 + x_2 + x_3 + \text{sen}(x_1x_2) = 0. \end{array}$$

Este exemplo possui solução única dada por $x^* = (0, 0, 0)^\top$ e $\lambda^* \in \{\lambda \in \mathbb{R}^2 : \lambda_1 + \lambda_2 = 0\}$, sendo os multiplicadores críticos $(-2, 2)^\top$ e $(6, -6)^\top$. Como aproximação inicial, usamos o vetor $(x^0, \lambda^0)^\top = (0.1, 0.2, 0.3, -1, 1)^\top$.

Exemplo 3.4.5 Considere

$$\begin{array}{ll} \text{Minimizar} & \alpha_1 x_1^2 \\ \text{s.a} & \alpha_2 x_1^2 = 0, \end{array}$$

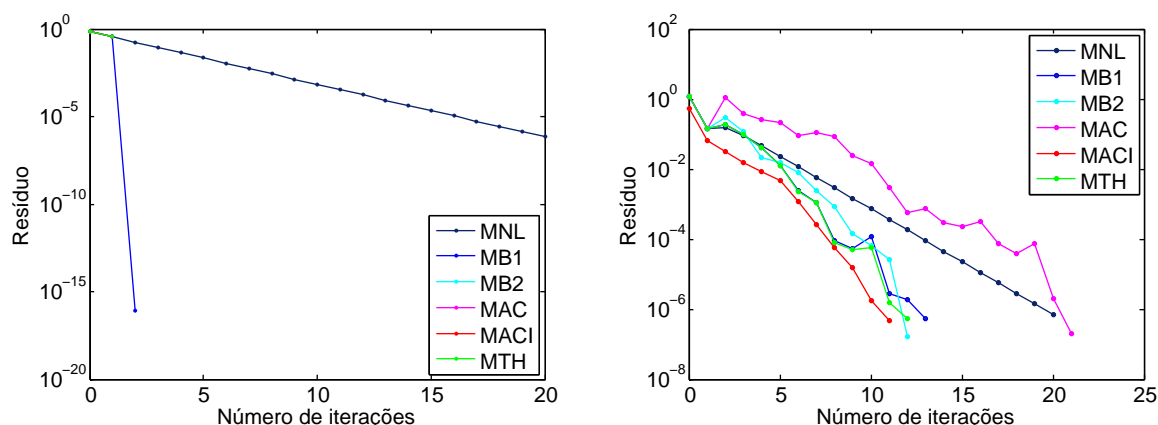


Figura 3.6: Taxa de convergência dos métodos testados para os Exemplos 3.4.1 (esquerda) e 3.4.6 (direita).

em que $\alpha_1 \neq 0$, $\alpha_2 \neq 0$.

O único multiplicador crítico é $\lambda = \frac{-\alpha_1}{\alpha_2}$. Vamos considerar os seguintes casos:

(a) $\alpha_1 = \alpha_2 = 1$;

(b) $\alpha_1 = 10$, $\alpha_2 = 1$.

Utilizamos o vetor unitário como aproximação inicial.

Exemplo 3.4.6 Considere

$$\begin{aligned} &\text{Minimizar} && \frac{1}{2}x^\top Ax \\ &\text{s.a} && \frac{1}{2}x^\top Bx = 0, \end{aligned}$$

em que A e B são matrizes simétricas e B é não singular.

Utilizamos $n = 100$, $A = I$ e B a matriz definida em (3.19) com $h = 0$. Como aproximação inicial, adotamos o vetor $(x^0, \lambda^0)^\top = (0.1, 0.1, \dots, 0.1)^\top$.

Exemplo 3.4.7 Considere

$$\begin{aligned} &\text{Minimizar} && 2x_1^2 + 2x_2^2 + 5x_1x_3 \\ &\text{s.a} && x_1x_2 + x_1x_3 + x_2x_3 = 0. \end{aligned}$$

Como aproximação inicial, utilizamos o vetor $(x^0, \lambda^0)^\top = (0.1, 0.1, 0.1, 5.1)^\top$.

Os resultados obtidos estão dispostos na Tabela 3.4. Nesta tabela, acrescentamos a notação:

*: quando o método convergiu para um multiplicador de Lagrange crítico;

** : quando o método convergiu para um multiplicador de Lagrange não crítico.

A Figura 3.6 reforça a convergência superlinear dos métodos de aceleração quase-Newton e a convergência linear do método de Newton-Lagrange na presença de multiplicadores críticos.

Os resultados apresentados na Tabela 3.4 mostram que, embora os métodos quase-Newton, em geral, também tenham convergido para uma solução primal-dual com multiplicadores de

Tabela 3.4: Comparativo entre os métodos testados para o Exemplo 3.4.

Método	Exemplo 3.4.1			Exemplo 3.4.2		
	NI	TEMPO	RES	NI	TEMPO	RES
MNL	20*	0.0000	6.74e-07	19*	0.0000	9.53e-07
MB1	2*	0.0000	7.85e-17	2**	0.0000	0.0
MB2	2*	0.0000	0.0	3*	0.0000	2.78e-17
MAC	2*	0.0000	0.0	2**	0.0000	0.0
MACI	2*	0.0000	0.0	3*	0.0000	2.77e-17
MTH	2*	0.0000	0.0	2**	0.0000	0.0
Método	Exemplo 3.4.3			Exemplo 3.4.4		
	NI	TEMPO	RES	NI	TEMPO	RES
MNL	19*	0.0000	9.50e-07	24*	0.0000	7.33e-07
MB1	11*	0.0000	3.91e-07	10*	0.0000	5.43e-07
MB2	7*	0.0000	2.36e-07	10*	0.0000	4.55e-07
MAC	4*	0.0000	5.00e-07	10*	0.0000	8.07e-08
MACI	4*	0.0000	5.00e-07	10*	0.0000	9.15e-08
MTH	11*	0.0000	3.91e-07	10*	0.0000	2.46e-07
Método	Exemplo 3.4.5(a)			Exemplo 3.4.5(b)		
	NI	TEMPO	RES	NI	TEMPO	RES
MNL	21*	0.0000	5.33e-07	23*	0.0000	6.58e-07
MB1	2*	0.0000	2.28e-16	2*	0.0000	5.55e-17
MB2	2*	0.0000	0.0	2*	0.0000	0.0
MAC	2*	0.0000	0.0	2*	0.0000	0.0
MACI	2*	0.0000	0.0	2*	0.0000	0.0
MTH	2*	0.0000	0.0	2*	0.0000	0.0
Método	Exemplo 3.4.6			Exemplo 3.4.7		
	NI	TEMPO	RES	NI	TEMPO	RES
MNL	20*	0.0120	7.07e-07	23*	0.0000	8.20e-07
MB1	13*	0.0080	5.40e-07	9*	0.0000	2.89e-07
MB2	12*	0.0000	1.65e-07	9*	0.0000	2.20e-07
MAC	21*	0.0120	1.97e-07	9*	0.0000	3.01e-08
MACI	11*	0.0120	4.77e-07	9*	0.0000	3.49e-08
MTH	12*	0.0080	5.44e-07	13*	0.0000	2.23e-07

Lagrange críticos, houve uma redução média de 56% no número de iterações. Além de reduzir de maneira expressiva o número de iterações, a aceleração quase-Newton também melhorou a precisão da solução encontrada em todos os exemplos considerados.

3.3.2 Experimento 2: aplicação prática

O Experimento 2 consiste em uma aplicação prática da estratégia quase-Newton proposta. Para isso, vamos considerar o método CGNewton.

Dado o problema irrestrito

Minimizar $f(x)$

a direção de Newton para uma iteração do problema é obtida resolvendo-se o sistema linear

$$\nabla^2 f(x)d = -\nabla f(x) \quad (3.24)$$

ou, de maneira equivalente, resolvendo-se o problema quadrático

$$\text{Minimizar } \frac{1}{2}d^\top \nabla^2 f(x)d + \nabla f(x)^\top d. \quad (3.25)$$

O método que estamos chamando CGNewton consiste em aplicar o Método de Gradientes Conjugados para resolver (3.25) de maneira inexata, realizando no máximo `maxcg` iterações do mesmo. Esse processo pode deixar o algoritmo com convergência linear, uma vez que podemos parar muito “longe” da direção de Newton desejada. Para recuperar uma boa taxa de convergência, vamos aplicar a aceleração quase-Newton ao método CGNewton.

Para esse fim, aplicaremos a abordagem quase-Newton a partir da função

$$G(x) = x + td,$$

em que d é a solução aproximada de (3.25) e $t > 0$. Ou seja, G define uma iteração do método CGNewton.

Na sequência definimos os problemas considerados.

Exemplo 3.5. Considere a função

$$f(x) = \sum_{i=1}^n \frac{x_i^2}{i+1}.$$

Exemplo 3.6. Considere a função

$$f(x) = 2x_2 - x_1 + \sum_{i=2}^{n-1} (x_{i-1} + 2x_i - x_{i+1}) + \sum_{i=1}^n x_i^4 + 2x_n - x_{n-1}.$$

Exemplo 3.7. Considere a função

$$f(x) = \sum_{i=1}^n (x_i \sin(i)) + \begin{cases} \sum_{i=1}^n \frac{1}{2}(x_i - |\cos(i)|)^2, & \text{se } x_i \leq |\cos(i)| \\ \sum_{i=1}^n \frac{1}{2}[x_i - 20|\cos(i)|]^2, & \text{se } x_i \geq |\cos(i)| \end{cases}.$$

Exemplo 3.8. Considere a função de Rosenbrock generalizada

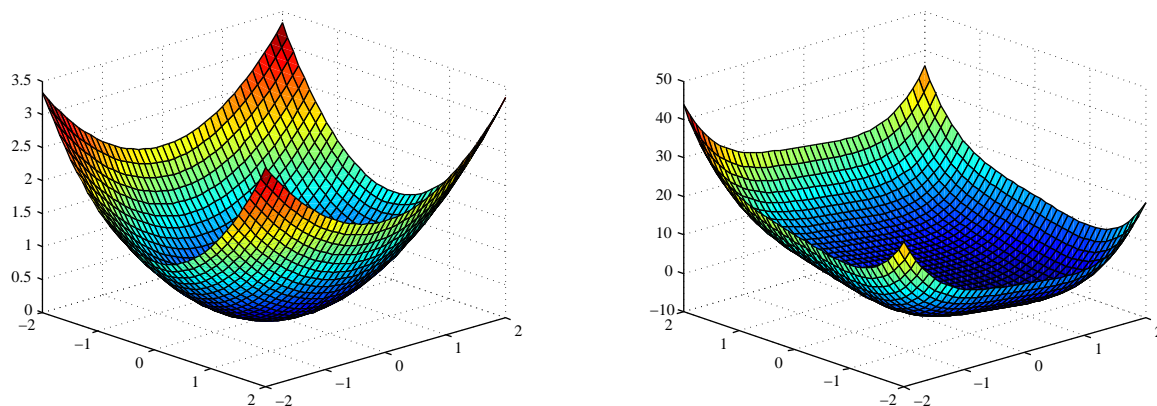


Figura 3.7: Funções dos Exemplos 3.5 (esquerda) e 3.6 (direita).

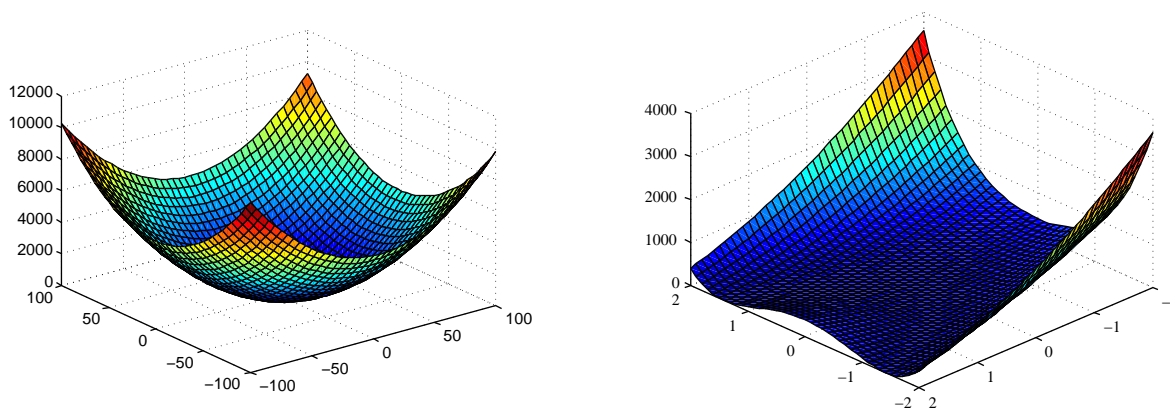


Figura 3.8: Funções dos Exemplos 3.7 (esquerda) e 3.8 (direita).

$$f(x) = \sum_{i=1}^{n-1} (c(x_{i+1} - x_i^2)^2 + (1 - x_i)^2).$$

Em nossos experimentos utilizamos `maxcg` = 10 e limitamos o número de iterações quase-Newton em `niqn` = 10, isto é, após 10 iterações quase-Newton realizamos uma iteração CG-Newton. Além disso, em cada iteração quase-Newton o iterado é avaliado quanto à redução no valor da função objetivo. Quando há redução, o iterado é aceito e continuamos o processo. Quando não há redução no valor da função, o iterado é descartado, realizamos uma iteração de ponto fixo (CGNewton) e reiniciamos o processo.

Como ponto inicial consideramos o vetor $x^0 \in \mathbb{R}^n$ dado por: $x_i^0 = \sin(i)$ para os Exemplos 3.5 e 3.6, $x_i^0 = 10 \sin(i)$ para o Exemplo 3.7 e $x^0 = (-1.2, 1, -1.2, 1, \dots)^\top$ para o Exemplo 3.8.

Analisando o comportamento do método CGNewton para os problemas acima temos que, para o Exemplo 3.5 o algoritmo utilizou a direção de Newton inexata em todas as iterações, exceto na última, apresentando uma taxa de convergência linear, indicando, portanto, que

podemos esperar um bom desempenho da aceleração quase-Newton.

Para o Exemplo 3.6, o método calculou a direção de Newton de maneira exata nas últimas iterações, exibindo convergência rápida nestas e, portanto, não esperamos grandes melhorias ao utilizar a estratégia de aceleração algorítmica.

Para o Exemplo 3.7, o algoritmo de gradientes conjugados encontrou direção de curvatura não positiva em praticamente todas as iterações, utilizando então a direção do gradiente. No entanto, apesar da direção oposta ao gradiente desfrutar de convergência linear, é uma opção razoável e, também, não temos a expectativa de grandes melhorias com o uso da aceleração quase-Newton.

O Exemplo 3.8 é um problema difícil de se resolver, por exemplo para $n = 2$, apresenta uma bacia de atração na direção da parábola $x_2 = x_1^2$. No entanto CGNewton comportou-se de maneira semelhante ao primeiro exemplo e podemos esperar um bom desempenho dos métodos acelerados.

Tabela 3.5: Comparativo entre os métodos testados para o Exemplo 3.5 com $n = 1000$ e $n = 5000$.

Método	Exemplo 3.5 com $n = 1000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	21	21	0	0	0.0213	2.10e-07	1.22e-12
MB1	15	15	3	1	0.0171	1.51e-07	4.65e-13
MB2	16	16	3	0	0.0180	1.72e-07	2.79e-13
MAC	16	16	3	1	0.0173	2.19e-07	4.95e-13
MACI	17	17	3	0	0.0173	1.31e-07	9.51e-13
MTH	14	14	3	0	0.0163	1.68e-07	5.27e-13
Método	Exemplo 3.5 com $n = 5000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	79	79	0	0	0.3917	2.01e-07	6.88e-12
MB1	33	33	11	4	0.1825	2.18e-07	1.22e-11
MB2	37	37	15	7	0.2031	2.07e-07	4.14e-12
MAC	35	35	15	7	0.1891	1.98e-07	4.05e-12
MACI	42	42	11	4	0.2157	1.82e-07	1.38e-11
MTH	35	35	15	7	0.1958	1.99e-07	6.25e-12

Na Tabela 3.5 reportamos o desempenho computacional dos métodos CGNewton e aceleração quase-Newton aplicados ao Exemplo 3.5. Na mesma, acrescentamos a notação:

NE: número de avaliações da função objetivo;

NCG: número de vezes em que o método mudou de uma iteração quase-Newton para uma iteração de ponto fixo, e vice-versa;

NNI: número de iterados quase-Newton não aceitos; ocorre quando o iterado não reduziu o valor da função objetivo. Neste caso retorna-se a uma iteração de ponto fixo;

‡: o algoritmo CGNewton ficou estagnado, isto é, não conseguiu satisfazer a busca linear com a direção encontrada.

No decorrer das Tabelas 3.6 a 3.8 apresentamos os resultados obtidos para os demais exemplos considerados.

Tabela 3.6: Comparativo entre os métodos testados para o Exemplo 3.6 com $n = 1000$ e $n = 5000$.

Método	Exemplo 3.6 com $n = 1000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	80	343	0	0	0.0808	1.30e-07	-1188.14
MB1	27 [‡]	219	12	9	0.0353	7.45e-04	-1188.16
MB2	87	396	48	25	0.1009	6.01e-08	-1188.14
MAC	38	144	11	6	0.0432	2.27e-10	-1188.14
MACI	70	314	35	21	0.0771	5.90e-08	-1188.14
MTH	27 [‡]	117	11	7	0.0322	2.32e-03	-1188.14
Método	Exemplo 3.6 com $n = 5000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	72 [‡]	335	0	0	0.3641	2.71e-03	-5950.34
MB1	23 [‡]	106	13	8	0.1326	1.08e-04	-5950.35
MB2	95	456	45	30	0.5476	1.23e-09	-5950.34
MAC	28 [‡]	142	7	4	0.1671	4.99e-02	-5950.36
MACI	66 [‡]	342	29	16	0.3650	1.38e-03	-5950.34
MTH	21	74	13	7	0.1221	1.34e-08	-5950.34

Tabela 3.7: Comparativo entre os métodos testados para o Exemplo 3.7 com $n = 1000$ e $n = 5000$.

Método	Exemplo 3.7 com $n = 1000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	586	41916	0	0	7.1125	6.67e-11	-3023.66
MB1	437	30899	245	291	5.3727	7.48e-11	-3023.66
MB2	457	32231	245	285	5.6453	1.56e-10	-3023.66
MAC	451	32345	208	158	5.5495	9.35e-07	-3023.66
MACI	518 [‡]	37085	265	217	6.3304	2.3370	-2751.52
MTH	427	30086	227	296	5.2334	5.07e-12	-3023.66
Método	Exemplo 3.7 com $n = 5000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	912 [‡]	70588	0	0	TMAX	0.11452	-14250.98
MB1	756	51254	417	502	45.224	6.03e-07	-15128.59
MB2	710	49007	415	439	43.317	8.82e-11	-15128.59
MAC	717 [‡]	46659	373	268	41.605	52.5094	-15128.59
MACI	823	59901	406	328	52.032	4.64e-10	-15128.59
MTH	754	50537	398	502	44.640	9.63e-07	-15128.59

Conforme esperado, para os Exemplos 3.5 e 3.8 todos os métodos quase-Newton apresentaram redução no número de iterações e tempo computacional, sem afetar a precisão da solução

Tabela 3.8: Comparativo entre os métodos testados para o Exemplo 3.8 com $n = 1000$ e $n = 5000$.

Método	Exemplo 3.8 com $n = 1000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	1915	3054	0	0	2.0610	3.28e-07	3.98662
MB1	1429	1784	743	975	1.7521	3.08e-07	1.57e-12
MB2	1584	1958	1128	743	1.9139	3.91e-07	5.03e-12
MAC	1424	1768	725	947	1.7726	1.45e-07	3.98662
MACI	1450	1755	856	892	1.7331	3.19e-07	3.98662
MTH	1419	1734	743	951	1.7914	7.42e-07	2.30e-11
Método	Exemplo 3.8 com $n = 5000$						
	NI	NE	NCG	NNI	TEMPO	RES	$f(x)$
CGN	9596	15463	0	0	51.1775	8.46e-07	3.98662
MB1	7056	8777	3717	4826	42.551	7.87e-07	3.98662
MB2	7767	9677	5604	3682	46.718	3.75e-07	4.59e-12
MAC	7016	8549	3458	4754	43.025	6.63e-07	3.98662
MACI	7100	8466	4108	4462	41.873	8.95e-08	3.98662
MTH	7048	8731	737	4777	43.760	4.01e-07	3.98662

encontrada quando comparados com o método CGNewton. Vale ressaltar que, no Exemplo 3.8, o método CGNewton não encontrou o minimizador global, enquanto que alguns métodos quase-Newton encontraram. As Figuras 3.9 e 3.12 enfatizam este resultado.

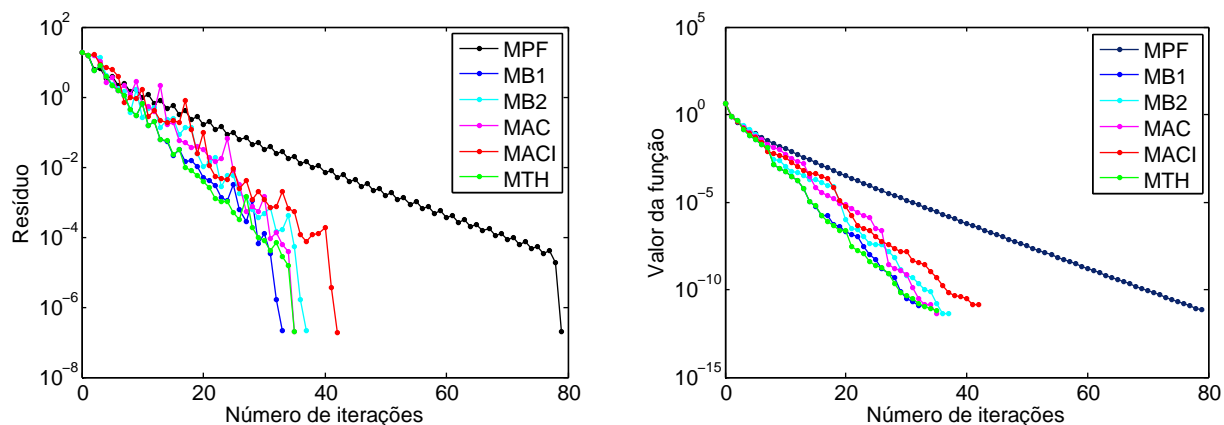


Figura 3.9: Taxa de convergência dos métodos testados para o Exemplo 3.5 com $n = 5000$.

Para o Exemplo 3.6, alguns métodos não conseguiram satisfazer o critério de busca linear de CGNewton e ficaram estagnados em um ponto próximo de uma solução, incluindo o método original para $n = 5000$. Tal comportamento pode ser observado também na Figura 3.10, onde constatamos que a solução encontrada pelos métodos possui um valor de função bem próximo do valor satisfeito no minimizador, porém, em alguns casos, o valor do resíduo foi maior do que a precisão desejada.

Para o Exemplo 3.7, MAC e MACI apresentaram dificuldades; o primeiro ficou estagnado

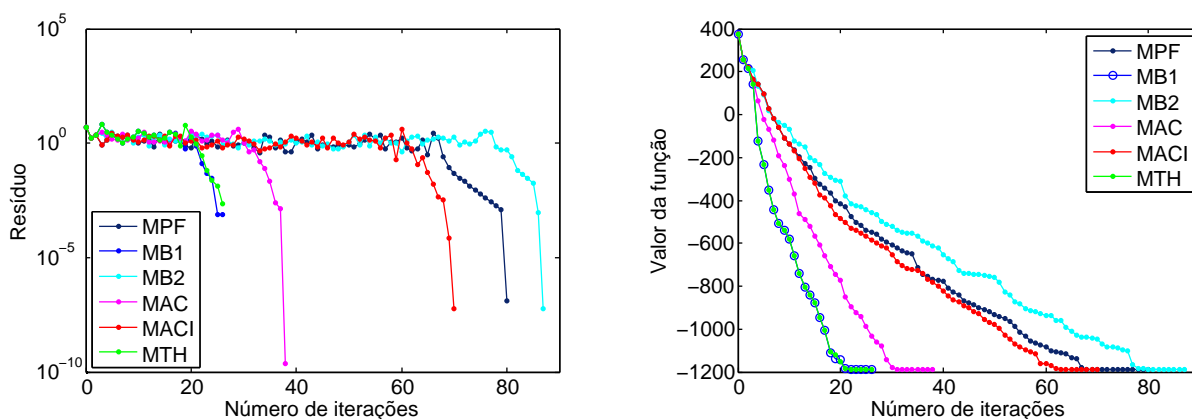


Figura 3.10: Taxa de convergência dos métodos testados para o Exemplo 3.6 com $n = 1000$.

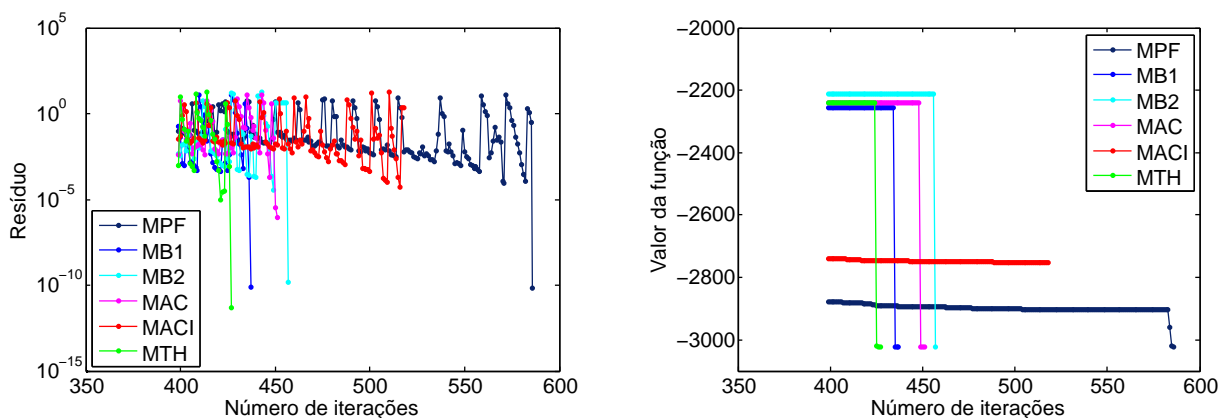


Figura 3.11: Taxa de convergência dos métodos testados para o Exemplo 3.7 com $n = 1000$.

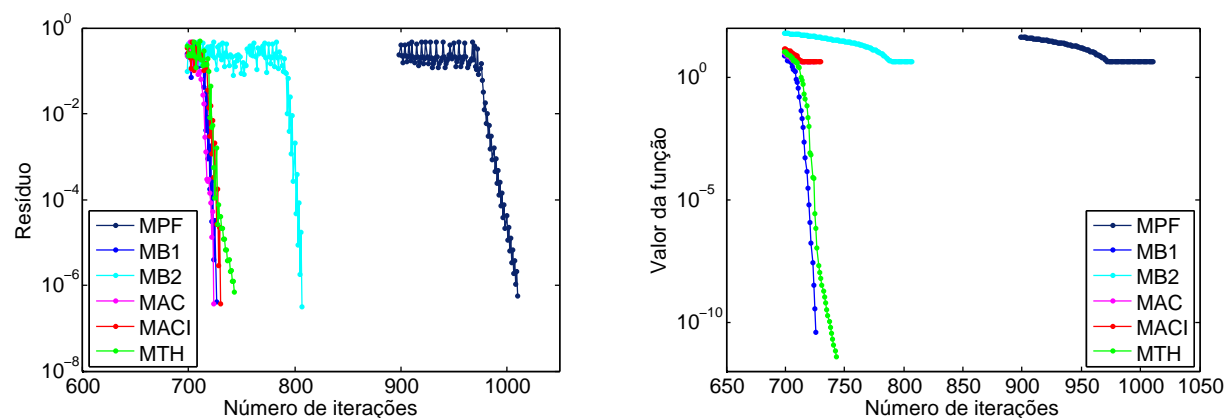


Figura 3.12: Taxa de convergência dos métodos testados para o Exemplo 3.8 com $n = 500$.

quando $n = 5000$, e o último satisfaz o segundo critério de parada quando $n = 1000$. Os demais métodos quase-Newton considerados apresentaram bons resultados, reduzindo o número de iterações sem afetar a qualidade da solução. A Figura 3.11 evidencia tais conclusões.

De modo geral, considerando todos os casos de sucesso, a estratégia de aceleração quase-

Newton alcançou uma redução média de 28% no número de iterações.

3.4 Conclusões

Pelo acima exposto, concluímos que a estratégia quase-Newton mostra-se muito eficiente quando trabalhamos com problemas de ponto fixo com taxa de convergência linear. Adicionalmente, consegue sempre aprimorar os resultados, exceto para casos específicos. Evidenciamos que para o Exemplo 3.3 os métodos quase-Newton apresentaram dificuldades em manter-se no conjunto viável, o que provocou os resultados desfavoráveis. Portanto, o uso da técnica estudada em problemas que apresentam tal característica não é recomendado.

Capítulo 4

Aplicação da estratégia quase-Newton de aceleração algorítmica em PET

4.1 Introdução

Neste capítulo vamos aplicar a estratégia de aceleração quase-Newton vista no capítulo anterior ao problema de reconstruir o comportamento cinético da água radioativa, $H_2^{15}O$, durante o exame de perfusão cardíaca.

Para este fim, utilizamos dados provenientes de exames de Tomografia por Emissão de Pósitrons (*Positron Emission Tomography* - PET) e adotamos o algoritmo desenvolvido por Reips [61]. Além da reconstrução da imagem que representa o comportamento cinético de $H_2^{15}O$, Reips [61] também propõe uma formulação do problema de identificação dos parâmetros associados ao problema inverso de reconstruir tal imagem. Este problema será resolvido para reconstruirmos os parâmetros cinéticos que compõe o modelo de equações diferenciais em que estamos interessados.

A Tomografia por Emissão de Pósitrons consiste em um exame de imagem aplicado em medicina nuclear capaz de produzir imagens dos processos fisiológicos em 2D ou 3D. O procedimento é simples e consiste em injetar glicose ligada a um elemento radioativo no paciente via circulação sanguínea. O elemento radioativo se espalha pelo sangue e regiões que metabolizam excesso de glicose (como tumores, por exemplo) são identificadas na imagem criada pelo computador. Um escâner típico utilizado em exames PET é mostrado na Figura 4.1.¹

Ainda de acordo com [61], a emissão de pósitrons ocorre com o decaimento dos isótopos do marcador radioativo. Os pósitrons são antimatéria dos elétrons, assim o aniquilamento entre pósitrons e elétrons produz um par de fótons gama que se deslocam em direções opostas. Esses fótons são detectados pelo escâner PET. Cada par de detectores define uma linha ao longo da qual a intensidade da aniquilação é medida. Essas intensidades podem ser descritas via integrais de linha e os dados são armazenados por um sinograma antes da reconstrução. Esse processo está esquematizado na Figura 4.2.²

¹<https://upload.wikimedia.org/wikipedia/commons/b/b8/ECAT-Exact-HR-PET-Scanner.jpg> (acessado em 17/09/2015).

² <https://upload.wikimedia.org/wikipedia/commons/c/c1/PET-schema.png> (acessado em 17/09/2015).



Figura 4.1: Um escâner típico de Tomografia por Emissão de Pósitrons (PET). ©Wikipedia

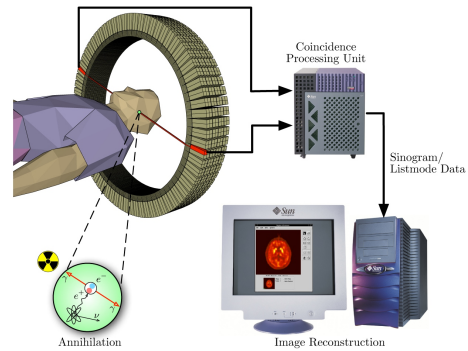


Figura 4.2: Esquema do processo do exame PET. ©Wikipedia

Com os dados $f(t)$ do sinograma PET, o problema de gerar a imagem u a partir destes dados consiste em resolver

$$f(t) = \xi(Ku(t)), \quad u = G(p)$$

em que ξ representa a medida de Poisson (estatística) e K é o operador de observação.

A solução para este problema inverso é dada pelo problema de minimização a seguir

$$(u, p) \in \arg \min_{(u, p)} \int_0^T \int_{\Omega} [Ku(t) - f(t) \log(Ku(t))] dy dt + \alpha \mathcal{R}(p) \quad (4.1)$$

em que \mathcal{R} é uma regularização e penaliza o desvio com relação a uma solução suave u .

Resolvendo o problema de minimização acima encontramos todos os parâmetros biológicos que compõem p . Por fim, utilizando o modelo de equações diferenciais proposto em [61] podemos prever o comportamento do fluxo do marcador radioativo durante o exame de perfusão cardíaca.

Detalhes sobre a resolução do problema de minimização (4.1) fogem ao escopo deste trabalho, para uma leitura detalhada recomendamos a tese de Reips [61].

4.2 Aplicação da aceleração quase-Newton

Os resultados obtidos em [61] indicam que o modelo proposto por Reips prevê com boa precisão a atividade do marcador radioativo. No entanto, o elevado número de iterações com um alto custo computacional tornam o método pouco competitivo.

De acordo com [7, 18] é interessante acelerar a convergência de problemas de ponto fixo de grande dimensão, em que a função F (definida em (3.2)) é continuamente diferenciável (mas a derivada não está disponível ou é muito cara para calcular), o custo de avaliar a função F é alto e os valores avaliados de $F(x)$ podem apresentar erros. À vista disso, o modelo proposto em [61] constitui o caso ideal para aplicarmos algum tipo de aceleração.

Para resolver (4.1) Reips trabalhou com o algoritmo de Maximização de Esperança (*Expectation Maximization* - EM) e com o problema de identificação de parâmetros. Assim, dada uma imagem u_k estima-se a imagem $u_{k+\frac{1}{2}}$ via algoritmo EM, a saber

$$u_{k+\frac{1}{2}} = \frac{u_k}{K^* \mathbf{1}} K^* \left(\frac{f}{Ku_k} \right)$$

em que $\mathbf{1}$ denota o vetor unitário e K^* o operador adjunto de K .

A partir da imagem $u_{k+\frac{1}{2}}$ é possível identificar todos os parâmetros biológicos $p_{k+\frac{1}{2}}$ utilizando o processo de identificação de parâmetros. A nova imagem u_{k+1} é obtida a partir dos parâmetros encontrados e o processo recomeça.

As variáveis consideradas no modelo são os parâmetros biológicos, isto é,

$$p = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ V_{xA} \\ V_{yA} \\ V_{xT} \\ V_{yT} \\ V_{xV} \\ V_{yV} \\ D_A \\ D_T \\ D_V \end{pmatrix} \quad (4.2)$$

em que k_1 , k_2 e k_3 , representam a troca de fluidos entre artéria, tecido (capilares) e veia, V_A , V_T e V_V são os parâmetros da velocidade, e finalmente, D_A , D_T e D_V os parâmetros de difusão na artéria, tecido e veias, respectivamente. Os parâmetros D_A , D_T , D_V e k_i , $i = 1, 2, 3$, são não negativos [61].

Todos os parâmetros são funções das coordenadas espaciais (x, y) , independentes em relação ao tempo. Para a escala com relação à coordenada x foi usada uma grade $G_x = \{x_i = ih_x - 1 : i = 0, \dots, 2n\}$ com tamanho de passo h_x . Analogamente, para a escala y foi utilizada a grade $G_y = \{y_j = jh_y : j = 0, \dots, m\}$ com tamanho de passo h_y . Note que, apesar das variáveis do problema estarem colocadas num contexto matricial, podemos reescrevê-las como vetores e aplicar o algoritmo estudado.

Dessa forma, uma iteração de ponto fixo está definida por uma iteração do método *Forward-Backward Splitting* [11] que atualiza o vetor de parâmetros p .

Na sequência definimos os problemas considerados e expomos os resultados obtidos.

4.3 Resultados numéricos e Problemas

O algoritmo descrito em [61] foi primeiro transcrito para Fortran 77 com dupla precisão. Os experimentos foram então realizados usando gfortran-4.6 em uma máquina Intel CORE I3-2310M @ 2.10 GHz com 100 Gb de HD e 4Gb de Ram. Como critério de parada utilizamos a medida de sensibilidade utilizada em [61], a saber

$$\frac{\|p_i^{k+1} - p_i^k\|_F}{\|p_i^{k+1}\|_F} < 10^{-4}$$

em que p_i^k representa uma componente do vetor p descrito em (4.2) na iteração k e $\|\cdot\|_F$ denota a norma de Frobenius.

Além do critério colocado acima, limitamos o número de iterações em $\max\{100, n^2\}$ e o tempo computacional (para os métodos quase-Newton) em 600 segundos.

Para todos os exemplos analisados, vamos considerar um operador K (16512×4225) associado à imagem PET real dada pela figura a seguir.

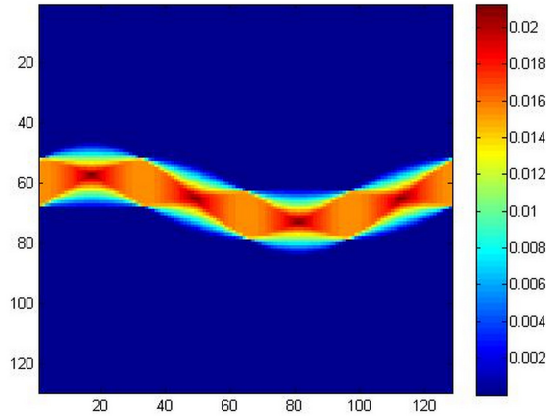


Figura 4.3: Operador de observação K de um escâner PET real.

Usaremos uma imagem 65×65 em um domínio Ω . Para a concentração inicial do marcador radioativo na artéria, C_A , usamos

$$C_A(x, y, 0) = \tau(1 - x^2)(N - y)y. \quad (4.3)$$

Utilizamos $N = 50$, caracterizando a imagem exposta na Figura 4.4.

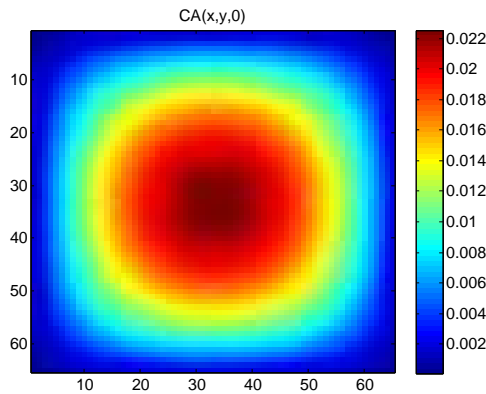


Figura 4.4: Concentração inicial do marcador radioativo na artéria.

Para as concentrações iniciais no tecido e veias consideramos

$$C_T(x, y, 0) = 0 \text{ e } C_V(x, y, 0) = 0$$

e, para o tamanho de passo no tempo utilizamos $\tau = 3 \cdot 10^{-5}$.

A seguir definiremos os demais parâmetros dos problemas e apresentamos os resultados obtidos.

Exemplo 4.1. Para este primeiro exemplo, os valores iniciais dos parâmetros biológicos considerados estão definidos na Tabela 4.1, em que α corresponde à regularização *a priori* e ξ à regularização do gradiente, associadas a cada um dos parâmetros. O valor da coluna $(\cdot)^*$ representa o valor típico do parâmetro utilizado na regularização *a priori* [61].

Tabela 4.1: Dados iniciais para o Exemplo 4.1.

Parâmetro	Valor Inicial	$(\cdot)^*$	Regul. <i>a priori</i> (α)	Regul. Grad. (ξ)
k_1 (1/cm)	0.90 (0)	0.89	0.01287520644013148965	0.0008
k_2 (1/cm)	0.75	0.70	0.01286792647011880155	0.0001
k_3 (1/cm)	0.90	0.85	0.01287621626481284896	0.0001
V_{xA} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yA} (cm/s)	700.0	15.0	1.1000	0.0001
V_{xT} (cm/s)	-50.0	-5.0	1.122098745999	0.0001
V_{yT} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{xV} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yV} (cm/s)	700.0	15.0	1.1000000001	0.0001
D_A (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444
D_T (cm ² /s)	3e-6	10 ⁻²	0.000344	0.000444
D_V (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444

Também avaliamos o comportamento do fluxo radioativo quando k_1 é igual a zero para algum intervalo e, por este motivo, na Tabela 4.1, o símbolo (0) representa o fato de k_1 não ser constante em toda a região de interesse. Quando $k_1 = 0$ não há troca de materiais entre artéria e tecido, o que significa que a concentração do marcador radioativo nessa região do tecido é nula.

Aplicando a aceleração quase-Newton apresentada no capítulo anterior conseguimos reduzir significativamente o número de iterações e, conseqüentemente, o tempo computacional. A Tabela 4.2 descreve o comportamento dos métodos de ponto fixo (FBS) e aceleração quase-Newton aplicados ao Exemplo 4.1. Nesta tabela, utilizamos a seguinte notação:

NI: número de iterações;

TEMPO: tempo computacional (em segundos) medido pela função `etime`;

RES: o resíduo encontrado, isto é, $\|F(\bar{x})\|$ em que \bar{x} é a solução encontrada pelo algoritmo;

SENS: medida de sensibilidade utilizada como critério de parada.

As imagens expostas na Figura 4.5 referem-se ao parâmetro V_{xT} calculado via método FBS e MB2 (Segundo Método de Broyden). Observe que, apesar de as imagens serem visualmente muito semelhantes, as mesmas apresentaram valores diferentes. Sob o ponto de vista teórico, os métodos quase-Newton melhoraram a solução obtida, uma vez que a imagem encontrada está mais próxima de um ponto fixo do problema. No entanto, sob o ponto de vista prático, a solução

Tabela 4.2: Comparativo entre os métodos testados para o Exemplo 4.1.

	$t_k = 1$			
	NI	TEMPO	RES	SENS
FBS	317	4138.5	6.6958	9.99e-5
MB1	2	53.699	7.97e-3	5.51e-5
MB2	2	58.959	7.97e-3	5.52e-5
MAC	2	59.827	9.29e-3	5.54e-5
MACI	2	60.703	1.46e-2	5.58e-5
MTH	2	59.827	7.97e-3	5.51e-5

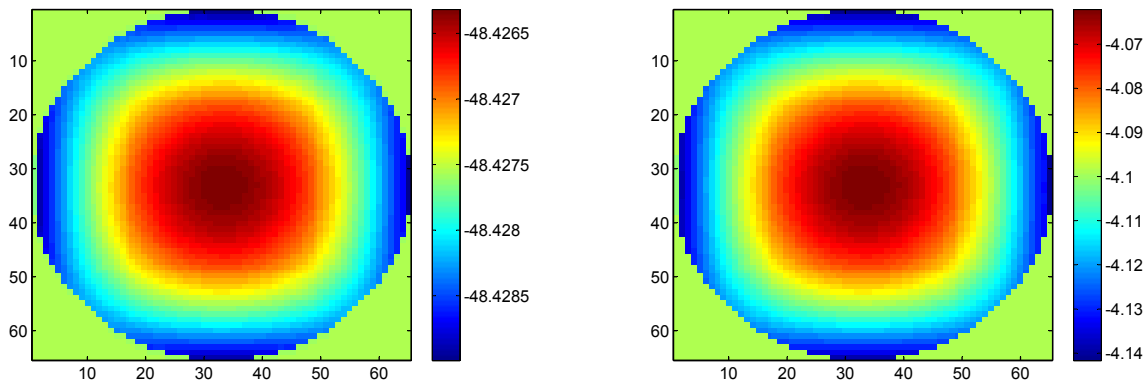
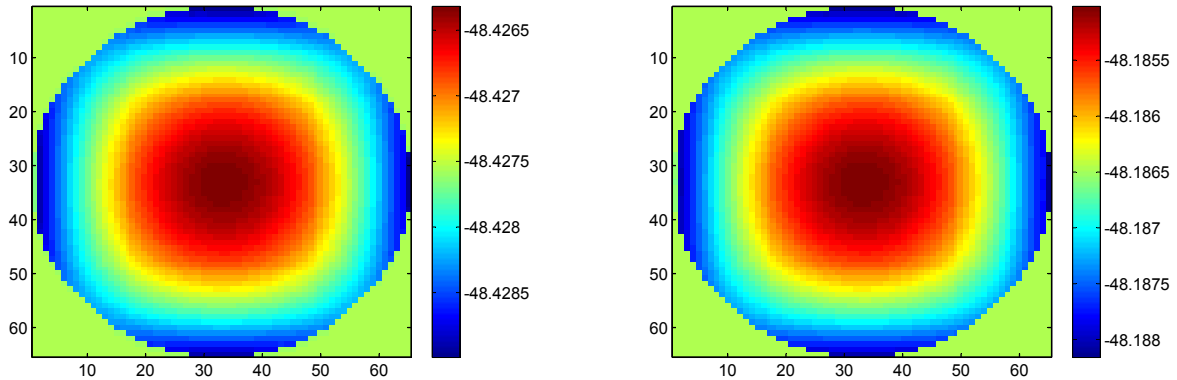


Figura 4.5: Imagens geradas pelos métodos FBS (esquerda) e MB2 (direita).

Figura 4.6: Imagens geradas pelos métodos FBS (esquerda) e MB2 com $t_k = 0.01$ (direita).

ideal é aquela que se assemelha à imagem obtida via método *Forward-Backward Splitting*. Por este motivo decidimos limitar o tamanho do passo em cada iteração quase-Newton por t_k , assim

$$p^{k+1} = p^k + t_k s^k$$

em que s^k é a direção quase-Newton. Neste primeiro momento utilizamos $t_k = 0.1$ e $t_k = 0.01$, o valor $t_k = 1$ representa o passo quase-Newton puro. Após essa correção de passo, conseguimos

aproximar muito bem as imagens obtidas, conforme exemplificado para o parâmetro $V_{x\mathcal{T}}$ na Figura 4.6.

Na Tabela 4.3 apresentamos o comportamento dos métodos quase-Newton para o Exemplo 4.1 para os tamanhos de passo $t_k = 0.1$ e $t_k = 0.01$. O resultado para o método FBS foi omitido uma vez que independe do tamanho de passo t_k .

Tabela 4.3: Comparativo entre os métodos testados para o Exemplo 4.1.

	$t_k = 0.1$				$t_k = 0.01$			
	NI	TEMPO	RES	SENS	NI	TEMPO	RES	SENS
MB1	2	53.723	6.2394	9.62e-5	5	93.317	6.6595	9.96e-5
MB2	2	53.367	6.2394	9.62e-5	5	102.31	6.6594	9.92e-5
MAC	2	52.911	6.2399	9.62e-5	5	91.401	6.6597	9.96e-5
MACI	2	52.995	6.2406	9.62e-5	5	91.325	6.6600	9.96e-5
MTH	2	52.947	6.2394	9.62e-5	5	91.273	6.6595	9.96e-5

Nas Figuras 4.7 e 4.8 expomos o gráfico de convergência dos métodos testados para o Exemplo 4.1. Observe que, mesmo restringindo o tamanho de passo para adequar a solução obtida pelo algoritmo ao problema PET, a redução no número de iterações e no tempo computacional foi de aproximadamente 98% e, mais importante, mantendo a qualidade da solução.

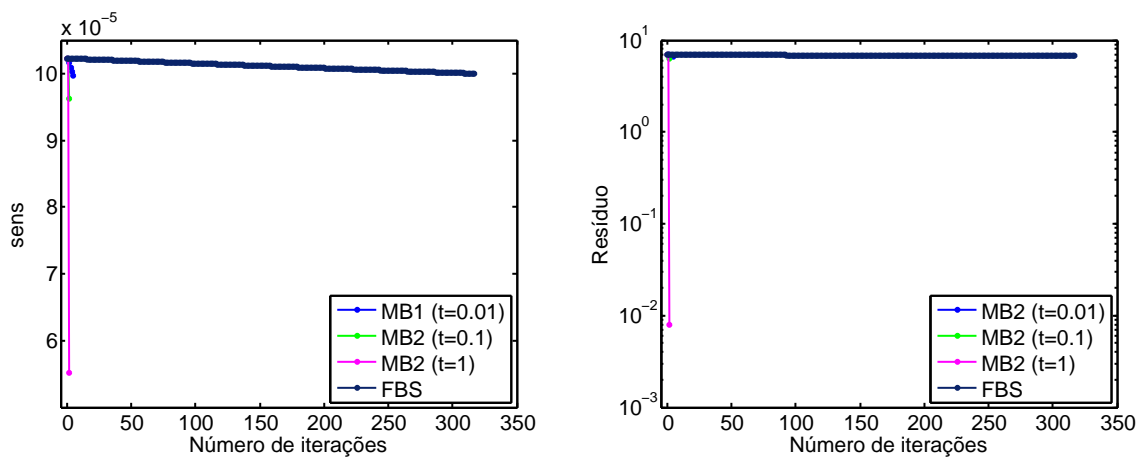


Figura 4.7: Gráfico de convergência dos métodos testados para o Exemplo 4.1.

Na sequência, apresentamos as figuras que representam a reconstrução dos parâmetros biológicos para o Exemplo 4.1 utilizando o método MB2. A reconstrução dos parâmetros k_2 e k_3 foi omitida uma vez que esses parâmetros foram constantes, com valores $k_2 \approx 0.7278$ e $k_3 \approx 0.8733$.

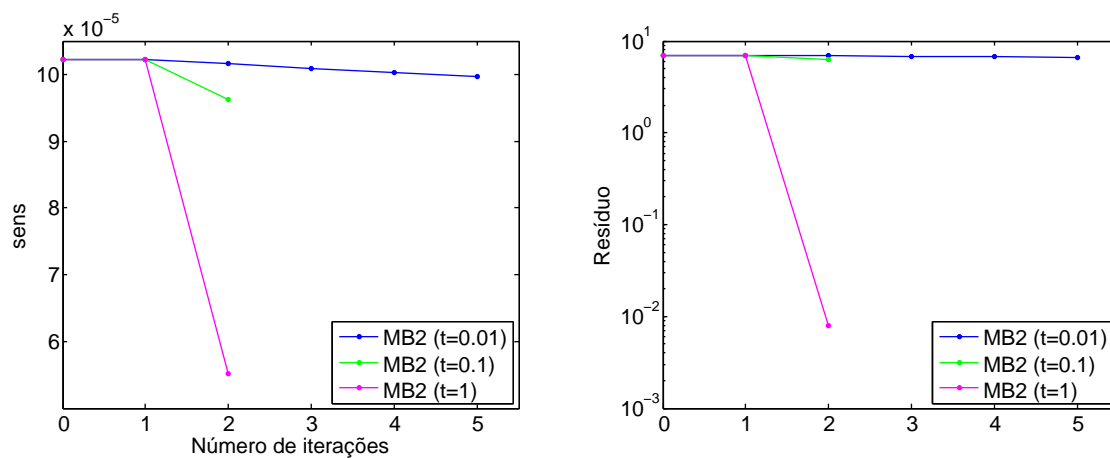


Figura 4.8: Gráfico de convergência do método MB2 com diferentes tamanhos de passo para o Exemplo 4.1.

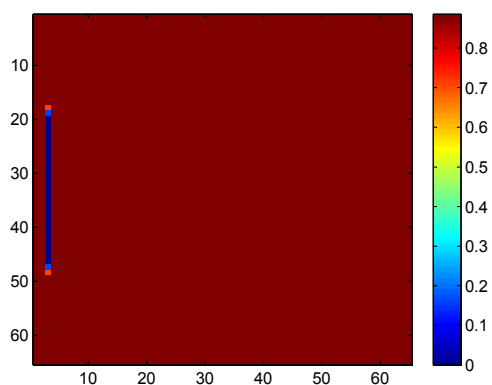


Figura 4.9: Reconstrução de k_1 .

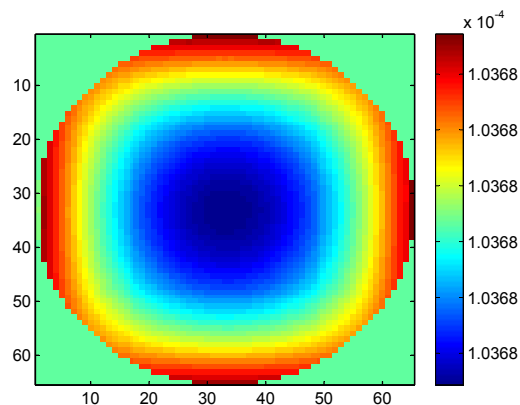


Figura 4.10: Reconstrução de $V_{x,A}$.

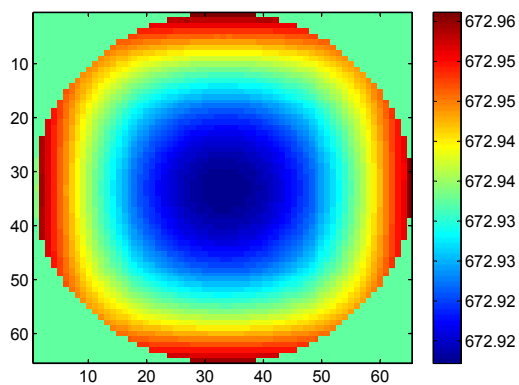


Figura 4.11: Reconstrução de $V_{y,A}$.

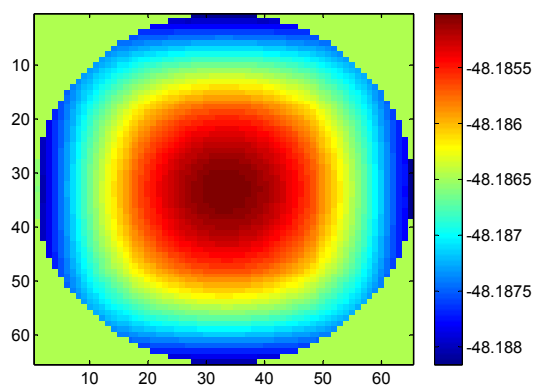


Figura 4.12: Reconstrução de $V_{x,T}$.

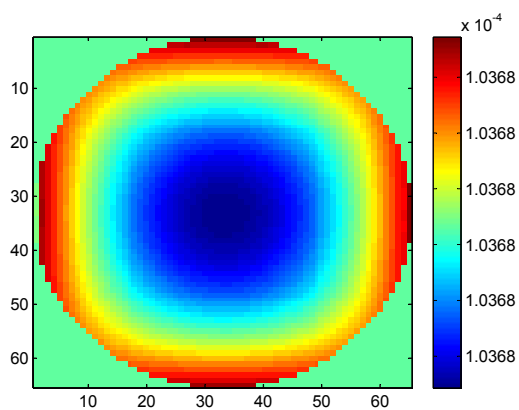


Figura 4.13: Reconstrução de $V_{y\mathcal{T}}$.

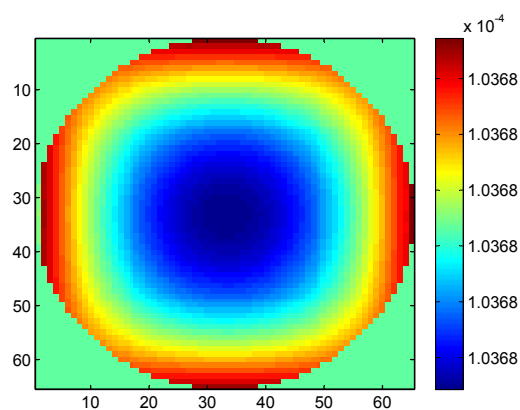


Figura 4.14: Reconstrução de $V_{x\mathcal{V}}$.

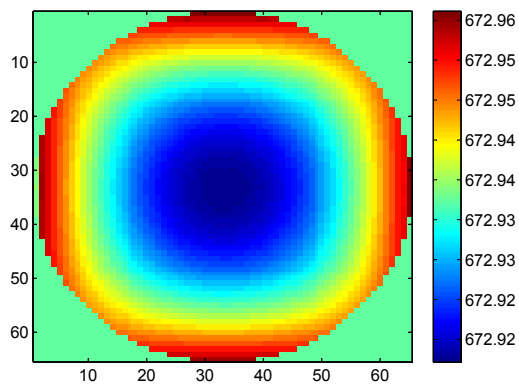


Figura 4.15: Reconstrução de $V_{y\mathcal{V}}$.

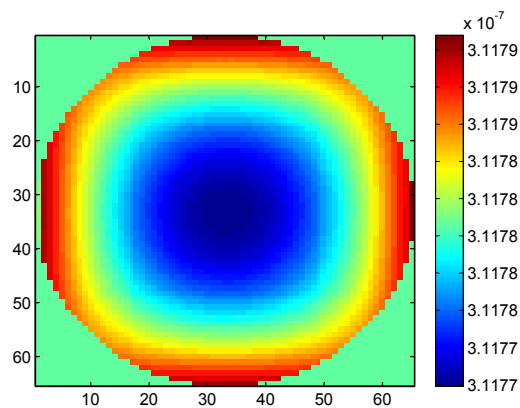


Figura 4.16: Reconstrução de $D_{x\mathcal{A}}$.

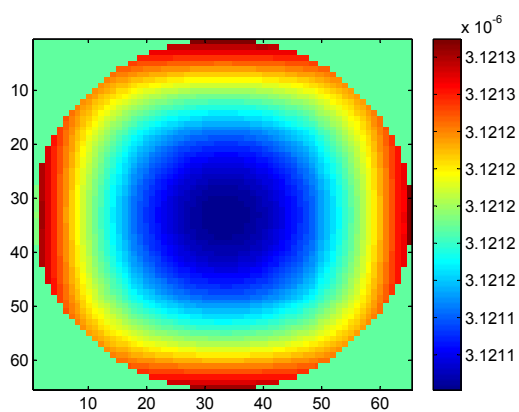


Figura 4.17: Reconstrução de $D_{x\mathcal{T}}$.

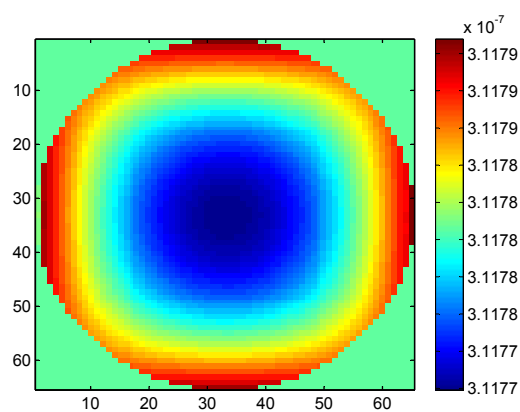


Figura 4.18: Reconstrução de $D_{x\mathcal{V}}$.

Exemplo 4.2. Para este exemplo realizamos pequenas mudanças nos dados iniciais com o objetivo de deixar o experimento mais difícil que o anterior. Os novos valores iniciais para os parâmetros biológicos estão definidos na Tabela 4.4.

Tabela 4.4: Dados iniciais para o Exemplo 4.2.

Parâmetro	Valor Inicial	(\cdot) [*]	Regul. <i>a priori</i> (α)	Regul. Grad. (ξ)
k_1 (1/cm)	0.90 (0)	0.89	0.017148965	0.0008
k_2 (1/cm)	0.75 (0)	0.70	0.015801553	0.0001
k_3 (1/cm)	0.01	0.85	0.050148965	0.0001
V_{xA} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yA} (cm/s)	700.0	15.0	1.1620587	0.0001
V_{xT} (cm/s)	-50.0	-5.0	1.2651822098745999	0.0001
V_{yT} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{xV} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yV} (cm/s)	700.0	15.0	1.1620587	0.0001
D_A (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444
D_T (cm ² /s)	3e-6	10 ⁻²	0.000344	0.000444
D_V (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444

Na Tabela 4.5 reportamos o desempenho computacional dos métodos FBS e aceleração quase-Newton aplicados ao Exemplo 4.2.

Tabela 4.5: Comparativo entre os métodos testados para o Exemplo 4.2.

	$t_k = 0.1$				$t_k = 0.01$			
	NI	TEMPO	RES	SENS	NI	TEMPO	RES	SENS
FBS	910	11757.4	6.5904	9.99e-5	910	11757.4	6.5904	9.99e-5
MB1	2	52.5872	6.5924	9.88e-5	7	118.815	6.8929	7.25e-5
MB2	3	70.5364	5.9331	9.42e-5	12	189.211	6.5582	9.97e-5
MAC	3	70.5684	5.9338	9.43e-5	12	186.275	6.5586	9.97e-5
MACI	3	68.0482	5.9351	9.43e-5	12	184.315	6.5593	9.97e-5
MTH	3	66.8961	5.9331	9.42e-5	12	188.375	6.5582	9.97e-5

Os resultados obtidos confirmam que a dificuldade do problema aumentou em relação ao exemplo anterior, no entanto os métodos quase-Newton mantiveram o bom desempenho, reduzindo em torno de 98% o número de iterações e tempo computacional, mantendo a precisão desejada. As Figuras 4.19 e 4.20 reforçam esse entendimento.

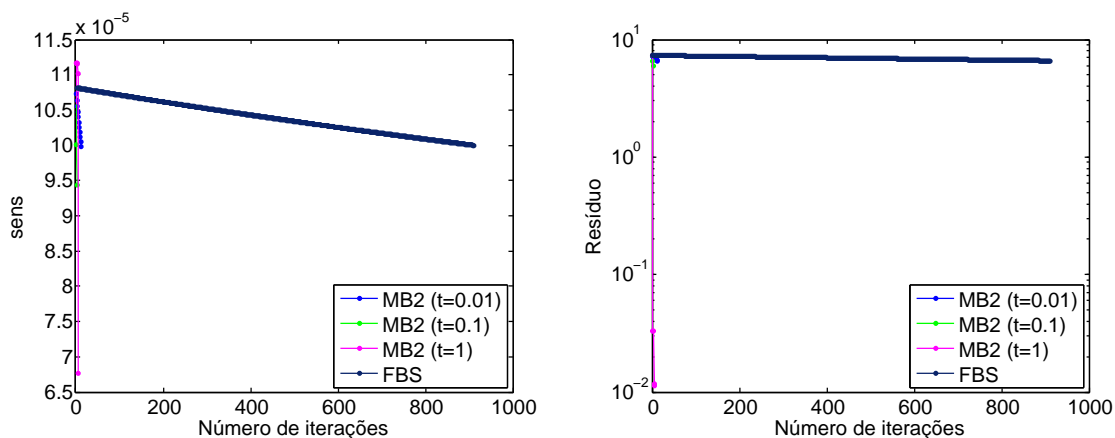


Figura 4.19: Gráfico de convergência dos métodos testados para o Exemplo 4.2.

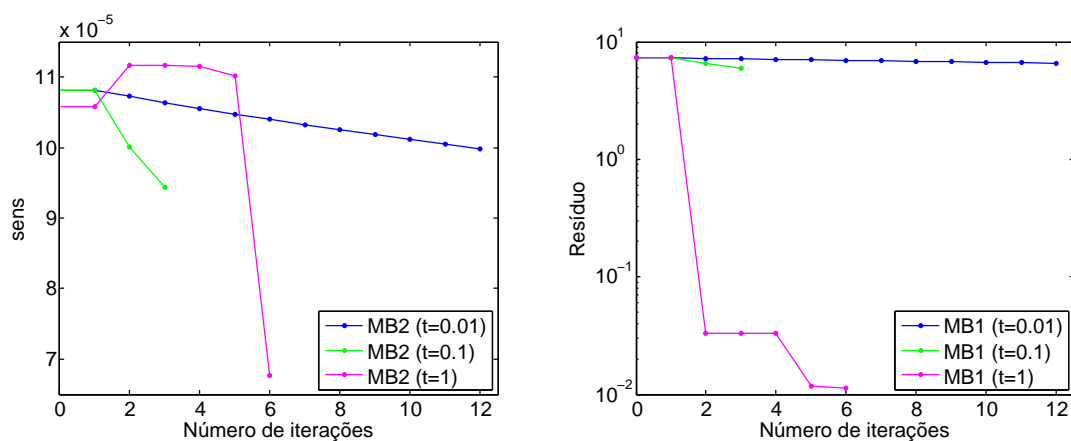


Figura 4.20: Gráfico de convergência do método MB2 com diferentes tamanhos de passo para o Exemplo 4.2.

A seguir expomos as imagens referentes à reconstrução do parâmetro k_2 e dos parâmetros que apresentaram maior dificuldade em aproximar os valores obtidos via método FBS, a saber, V_{yA} , V_{xT} e V_{yV} .

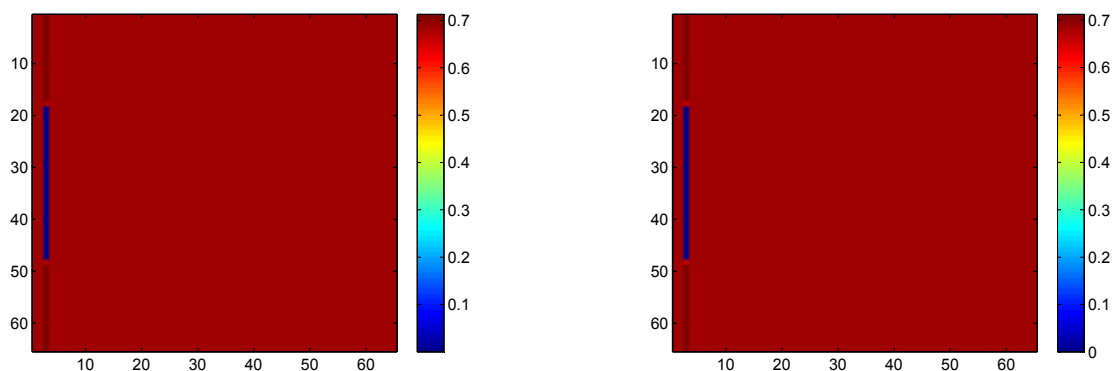


Figura 4.21: Reconstrução do parâmetro k_2 pelo método FBS (esquerda) e MB2 (direita).

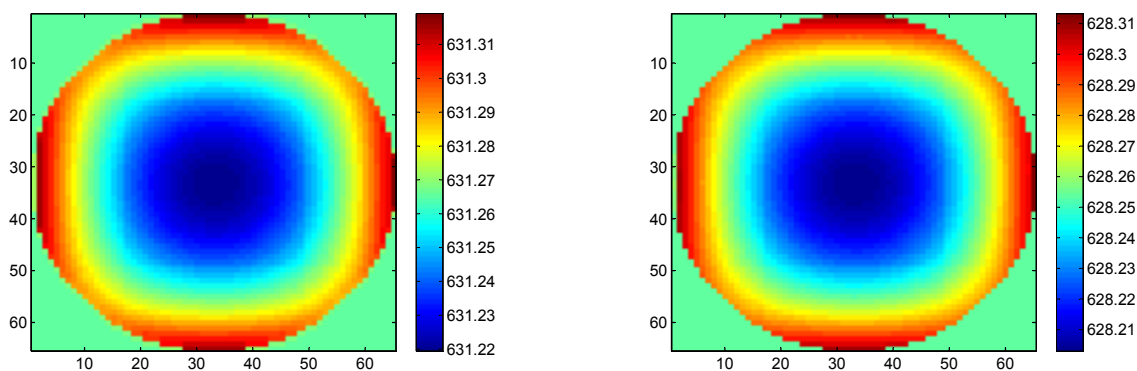


Figura 4.22: Reconstrução do parâmetro V_{yA} pelo método FBS (esquerda) e MB2 (direita).

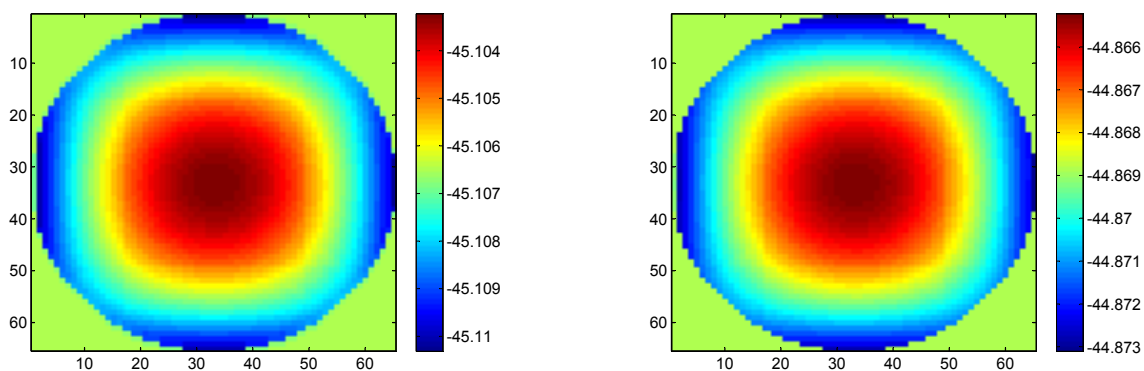


Figura 4.23: Reconstrução do parâmetro V_{xT} pelo método FBS (esquerda) e MB2 (direita).

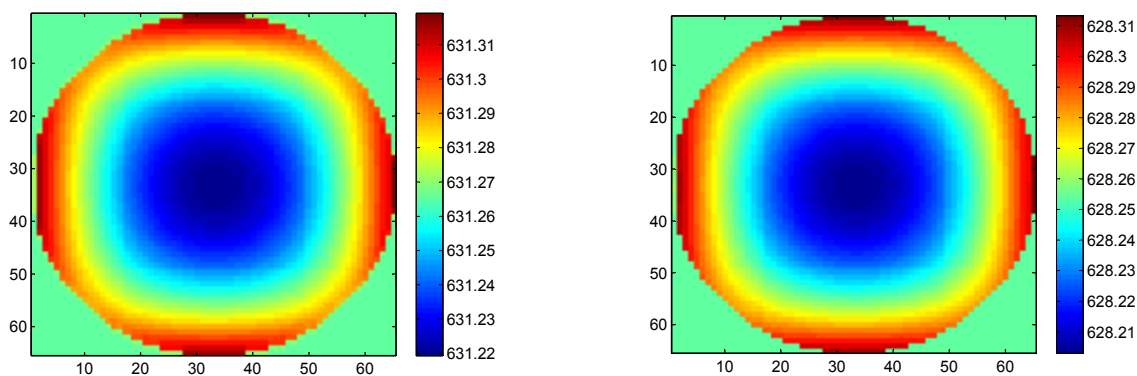


Figura 4.24: Reconstrução do parâmetro V_{yV} pelo método FBS (esquerda) e MB2 (direita).

Exemplo 4.3. Neste último exemplo vamos analisar o comportamento do marcador radioativo considerando um caso um pouco diferente. Os valores iniciais correspondentes aos parâmetros

k_1 e k_2 apresentam zeros na região central, o que visualmente corresponde às lacunas da Figura 4.25. Com essa definição, estamos considerando a presença de algo que impede a passagem do fluxo, como um tumor, por exemplo. Os novos valores iniciais estão definidos na Tabela 4.6.

Tabela 4.6: Dados iniciais para o Exemplo 4.3.

Parâmetro	Valor Inicial	$(\cdot)^*$	Regul. <i>a priori</i> (α)	Regul. Grad. (ξ)
k_1 (1/cm)	0.90 (0)	0.89	0.017148965	0.0008
k_2 (1/cm)	0.75 (0)	0.70	0.016801553	0.0001
k_3 (1/cm)	0.01	0.85	0.051822197678965	0.0001
V_{xA} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yA} (cm/s)	700.0	15.0	1.1	0.0001
V_{xT} (cm/s)	-50.0	-5.0	1.22098745999	0.0001
V_{yT} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{xV} (cm/s)	1e-4	0.10	0.001024495	0.0001
V_{yV} (cm/s)	700.0	15.0	1.1000000001	0.0001
D_A (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444
D_T (cm ² /s)	3e-6	10 ⁻²	0.000344	0.000444
D_V (cm ² /s)	3e-7	10 ⁻³	0.0003344	0.000444

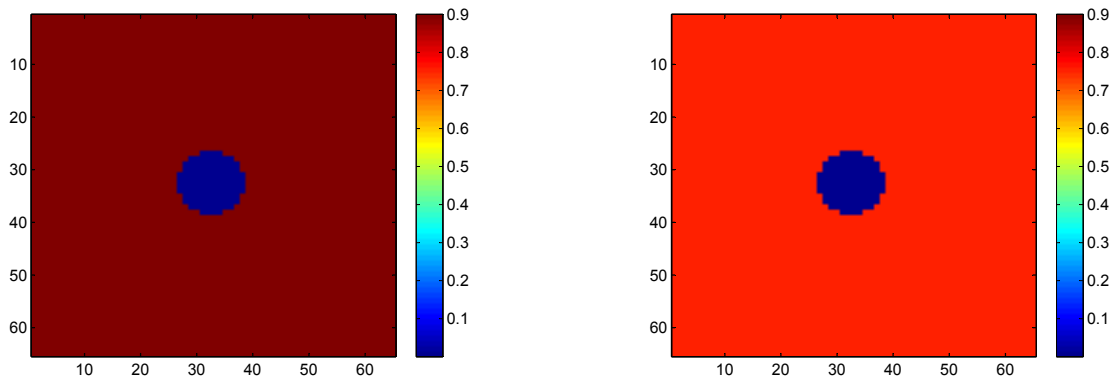


Figura 4.25: Parâmetros k_1 (esquerda) e k_2 (direita) iniciais para o Exemplo 4.3.

Na Tabela 4.7 apresentamos o comportamento do método FBS e dos métodos quase-Newton para o Exemplo 4.3.

Novamente, os resultados obtidos mostraram uma redução significativa no número de iterações e, conseqüentemente, no tempo computacional. Tal redução foi de aproximadamente 98% para $t_k = 0.01$, o tamanho de passo que melhor aproximou os resultados obtidos via método FBS. E, como podemos notar nas Figuras 4.26 e 4.27, a redução alcançada não prejudicou a qualidade da solução.

Tabela 4.7: Comparativo entre os métodos testados para o Exemplo 4.3.

	$t_k = 0.1$				$t_k = 0.01$			
	NI	TEMPO	RES	SENS	NI	TEMPO	RES	SENS
FBS	833	10734.3	6.3275	9.99e-5	833	10734.3	6.3275	9.99e-5
MB1	2	52.6392	6.2407	9.93e-5	12	181.579	6.2069	9.61e-5
MB2	2	53.1473	6.2407	9.93e-5	11	221.853	6.2711	9.95e-5
MAC	2	52.6112	6.2410	9.93e-5	11	168.474	6.2714	9.95e-5
MACI	2	52.6392	6.2417	9.93e-5	11	169.674	6.2721	9.96e-5
MTH	2	53.0833	6.2407	9.93e-5	11	170.190	6.2711	9.95e-5

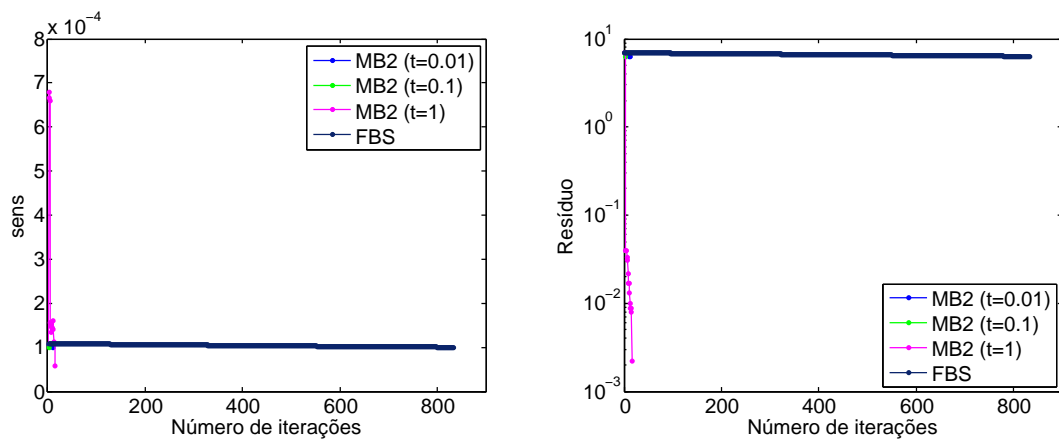


Figura 4.26: Gráfico de convergência dos métodos testados para o Exemplo 4.3.

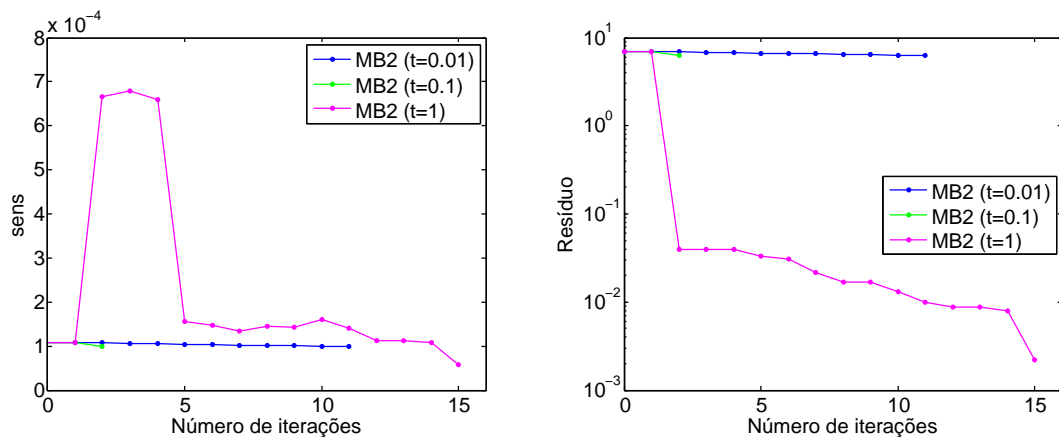


Figura 4.27: Gráfico de convergência do método MB2 com diferentes tamanhos de passo para o Exemplo 4.3.

Na sequência comparamos as imagens referentes à reconstrução dos parâmetros k_1 e $V_{x\mathcal{T}}$, dois dos parâmetros que apresentaram maior dificuldade em aproximar os valores obtidos via método FBS.

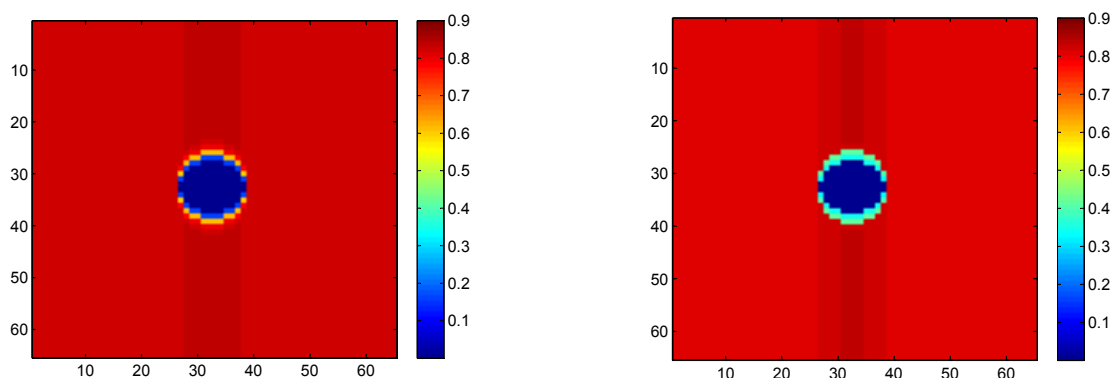


Figura 4.28: Reconstrução do parâmetro k_1 pelo método FBS (esquerda) e MB2 (direita).

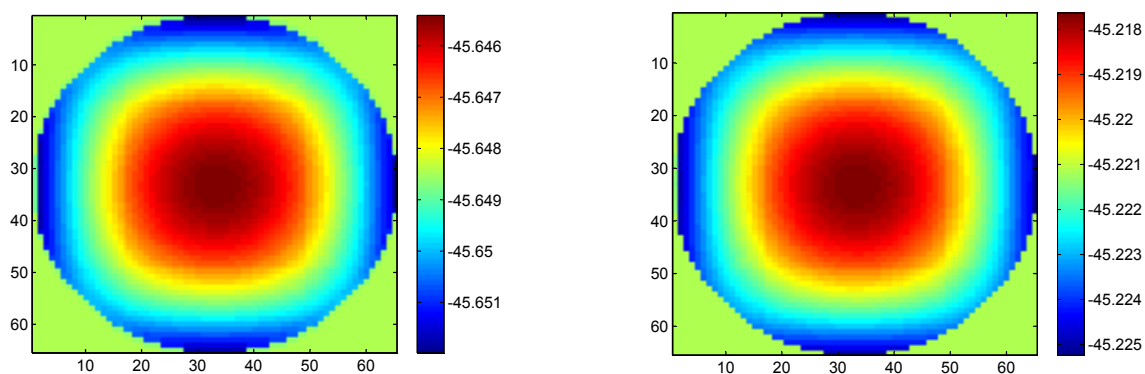


Figura 4.29: Reconstrução do parâmetro V_{xT} pelo método FBS (esquerda) e MB2 (direita).

A seguir apresentamos as figuras que representam a reconstrução dos demais parâmetros biológicos para o Exemplo 4.3 utilizando o Segundo Método de Broyden (MB2). A reconstrução do parâmetro k_3 foi omitida uma vez apresentou o valor constante de $k_3 \approx 0.0109$.

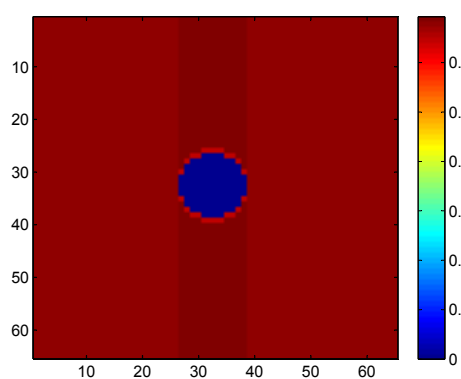


Figura 4.30: Reconstrução de k_2 .

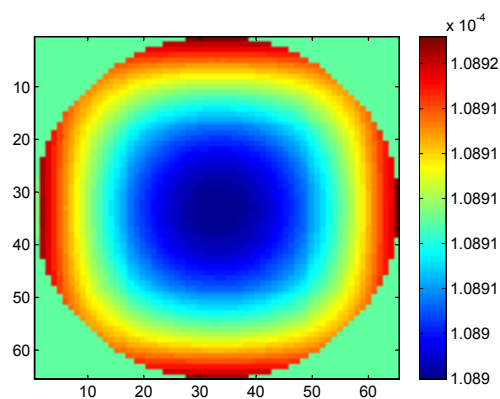


Figura 4.31: Reconstrução de V_{xA} .

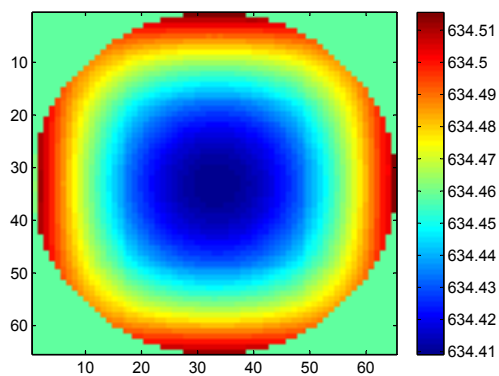


Figura 4.32: Reconstrução de V_{yA} .

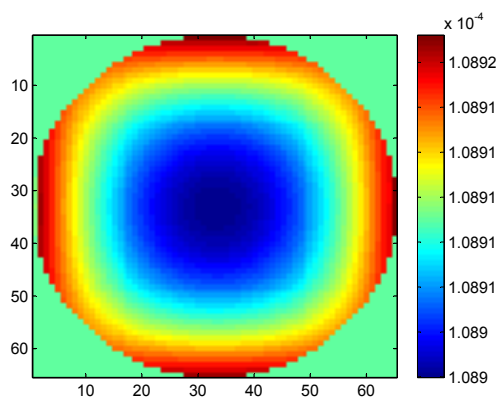


Figura 4.33: Reconstrução de V_{yT} .

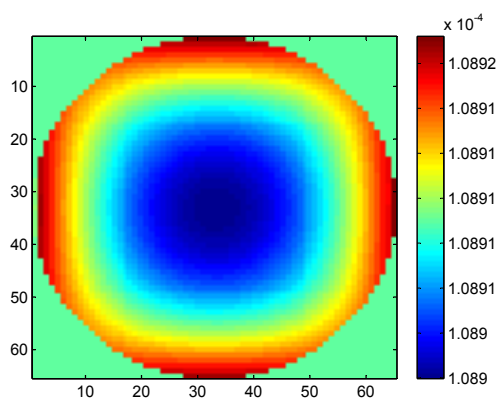


Figura 4.34: Reconstrução de V_{xV} .

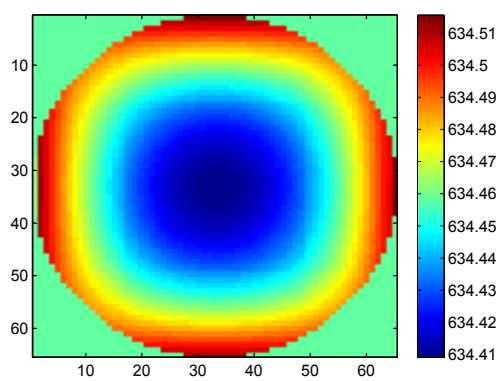


Figura 4.35: Reconstrução de V_{yV} .

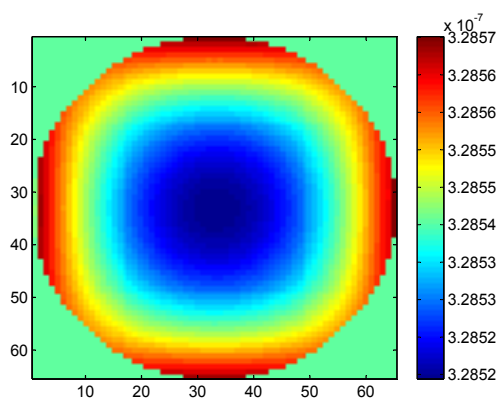


Figura 4.36: Reconstrução de D_{xA} .

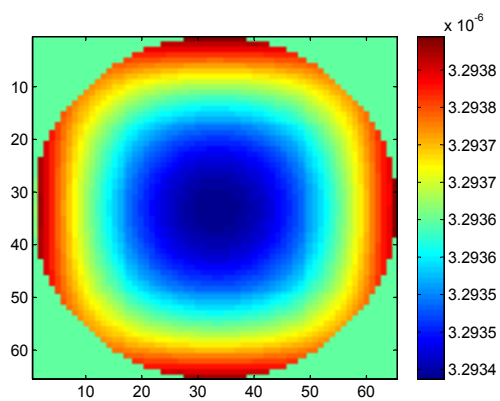


Figura 4.37: Reconstrução de D_{xT} .

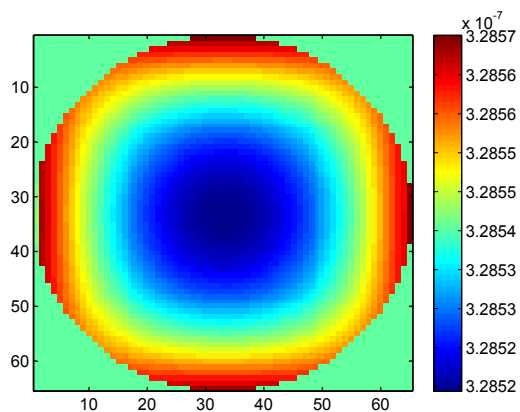


Figura 4.38: Reconstrução de $D_{x\nu}$.

No decorrer das próximas imagens, a concentração do material radioativo no tecido $C_{\mathcal{T}}$, na veia $C_{\mathcal{V}}$ e a imagem u , para diferentes estágios de tempo.

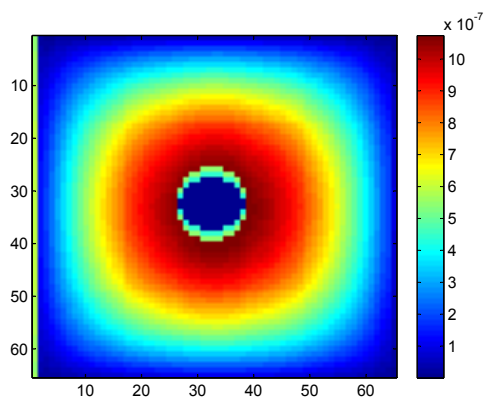


Figura 4.39: Reconstrução de $C_{\mathcal{T}} - t_3$.

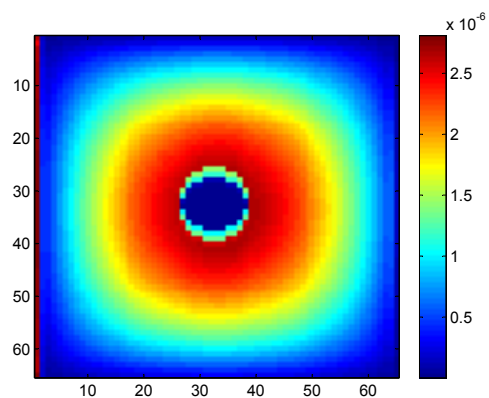


Figura 4.40: Reconstrução de $C_{\mathcal{T}} - t_6$.

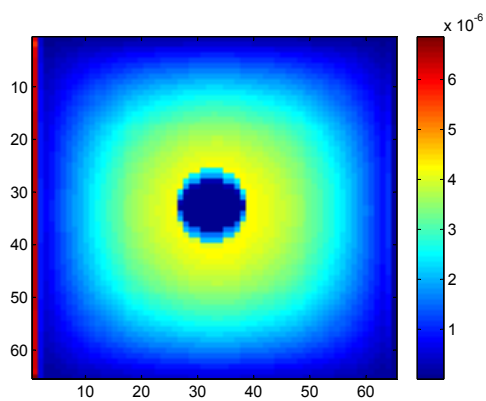


Figura 4.41: Reconstrução de $C_{\mathcal{T}} - t_9$.

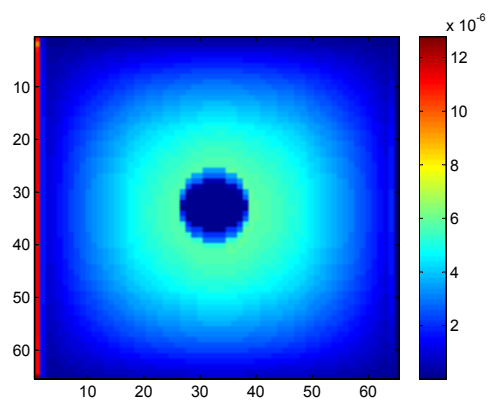


Figura 4.42: Reconstrução de $C_{\mathcal{T}} - t_{12}$.

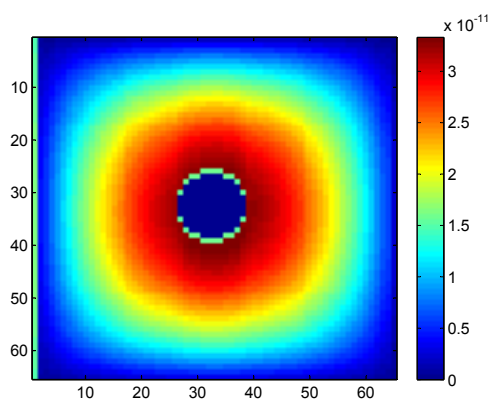


Figura 4.43: Reconstrução de $C_\gamma - t_3$.

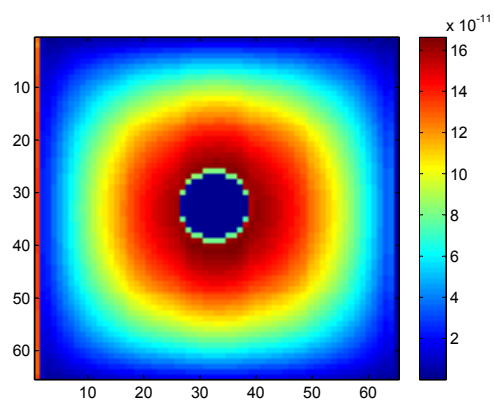


Figura 4.44: Reconstrução de $C_\gamma - t_6$.

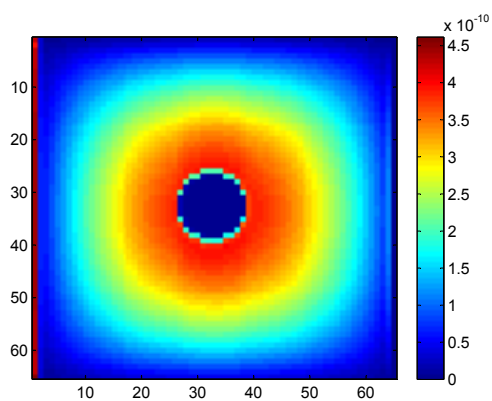


Figura 4.45: Reconstrução de $C_\gamma - t_9$.

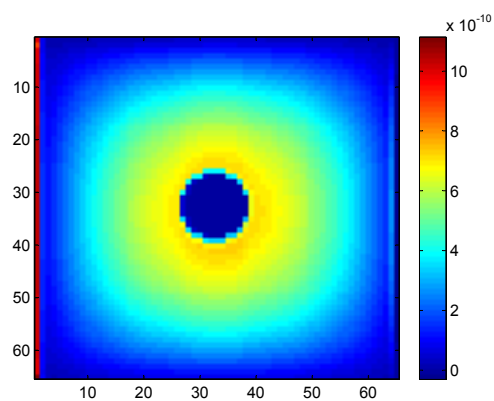


Figura 4.46: Reconstrução de $C_\gamma - t_{12}$.

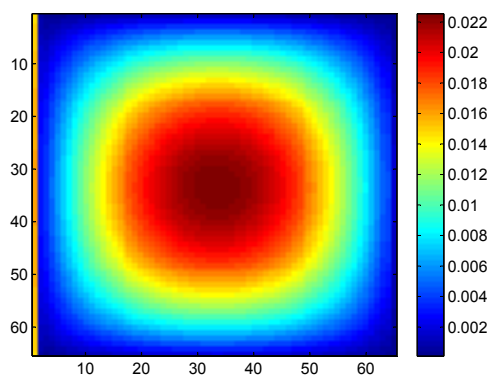


Figura 4.47: Reconstrução de $u - t_3$.

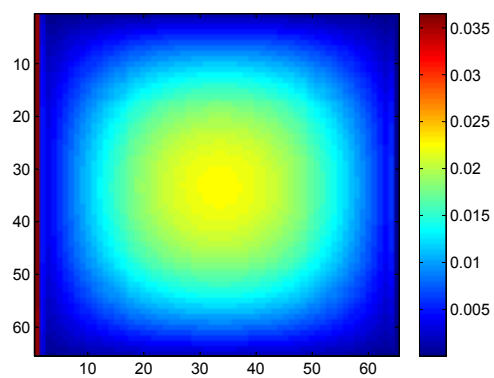
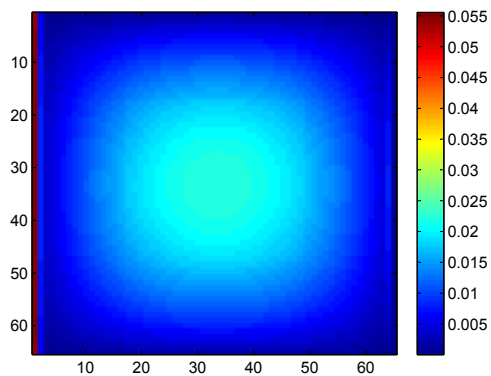
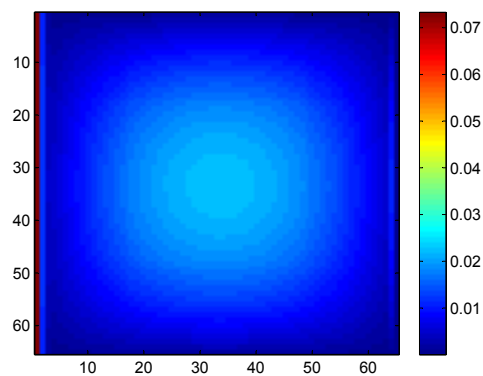


Figura 4.48: Reconstrução de $u - t_6$.

Figura 4.49: Reconstrução de $u - t_9$.Figura 4.50: Reconstrução de $u - t_{12}$.

Nas imagens acima percebemos que os valores nulos de k_1 e k_2 nos dados de entrada expostos na Figura 4.25, se refletem nos gráficos que representam a concentração do marcador radioativo no tecido e veia, através das lacunas que aparecem na mesma posição.

4.4 Conclusões

De maneira geral, os resultados aqui expostos evidenciam a eficiência da estratégia quase-Newton para problemas de ponto fixo com taxa de convergência linear, já verificada no capítulo anterior. Destacamos que a estratégia de redução no tamanho do passo mostrou-se muito eficiente, melhorando, do ponto de vista prático, a qualidade da solução.

Capítulo 5

Introdução aos métodos homotópicos

5.1 Introdução

Considere o sistema não linear

$$F(x) = 0 \tag{5.1}$$

em que $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é continuamente diferenciável.

Sabemos que o método de Newton é preferênciada para resolver (5.1), uma vez que desfruta de convergência quadrática. Porém, essa convergência ocorre apenas localmente, isto é, depende de uma boa aproximação inicial. Quando um ponto inicial adequado não é conhecido, a eficiência do método de Newton pode estar comprometida. Nesse contexto, surgem várias maneiras de globalizar o método, a busca linear [54] vista no Capítulo 2 é uma prática bem conhecida. No entanto, apenas garantimos a convergência para um ponto estacionário da função de mérito associada, a saber

$$f(x) = \frac{1}{2} \|F(x)\|^2.$$

Ou seja, o ponto encontrado pelo algoritmo pode ser um minimizador local, ou ainda, um maximizador.

Uma das maneiras de corrigir o problema de convergência a minimizadores locais (ou maximizadores) consiste em deformar o problema original através de homotopias. Os primeiros a usarem essa técnica de resolução de sistemas não lineares foram Lahaye em 1934 e Davidenko em 1953 [1]. Ao longo dos anos, os algoritmos homotópicos foram utilizados em várias áreas, sendo uma alternativa aos métodos locais [15, 53, 72, 73]. Na sequência apresentamos o método estudado.

5.2 O Método Homotópico

Na definição a seguir expomos o conceito de homotopia utilizado.

Definição 5.1. Uma homotopia é uma função $H : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ suficientemente suave, satisfazendo as condições:

- (i) $H(x^0, 0) = 0$;
- (ii) $H(x, 1) = F(x)$.

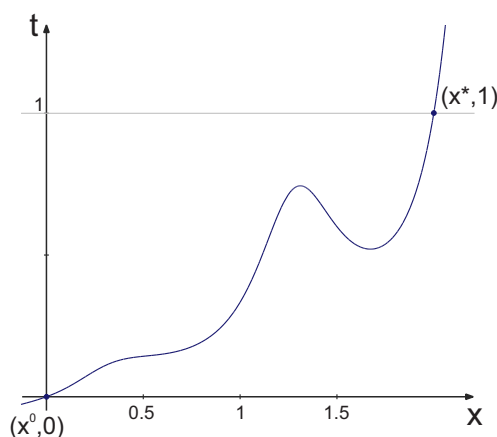


Figura 5.1: Exemplo de curva γ definida a partir da homotopia regularizante.

Um método homotópico consiste em percorrer a curva γ definida a partir do conjunto

$$\gamma \equiv \{(x, t) \in \mathbb{R}^n \times \mathbb{R} : H(x, t) = 0, 0 \leq t \leq 1\}. \quad (5.2)$$

Assim, partindo do ponto $(x^0, 0)$ espera-se encontrar a solução x^* de (5.1) ao atingirmos $t = 1$, ver Figura 5.1.

Entre as homotopias mais utilizadas, destacamos a homotopia regularizante

$$H(x, t) = tF(x) + (1 - t)(x - x^0) \quad (5.3)$$

em que $x^0 \in \mathbb{R}^n$, a homotopia global

$$H(x, t) = F(x) - (1 - t)F(x^0)$$

e a homotopia convexa

$$H(x, t) = tF(x) + (1 - t)G(x)$$

em que $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é uma função suave cujos zeros são conhecidos ou facilmente obtidos. Note que a homotopia regularizante é um caso particular da homotopia convexa com $G(x) = x - x^0$. Neste trabalho vamos considerar apenas a homotopia regularizante.

A princípio não há garantias de que seguir a curva γ partindo de um ponto $(x^0, 0)$ irá nos levar a uma solução $(x^*, 1)$ do problema (5.1). Para tanto é preciso assumir algumas propriedades adicionais, como colocado no Teorema de Chow-Mallet-Paret-Yorke [12] a seguir.

Teorema 5.2. Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuamente diferenciável. Suponha que exista uma homotopia $H : \mathbb{R}^m \times \mathbb{R}^n \times [0, 1) \rightarrow \mathbb{R}^n$ continuamente diferenciável tal que:

- (i) A matriz Jacobiana $H'(a, x, t) \in \mathbb{R}^{n \times (m+n+1)}$ tem posto n no conjunto

$$H^{-1}(0) = \{(a, x, t) : a \in \mathbb{R}^m, t \in [0, 1), H(a, x, t) = 0\}$$

para qualquer $a \in \mathbb{R}^m$ fixo;

- (ii) $H(a, x, 0) = 0$ tem única solução e $H'(a, x, 0)$ é não singular;

(iii) $H(a, x, 1) = F(x)$;

(iv) $H^{-1}(0)$ é limitado.

Então, para quase todo $a \in \mathbb{R}^m$ existe uma curva γ , definida em (5.2), ao longo da qual o Jacobiano de $H(a, x, t)$ com respeito a (x, t) tem posto n , partindo de $(a, 0)$ e atingindo uma solução x^* de $F(x) = 0$ em $t = 1$. Além disso, a curva γ não se auto intercepta e é disjunta de todos os outros zeros de $H(a, x, t)$.

Demonstração. Ver [12]. □

Neste trabalho, vamos considerar $m = n$ e $a = x^0$. Observando que a matriz Jacobiana da homotopia regularizante (5.3) é

$$H'(a, x, t) = \begin{bmatrix} (t-1)I & tF'(x) + (1-t)I & F(x) - (x-a) \end{bmatrix},$$

percebemos que a condição (i) do Teorema 5.2 é satisfeita. Além disso, as condições (ii) e (iii) do mesmo teorema são facilmente verificadas. No entanto, a condição (iv) precisa ser analisada para cada problema considerado. O próximo teorema, de [73], expressa uma condição suficiente sobre F para que a homotopia regularizante satisfaça a condição (iv) do Teorema 5.2.

Teorema 5.3. Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ continuamente diferenciável tal que para algum $r > 0$, $x^\top F(x) \geq 0$ sempre que $\|x\| = r$. Então, F tem um zero no conjunto $\{x \in \mathbb{R}^n : \|x\| \leq r\}$, e, para quase todo $a \in \mathbb{R}^m$, $\|a\| < r$, existe uma curva γ associada à homotopia regularizante (5.3) ao longo da qual o Jacobiano de $H(a, x, t)$ com respeito a (x, t) tem posto n , partindo de $(a, 0)$ e atingindo uma solução x^* de $F(x) = 0$ em $t = 1$. Além disso, se F' é não singular então γ tem comprimento de arco finito.

Algoritmos baseados em homotopias que satisfazem as propriedades (i)-(iv) acima são ditos globalmente convergentes com probabilidade um. Dentre os métodos homotópicos utilizados para resolver (5.1), escolhemos trabalhar com algoritmos que acompanham a curva γ . Na sequência, descrevemos esse processo de acordo com [74].

5.2.1 Seguindo a curva γ com Euler

Seguir a curva γ consiste em resolver, a cada iteração, um problema de valor inicial. Para tanto, assumimos que $H(x, t)$ tem posto completo ao longo de γ e que a componente de $H^{-1}(0)$ que contém o ponto $(x^0, 0)$ é limitada. Podemos então parametrizar γ pelo comprimento de arco s , de maneira que

$$H(x(s), t(s)) = 0. \tag{5.4}$$

Derivando (5.4) com relação a s , obtemos

$$H'(x(s), t(s)) \begin{bmatrix} \frac{dx}{ds} \\ \frac{dt}{ds} \end{bmatrix} = 0 \tag{5.5}$$

$$x(0) = x^0, \quad t(0) = 0$$

e

$$\left\| \begin{bmatrix} \frac{dx}{ds} \\ \frac{dt}{ds} \end{bmatrix} \right\| = 1. \tag{5.6}$$

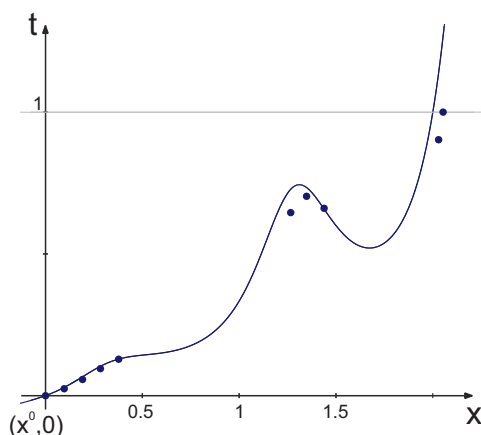


Figura 5.2: Alguns iterados do método homotópico com Euler.

Para resolver as equações (5.5) e (5.6) podemos utilizar qualquer método de solução de problemas de valor inicial (PVI). Neste trabalho, optamos por usar o método de Euler.

Defina $y(s) = (x(s), t(s))^T$. Para entender melhor como o método de Euler será aplicado na solução do PVI (5.5)-(5.6), vamos reescrever este problema na forma $y' = g(y(s))$. Desse modo, a atualização de Euler está definida por

$$y^{k+1} = y^k + hg(y^k). \quad (5.7)$$

Note que,

$$y' = \begin{bmatrix} \frac{dx}{ds} \\ \frac{dt}{ds} \end{bmatrix}.$$

Então, por (5.5) $y'(s) \in N(H'(x(s), t(s)))$. Assim, basta encontrar y' e substituir o seu valor na equação (5.7). Uma maneira de encontrar este vetor será apresentada na próxima seção.

Quando o comprimento de arco s torna-se muito grande, os pontos calculados podem estar muito distantes dos pontos verdadeiros. Sempre que isso ocorre, atualizamos o ponto atual (x, t) por

$$\bar{x}_0 = \frac{tF(x) + (1-t)x}{1-t}. \quad (5.8)$$

Dessa forma o novo ponto (\bar{x}_0, t) pertence à curva γ definida a partir da homotopia

$$H(x, t) = tF(x) + (1-t)(x - \bar{x}_0).$$

A Figura 5.2 exemplifica o processo de percorrer a curva γ usando o método de Euler.

Na sequência comentamos como encontrar um vetor tangente à curva γ , isto é, um vetor não nulo no núcleo de $H'(x, t)$.

5.2.2 Calculando y'

Para encontrar um vetor não nulo no núcleo de $H'(x, t)$ podemos usar a fatoração LU .

Assim, seja $A = H'(x, t)$ e considere a fatoração LU de A usando pivotamento parcial com troca de linhas, $A = PLU$. Dessa forma, $P \in \mathbb{R}^{n \times n}$ é matriz de permutação, $L \in \mathbb{R}^{n \times n}$ é uma

matriz triangular inferior com 1's na diagonal e $U \in \mathbb{R}^{n \times (n+1)}$ é uma matriz de posto n com $u_{ij} = 0$ se $i > j$ (matriz trapezoidal superior).

Note que, $z \in N(A)$ é equivalente a $z \in N(U)$, pois a matriz PL da decomposição LU é não singular. Então, para encontrar $y' = \left[\frac{dx}{ds} \right]$ tal que $\|y'\| = 1$, vamos resolver o seguinte sistema triangular superior

$$\begin{bmatrix} U \\ e_{n+1}^\top \end{bmatrix} z = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.9)$$

em que $e_{n+1} = (0, \dots, 1)^\top \in \mathbb{R}^{n+1}$.

Definindo

$$y' = \pm \frac{z}{\|z\|} \quad (5.10)$$

temos que $y' \in N(A)$ e $\|y'\| = 1$, conforme desejado.

Agora, se a fatoração LU de A não pôde ser concluída, então no estágio j todos os possíveis pivôs da coluna j são nulos. Segue que, existe uma matriz não singular C tal que as entradas $(j, j), \dots, (n, j)$ da matriz CA são nulas. Neste caso, o sistema $Az = 0$ é equivalente a $CAz = 0$ e uma solução não nula pode ser encontrada fazendo $z_i = 0$, se $i > j$, $z_j = 1$ e calculando z_1, \dots, z_{j-1} por substituição regressiva (ou retroativa). Por fim, definimos y' como em (5.10).

Note que, os vetores z e $-z$ são tangentes à curva γ . Para evitar que a orientação da curva seja alterada durante o processo, vamos escolher o vetor tangente que mantenha um ângulo agudo com o vetor calculado na iteração anterior. Ou seja, na iteração k definimos

$$y'_k = \frac{z^k}{\|z^k\|} \quad \text{se } (z^k)^\top z^{k-1} \geq 0,$$

e

$$y'_k = -\frac{z^k}{\|z^k\|} \quad \text{se } (z^k)^\top z^{k-1} < 0.$$

Para $k = 0$ definimos $y'_0 = \frac{z^0}{\|z^0\|}$ se $z_{n+1}^0 \geq 0$, e $y'_0 = -\frac{z^0}{\|z^0\|}$ caso contrário.

Podemos agora estabelecer o Método Homotópico, descrito no algoritmo abaixo.

Algoritmo 5.1. Método homotópico - MHO

Inicialização: Escolha $x^0 \in \mathbb{R}^n$, $\Delta s > 0$, $tol_s > 0$. Faça $y^0 = (x^0, 0)$ e $\bar{x}_0 = x^0$. Defina $H(x, t) = tF(x) + (1 - t)(x - \bar{x}_0)$. Faça $k = 0$.

Passo 1. Critério de parada: Se $y_{n+1}^k = t_k > 1$ corrija o iterado para obter uma solução, pare.

Passo 2. Cálculo da direção: Calcule o vetor tangente à curva γ de acordo com (5.9) e (5.10). Escolha o sinal apropriado.

Passo 3. Atualização de Euler: Defina $y^{k+1} = y^k + \Delta s y'$ e atualize $s_k = s_{k-1} + \Delta s$.

Passo 4. Correção da curva: Se $s_k \leq tol_s$, faça $k = k + 1$ e volte ao Passo 1. Caso contrário, atualize o ponto \bar{x}_0 de acordo com (5.8), faça $k = k + 1$ e volte ao Passo 2.

Na sequência descrevemos um algoritmo que combina os métodos de Newton e Homotopia.

5.3 Algoritmo híbrido de Newton com Homotopia (MNH)

Os métodos homotópicos destacam-se devido à garantia de convergência global para uma solução do sistema não linear (5.1). No entanto, tal convergência pode ocorrer de maneira muito lenta. Apesar disso, quando combinada com métodos locais rápidos, o uso da homotopia apresenta bons resultados [15, 38, 63].

Nosso objetivo é utilizar a estratégia homotópica apresentada na seção anterior para corrigir o método de Newton quando este último fracassa. Diferentemente da linha seguida nos algoritmos que usam a homotopia apenas para gerar pontos iniciais, ou que aplicam o método de Newton para resolver o sistema homotópico (5.3) (métodos de continuação) [1], vamos seguir a estratégia de Billups [8] e utilizar os métodos de Newton e Homotopia combinados. Para tanto, recorreremos à homotopia regularizante quando o método de Newton apresentar as seguintes dificuldades:

- (i) Na iteração k , a direção de Newton d^k é tal que $\|d^k\| < \varepsilon_1$;
- (ii) O algoritmo está convergindo para um ponto estacionário da função de mérito que não é solução do sistema não linear, isto é, quando

$$\|F(x^k)\| \geq \varepsilon_2 \quad \text{e} \quad |f(x^{k-1}) - f(x^k)| \leq \varepsilon_3;$$

- (iii) A condição de Armijo não é satisfeita na iteração k para um tamanho de passo $t_k \geq \varepsilon_4$.

Quando uma das condições acima ocorre, atualizamos o ponto \bar{x}_0 e recorreremos a uma iteração do método homotópico descrito no Algoritmo 5.1. Em seguida, continuamos com o método de Newton até que este convirja, ou então, recorreremos novamente à homotopia se uma das condições (i)-(iii) voltarem a ocorrer. O algoritmo híbrido está descrito a seguir.

Algoritmo 5.2. Algoritmo híbrido de Newton e Homotopia

Inicialização: Sejam $c_1 > 0$, $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4 > 0$, $tol > 0$, $\Delta_s > 0$, $\gamma > 0$ e $x^0 \in \mathbb{R}^n$. Faça $y^0 = (x^0, 0)$ e $\bar{x}_0 = x^0$. Defina $H(x, t) = tF(x) + (1 - t)(x - \bar{x}_0)$. Faça $k = 0$.

Passo 1. *Critério de parada:* Se $f(x^k) < tol$, pare.

Passo 2. *Iteração de Newton:* Calcule um iterado de Newton de acordo com os Passos 2, 3 e 4 do Algoritmo 2.1.

Passo 3. *Avaliação do iterado:* Caso alguma das dificuldades (i)-(iii) tenha ocorrido, vá para o Passo 4. Caso contrário, faça $k = k + 1$ e retorne ao Passo 1.

Passo 4. *Mudando para homotopia:* Se o iterado x^k foi calculado via método de Newton, atualize \bar{x}_0 de acordo com (5.8).

Passo 5. *Iteração de homotopia:* Calcule um iterado de homotopia de acordo com os Passos 2 e 3 do Algoritmo 5.1. Faça $x^{k+1} = (y_1^{k+1}, \dots, y_n^{k+1})^\top$, $k = k + 1$ e retorne ao Passo 1.

Na próxima seção expomos os resultados obtidos.

5.4 Resultados Numéricos e Problemas

Assim como os algoritmos apresentados nos capítulos anteriores, o algoritmo híbrido foi escrito em Fortran 77 com dupla precisão e os experimentos foram realizados usando gfortran-4.6 em uma máquina Intel CORE I3-2310M @ 2.10 GHz com 100 Gb de HD e 4Gb de Ram.

Para as constantes do Algoritmo 5.1 foram utilizados os valores: $\Delta s = 0.1$ e $tol_s = 0.9$. Após satisfazer o critério de parada, isto é, quando $y_{n+1}^k = t_k > 1$ realizamos uma iteração de Newton para corrigir o iterado encontrado e retornar à curva γ . Já para as constantes do Algoritmo 5.2, utilizamos os seguintes valores: $c_1 = 100$, $\varepsilon_1 = \varepsilon_2 = 10^{-6}$, $\varepsilon_3 = 10^{-4}$, $\varepsilon_4 = 10^{-3}$, $\Delta s = 0.1$, $\gamma = 10^{-4}$.

Consideramos uma tolerância de 10^{-6} significando que o seguinte critério de parada foi utilizado:

$$\text{Parar com } x^k \text{ se } \frac{1}{2} \|F(x^k)\|^2 < 10^{-6}.$$

Na sequência, definimos os problemas testados e expomos os resultados obtidos.

Exemplo 5.1. Considere:

$$F(x_1, x_2) = \begin{pmatrix} (x_1 - 1)^2 + (x_2 - 1)^2 - 1 \\ x_1 x_2 \end{pmatrix} = 0.$$

Exemplo 5.2. Considere:

$$F(x_1, x_2) = \begin{pmatrix} x_1 + x_2 - 1 \\ x_1 x_2 \end{pmatrix} = 0.$$

Os Exemplos 5.1 e 5.2 são problemas já considerados no Capítulo 2, a saber, *scale4* e *kth1*, respectivamente. Ambos apresentam simetria entre as variáveis x_1 e x_2 . Portanto, começando com um ponto x^0 simétrico, o método de Newton não consegue convergir para uma das soluções $x^* = (1, 0)^\top$ ou $x^* = (0, 1)^\top$. Neste experimento consideramos $x^0 = (1, 1)^\top$.

Exemplo 5.3. Considere:

$$F(x_1, x_2) = \begin{pmatrix} x_1^2 + x_2^2 - 2 \\ \exp(x_1 - 1) + x_2^2 - 2 \end{pmatrix} = 0.$$

Este problema foi retirado de [38] e possui quatro soluções, que correspondem aos minimizadores globais da função de mérito associada, ver Figura 5.3. Além disso, o ponto $\bar{x} \approx (-0.13, 0)^\top$ é maximizador local da função de mérito. Logo, quando começamos com o iterado $(0, 0)^\top$, o método de Newton não consegue convergir, uma vez que a segunda variável fica estagnada em $x_2 = 0$.

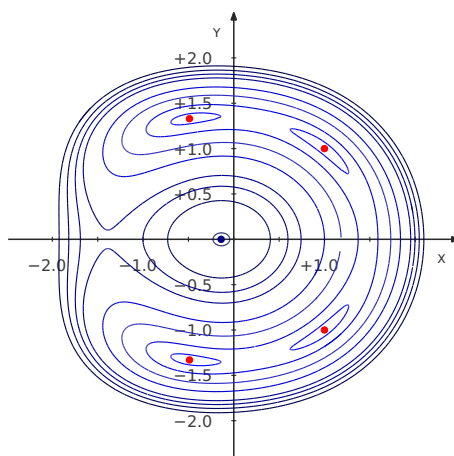


Figura 5.3: Curvas de nível da função de mérito associada ao problema do Exemplo 5.3. Em vermelho, as soluções do problema e, em azul o maximizador local.

Exemplo 5.4. Considere:

$$F(x_1, x_2, x_3) = \begin{pmatrix} 3x_1 - \cos(x_2x_3) - 0.5 \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 \\ \exp(-x_1x_2) + 20x_3 + \frac{(10\pi-3)}{3} \end{pmatrix} = 0$$

com ponto inicial $x^0 = (0, 0, 0)^\top$.

A Tabela 5.1 expõe os resultados para os exemplos definidos anteriormente. Nesta tabela, utilizamos a seguinte notação:

NNIT: número de iterações de Newton;

NHIT: número de iterações de Homotopia;

s_k : comprimento do arco na solução homotópica;

RES: resíduo encontrado, isto é, $\frac{1}{2}\|F(\bar{x})\|^2$ com \bar{x} o ponto encontrado pelo algoritmo;

*: resíduo foi maior do que a precisão exigida.

O tempo computacional não é apresentado, pois o mesmo foi nulo em todos os casos.

Nas Figuras 5.4 e 5.5 apresentamos os gráficos de convergência para cada um dos problemas testados.

Os resultados expostos na Tabela 5.1 são promissores e confirmam que utilizar o método homotópico em conjunto com o método de Newton, quando este último apresenta insucesso, é mais vantajoso do que utilizar apenas a abordagem homotópica. A mesma conclusão é enfatizada pelas Figuras 5.4 e 5.5.

Tabela 5.1: Comparativo entre o método homotópico e o algoritmo híbrido de Newton com homotopia para os diferentes exemplos testados.

Problema	Homotopia				Newton + Homotopia		
	NNIT	NHIT	RES	s_k	NNIT	NHIT	RES
Ex. 5.1	1	49	3.93e-07	0.3399	11	5	4.31e-07
Ex. 5.2	1	35	3.99e-11	0.2100	6	1	1.29e-09
Ex. 5.3	1	24	6.83e-12	0.1000	9	1	3.87e-09
Ex. 5.4	1	23	1.71e-04*	0.0899	11	4	3.01e-07

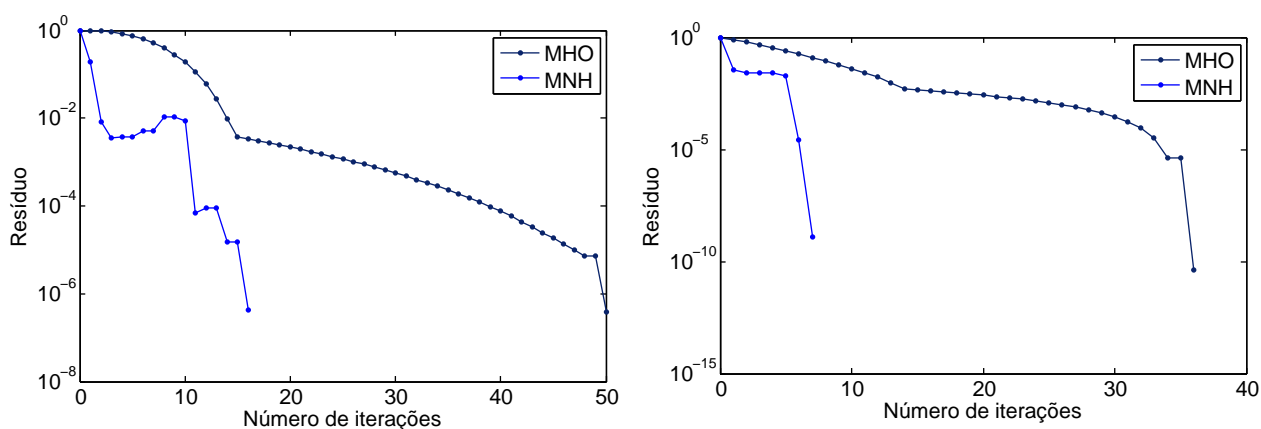


Figura 5.4: Gráfico de convergência dos métodos testados para o Exemplo 5.1 (esquerda) e Exemplo 5.2 (direita).

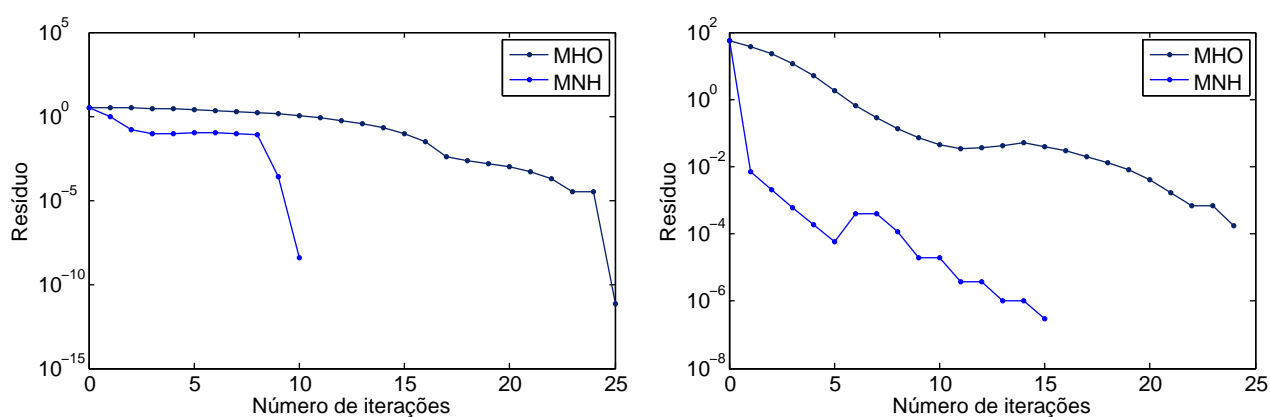


Figura 5.5: Gráfico de convergência dos métodos testados para o Exemplo 5.3 (esquerda) e Exemplo 5.4 (direita).

5.5 Conclusões

Neste capítulo realizamos um breve estudo sobre os métodos homotópicos, com o intuito de complementar o método de Newton visto no Capítulo 2. O algoritmo resultante é um método

híbrido que combina iterações de Newton e Homotopia. Os resultados obtidos confirmam que esta estratégia é efetiva e exhibe resultados promissores para os problemas onde o método de Newton apresenta dificuldades.

Capítulo 6

Considerações Finais

Neste trabalho estudamos diferentes abordagens para resolver sistemas não lineares, entre elas o método de Newton, métodos quase-Newton e Homotopia.

No Capítulo 2 apresentamos uma implementação esparsa para o método de Newton. Em seguida, aplicamos o algoritmo em problemas com restrições de complementaridade, dando origem a um método globalmente convergente que combina iterações de pontos interiores com gradientes projetados (PGUN). Ressaltamos que esse tipo de restrições aparece em importantes problemas práticos de programação não linear. Com o objetivo de verificar os resultados teóricos e analisar o desempenho do algoritmo apresentado, realizamos experimentos computacionais em uma coleção conhecida de problemas. Os resultados alcançados indicam que o algoritmo PGUN apresentou um bom desempenho para calcular soluções viáveis para MPCC e parece ser mais eficiente do que um método do tipo Levenberg-Marquardt Projetado (PLM) concebido anteriormente para o mesmo objetivo. A escolha do ponto inicial para os algoritmos PGUN e PLM parece ter um impacto importante sobre a eficiência desses algoritmos e trabalhos futuros podem seguir nessa direção.

Na sequência, no Capítulo 3, estudamos os métodos quase-Newton como aceleradores de processos iterativos lineares. Os resultados mostrados no capítulo indicam que, em geral, a estratégia quase-Newton mostrou-se eficiente na aceleração algorítmica. Com efeito, exceto para exemplos pontuais, os métodos quase-Newton obtiveram um desempenho superior ao método original, reduzindo o número de iterações e, principalmente, o tempo computacional. Em seguida, no Capítulo 4, uma aplicação prática em problemas oriundos de exames de Tomografia por Emissão de Pósitrons (PET) foi considerada, reforçando a efetividade da técnica estudada.

Por fim, no Capítulo 5, descrevemos uma estratégia de globalização baseada em homotopias. O algoritmo resultante é um método híbrido que combina iterações locais de Newton com iterações globais de homotopia. Realizamos experimentos iniciais para validar o método proposto, os resultados obtidos foram promissores e confirmaram a superioridade do algoritmo híbrido. Uma vez que o objetivo era realizar uma breve análise, estamos cientes que é necessário realizar testes em uma gama mais ampla de problemas. Efetuar tais experimentos está em nossos planos futuros.

Pelo acima exposto, percebemos que o trabalho se mostrou efetivo para os problemas considerados. Ademais, os resultados obtidos por meio dos diferentes métodos nos encorajam a aplicá-los em problemas futuros.

Bibliografia

- [1] E. L. Allgower e K. Georg. *Numerical continuation methods: an introduction*. Vol. 13. Springer Series in Computational Mathematics. Springer-Verlag, 1990.
- [2] R. Andreani, C. Dunder e J. M. Martínez. “Nonlinear-programming reformulation of the order-value optimization problem”. Em: *Mathematical Methods of Operations Research* 61.3 (2005), pp. 365–384.
- [3] R. Andreani, J. J. Júdice, J. M. Martínez e T. Martini. “Feasibility problems with complementarity constraints”. Em: *European Journal of Operational Research* 249.1 (2016), pp. 41–54.
- [4] R. Andreani, J. J. Júdice, J. M. Martínez e J. Patrício. “A projected-gradient interior-point algorithm for complementarity problems”. Em: *Numerical Algorithms* 57.4 (2011), pp. 457–485.
- [5] R. Andreani, J. J. Júdice, J. M. Martínez e J. Patrício. “On the natural merit function for solving complementarity problems”. Em: *Mathematical Programming* 130.1 (2011), pp. 211–223.
- [6] A. Ben-Israel. “A Newton-Raphson method for the solution of systems of equations”. Em: *Journal of Mathematical Analysis and Applications* 15.2 (1966), pp. 243–252.
- [7] M. Bierlaire e F. Crittin. “Solving noisy, large-scale fixed-point problems and systems of nonlinear equations”. Em: *Transportation Science* 40.1 (2006), pp. 44–63.
- [8] S. C. Billups. “A Homotopy-Based Algorithm for Mixed Complementarity Problems”. Em: *SIAM Journal on Optimization* 12.3 (2002), pp. 583–605.
- [9] C. G. Broyden. “A Class of Methods for Solving Nonlinear Simultaneous Equations”. Em: *Mathematics of Computation* 19.92 (1965), pp. 577–593.
- [10] R. L. Burden e J. D. Faires. *Numerical Analysis*. 6. Brooks/Cole Publishing Company, Califórnia, 1997.
- [11] G. H-G. Chen e R. T. Rockafellar. “Convergence rates in forward-backward splitting”. Em: *SIAM Journal on Optimization* 7.2 (1997), pp. 421–444.
- [12] S. N. Chow, J. Mallet-Paret e J. A. Yorke. “Finding zeroes of maps: homotopy methods that are constructive with probability one”. Em: *Mathematics of Computation* 32.143 (1978), pp. 887–899.
- [13] J. W. Demmel. *Applied Numerical Linear Algebra*. Philadelphia: SIAM, 1997.
- [14] P. Deuffhard, R. Freund e A. Walter. “Fast secant methods for the iterative solution of large nonsymmetric linear systems”. Em: *IMPACT of Computing in Science and Engineering* 2.3 (1990), pp. 244–276.

-
- [15] D. M. Dunlavy, D. P. O’Leary, D. Klimov e D. Thirumalai. “HOPE: a homotopy optimization method for protein structure prediction”. Em: *Journal of Computational Biology* 12.10 (2005), pp. 1275–1288.
- [16] A. Ehrenmann e K. Neuhoff. “A comparison of electricity market designs in networks”. Em: *Operations Research* 57.2 (2009), pp. 274–286.
- [17] V. Eyert. “A comparative study on methods for convergence acceleration of iterative vector sequences”. Em: *Journal of Computational Physics* 124.2 (1996), pp. 271–285.
- [18] H. Fang e Y. Saad. “Two classes of multiseccant methods for nonlinear acceleration”. Em: *Numerical Linear Algebra with Applications* 16.3 (2009), pp. 197–221.
- [19] L. Fernandes, A. Friedlander, M. Guedes e J. J. Júdice. “Solution of a general linear complementarity problem using smooth optimization and its application to bilinear programming and LCP”. Em: *Applied Mathematics & Optimization* 43.1 (2001), pp. 1–19.
- [20] M. C. Ferris e J. S. Pang. “Engineering and economic applications of complementarity problems”. Em: *SIAM Review* 39.4 (1997), pp. 669–713.
- [21] M. Frank e P. Wolfe. “An algorithm for quadratic programming”. Em: *Naval Research Logistics Quarterly* 3.1-2 (1956), pp. 95–110.
- [22] R. García-Rodenas e D. Verastegui-Rayo. “A column generation algorithm for the estimation of origin–destination matrices in congested traffic networks”. Em: *European Journal of Operational Research* 184.3 (2008), pp. 860–878.
- [23] M. S. Gowda. “Reducing a monotone horizontal LCP to an LCP”. Em: *Applied Mathematics Letters* 8.1 (1995), pp. 97–100.
- [24] A. Griewank e A. Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*. Philadelphia: SIAM, 2008.
- [25] L. Guo, G-H. Lin, D. Zhang e D. Zhu. “An MPEC reformulation of an EPEC model for electricity markets”. Em: *Operations Research Letters* 43.3 (2015), pp. 262–267.
- [26] HSL. *A collection of Fortran codes for large scale scientific computation*. 2013. URL: <http://www.hsl.rl.ac.uk>.
- [27] X. Hu e D. Ralph. “Using EPECs to model bilevel games in restructured electricity markets with locational prices”. Em: *Operations Research* 55.5 (2007), pp. 809–827.
- [28] A.F. Izmailov. “On the analytical and numerical stability of critical Lagrange multipliers”. Em: *Zhurnal Vychislitel’noi Matematiki i Matematicheskoi Fiziki* 45.6 (2005), pp. 966–982.
- [29] A.F. Izmailov. “On the limiting properties of dual trajectories in the Lagrange multipliers method”. Em: *Computational Mathematics and Mathematical Physics* 51.1 (2011), pp. 1–20.
- [30] A.F. Izmailov. “Solution sensitivity for Karush-Kuhn-Tucker systems with non-unique Lagrange multipliers”. Em: *Optimization* 59.5 (2010), pp. 747–775.
- [31] A.F. Izmailov e M.V. Solodov. “Critical Lagrange multipliers: what we currently know about them, how they spoil our life, and what we can do about it”. Em: *TOP* 23.1 (2015), pp. 1–26.

-
- [32] A.F. Izmailov e E.I. Uskov. “Attraction of Newton method to critical Lagrange multipliers: fully quadratic case”. Em: *Mathematical Programming* (2012), pp. 1–41.
- [33] M. Jamshidian e R. I. Jennrich. “Acceleration of the EM Algorithm by using Quasi-Newton Methods”. Em: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59.3 (1997), pp. 569–587.
- [34] D. D. Johnson. “Modified Broyden’s method for accelerating convergence in self-consistent calculations”. Em: *Physical Review B* 38.18 (1988), pp. 12807–12813.
- [35] J. J. Júdice. “Optimization with linear complementarity constraints”. Em: *Pesquisa Operacional* 34.3 (2014), pp. 559–584.
- [36] C. Kanzow, N. Yamashita e M. Fukushima. “Levenberg-Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints”. Em: *Journal of Computational and Applied Mathematics* 173.2 (2005), pp. 321–343.
- [37] R. M. Kovacevic e Ch. G. Pflug. “Electricity swing option pricing by stochastic bilevel optimization: A survey and new approaches”. Em: *European Journal of Operational Research* 237.2 (2014), pp. 389–403.
- [38] C.-Y. Ku, W. Yeih e C.-S. Liu. “Solving non-linear algebraic equations by a scalar Newton-homotopy continuation method”. Em: *International Journal of Nonlinear Sciences and Numerical Simulation* 11.6 (2010), pp. 435–450.
- [39] K. Lange. “A quasi-Newton acceleration of the EM algorithm”. Em: *Statistica Sinica* 5.1 (1995), pp. 1–18.
- [40] S. Leyffer. “MacMPEC: AMPL collection of MPECs”. Em: *Argonne National Laboratory*. Available at <http://wiki.mcs.anl.gov/leyffer/index.php/MacMPEC> (2000).
- [41] D. H. Li e M. Fukushima. “A derivative-free line search and global convergence of Broyden-like method for nonlinear equations”. Em: *Optimization Methods and Software* 13.3 (2000), pp. 181–201.
- [42] G-H. Lin, D. Zhang e Y-C Liang. “Stochastic multiobjective problems with complementarity constraints and applications in healthcare”. Em: *European Journal of Operational Research* 226.3 (2013), pp. 461–470.
- [43] Z. Q. Luo, J. S. Pang e D. Ralph. *Mathematical programs with equilibrium constraints*. Cambridge: Cambridge University Press, 1996.
- [44] J. M. Martínez. “A Quasi-Newton method with modification of one column per iteration”. Em: *Computing* 33.3-4 (1984), pp. 353–362.
- [45] J. M. Martínez. “Practical quasi-Newton methods for solving nonlinear systems”. Em: *Journal of Computational and Applied Mathematics* 124 (2000), pp. 97–122.
- [46] J. M. Martínez e S. A. Santos. *Métodos Computacionais de Otimização*. Colóquio Brasileiro de Matemática - Apostilas. Rio de Janeiro: IMPA, SBM, 1995.
- [47] J. M. Martínez e M. C. Zambaldi. “An inverse Column-Updating method for solving Large-Scale Nonlinear Systems of Equations”. Em: *Optimization Methods and Software* 1 (1992), pp. 129–140.
- [48] J. J. Moré. “A Collection of Nonlinear Model Problems”. Em: *Computational Solution of Nonlinear Systems of Equations*. Ed. por E. L. Allgower e K. Georg. Vol. 26. Lectures in Applied Mathematics. American Mathematical Society, 1990.

-
- [49] J. J. Moré e M. Y. Cosnard. “Numerical Solution of Nonlinear Equations”. Em: *ACM Transactions on Mathematical Software* 5.1 (1979), pp. 64–85.
- [50] G. Muehllehner e J. S. Karp. “Positron emission tomography”. Em: *Physics in medicine and biology* 51.13 (2006), R117–R137.
- [51] K. G. Murty. *Linear complementarity*. Linear e Nonlinear Programming, Heldermann, Berlin, 1988.
- [52] Y. Nievergelt. “The condition of Steffensen’s acceleration in several variables”. Em: *Journal of Computational and Applied Mathematics* 58.3 (1995), pp. 291–305.
- [53] D. Niu, Z. Jin, X. Wu e Y. Inoue. “A globally convergent and highly efficient homotopy method for MOS transistor circuits”. Em: *7th International Conference on Computing and Convergence Technology*. 2012, pp. 1349–1352.
- [54] J. Nocedal e S. J. Wright. *Numerical Optimization*. Springer, New York, 2006.
- [55] J. M. Ollinger e J. A. Fessler. “Positron-emission tomography”. Em: *Signal Processing Magazine* 14.1 (1997), pp. 43–55.
- [56] J. M. Ortega e W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*, Philadelphia: SIAM, 2000.
- [57] J. Outrata, M. Kocvara e J. Zowe. *Nonsmooth approach to optimization problems with equilibrium constraints: theory, applications and numerical results*. Vol. 28. Springer Science & Business Media, 1998.
- [58] J. S. Pang. “Partially B-regular optimization and equilibrium problems”. Em: *Mathematics of Operations Research* 32.3 (2007), pp. 687–699.
- [59] D. Ralph. *Nonlinear programming advances in mathematical programming with complementarity constraints*. 2008. URL: <http://www.eng.cam.ac.uk/~dr241/Papers/MPCC-review.pdf>.
- [60] D. Ralph e O. Stein. “The C-index: A new stability concept for quadratic programs with complementarity constraints”. Em: *Mathematics of Operations Research* 36.3 (2011), pp. 504–526.
- [61] L. Reips. “Parameter Identification in Medical Imaging”. Tese de doutorado. Münster: Westfälische Wilhelms Universität Münster, 2013.
- [62] R. Saigal e M. J. Todd. “Efficient acceleration techniques for fixed point algorithms”. Em: *SIAM Journal on Numerical Analysis* 15.5 (1978), pp. 997–1007.
- [63] A. Speight. “Using Homotopy Methods to Solve Nonsmooth Equations”. Tese de doutorado. Denver: University of Colorado at Denver, 1999.
- [64] G. P. Srivastava. “Broyden’s method for self-consistent field convergence acceleration”. Em: *Journal of Physics A: Mathematical and General* 17.6 (1984), pp. L317–L321.
- [65] S. W. Thomas. *Sequential estimation techniques for quasi-Newton algorithms*. Rel. téc. Cornell University, 1975.
- [66] A. N. Tikhonov e V. Y. Arsenin. *Solutions of ill-Posed Problems*. New York: Wiley, 1977.
- [67] D. Vanderbilt e S. G. Louie. “Total energies of diamond (111) surface reconstructions by a linear combination of atomic orbitals method”. Em: *Physical Review B* 30.10 (1984), pp. 6118–6130.

-
- [68] R. Varadhan e C. Roland. “Simple and globally convergent methods for accelerating the convergence of any EM algorithm”. Em: *Scandinavian Journal of Statistics* 35.2 (2008), pp. 335–353.
- [69] S. Villa, S. Salzo, L. Baldassarre e A. Verri. “Accelerated and inexact forward-backward algorithms”. Em: *SIAM Journal on Optimization* 23.3 (2013), pp. 1607–1633.
- [70] H. F. Walker e P. Ni. “Anderson acceleration for fixed-point iterations”. Em: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735.
- [71] J. Walpen, M. E. Mancinelli e P. A. Lotito. “A heuristic for the OD matrix adjustment problem in a congested transport network”. Em: *European Journal of Operational Research* 242.3 (2015), pp. 807–819.
- [72] L. T. Watson. “Engineering applications of the Chow-Yorke algorithm”. Em: *Applied Mathematics and Computation* 9.2 (1981), pp. 111–133.
- [73] L. T. Watson. “Globally convergent homotopy algorithms for nonlinear systems of equations”. Em: *Nonlinear Dynamics* 1.2 (1990), pp. 143–191.
- [74] L. T. Watson, S. C. Billups e A. P. Morgan. “Algorithm 652: HOMPACK: A suite of codes for globally convergent homotopy algorithms”. Em: *ACM Transactions on Mathematical Software (TOMS)* 13.3 (1987), pp. 281–310.
- [75] D. Wu, Y. Yin e S. Lawphongpanich. “Pareto-improving congestion pricing on multimodal transportation networks”. Em: *European Journal of Operational Research* 210.3 (2011), pp. 660–669.
- [76] J. Yao, I. Adler e S. S. Oren. “Modeling and computing two-settlement oligopolistic equilibrium in a congested electricity network”. Em: *Operations Research* 56.1 (2008), pp. 34–47.
- [77] J. J. Ye. “Necessary optimality conditions for multiobjective bilevel programs”. Em: *Mathematics of Operations Research* 36.1 (2011), pp. 165–184.
- [78] T. J. Ypma. “Historical development of the Newton-Raphson method”. Em: *SIAM Review* 37.4 (1995), pp. 531–551.
- [79] H. Zhou, D. Alexander e K. Lange. “A quasi-Newton acceleration for high-dimensional optimization algorithms”. Em: *Statistics and Computing* 21.2 (2011), pp. 261–273.

