

UNIVERSIDADE ESTADUAL PAULISTA “JULIO DE MESQUITA FILHO”  
FACULDADE DE ENGENHARIA DE BAURU  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ELLEN CRISTINA FERREIRA

**UMA INVESTIGAÇÃO DA INFLUÊNCIA DO PARÂMETRO DE  
BARREIRA NA SOLUÇÃO DO PROBLEMA DE FLUXO DE  
POTÊNCIA ÓTIMO**

BAURU/SP

2013

ELLEN CRISTINA FERREIRA

**UMA INVESTIGAÇÃO DA INFLUÊNCIA DO PARÂMETRO DE  
BARREIRA NA SOLUÇÃO DO PROBLEMA DE FLUXO DE  
POTÊNCIA ÓTIMO**

Dissertação apresentada ao Programa de Pós  
Graduação em Engenharia Elétrica da  
Universidade Estadual Paulista “Júlio de  
Mesquita Filho” Campus de Bauru, como  
parte dos requisitos para obtenção do título de  
Mestre em Engenharia Elétrica.

ORIENTADORA: Prof<sup>a</sup>. Dr<sup>a</sup>. Edméa Cássia Baptista  
CO-ORIENTADORA: Prof<sup>a</sup>. Dr<sup>a</sup>. Edilaine Martins Soler

BAURU/SP

2013

Ferreira, Ellen Cristina.

Uma investigação da influência do parâmetro de barreira na solução do problema de fluxo de potência ótimo / Ellen Cristina Ferreira, 2013  
141 f.

Orientador: Edméa Cássia Baptista.

Co-orientador: Edilaine Martins Soler

Dissertação (Mestrado)-Universidade Estadual Paulista. Faculdade de Engenharia, Bauru, 2013

1. Métodos de Pontos Interiores. 2. Método de Região de Confiança. 3. *Knitro*. 4. Fluxo de Potência Ótimo I. Universidade Estadual Paulista. Faculdade de Engenharia. II. Título.

**ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE ELLEN CRISTINA FERREIRA, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, DO(A) FACULDADE DE ENGENHARIA DE BAURU.**

Aos 08 dias do mês de março do ano de 2013, às 09:30 horas, no(a) ANFITEATRO DA SEÇÃO TÉCNICA DE PÓS-GRADUAÇÃO DA FACULDADE DE ENGENHARIA DE BAURU, reuniu-se a Comissão Examinadora da Defesa Pública, composta pelos seguintes membros: Profa. Dra. EDMEA CASSIA BAPTISTA do(a) Departamento de Matemática / Faculdade de Ciências de Bauru - UNESP, Prof. Dr. EDUARDO NOBUHIRO ASADA do(a) Departamento de Engenharia Elétrica / Escola de Engenharia de São Carlos - USP, Prof. Dr. ANTONIO ROBERTO BALBO do(a) Departamento de Matemática / Faculdade de Ciências de Bauru - UNESP, sob a presidência do primeiro, a fim de proceder a arguição pública da DISSERTAÇÃO DE MESTRADO de ELLEN CRISTINA FERREIRA, intitulada "UMA INVESTIGAÇÃO DA INFLUÊNCIA DO PARÂMETRO DE BARREIRA NA SOLUÇÃO DO PROBLEMA DE FLUXO DE POTÊNCIA ÓTIMO". Após a exposição, a discente foi arguida oralmente pelos membros da Comissão Examinadora, tendo recebido o conceito final: APROVADA. Nada mais havendo, foi lavrada a presente ata, que, após lida e aprovada, foi assinada pelos membros da Comissão Examinadora.

  
Profa. Dra. EDMEA CASSIA BAPTISTA

  
Prof. Dr. EDUARDO NOBUHIRO ASADA

  
Prof. Dr. ANTONIO ROBERTO BALBO

“Dedico aos meus pais, pois sempre  
acreditaram em mim.”

## **Agradecimentos**

A Deus, por sempre estar ao meu lado guiando meus passos.

Aos meus pais, Claudionor e Ilza, por me incentivarem incansavelmente e às minhas irmãs, Ana e Elaine, que sempre acreditaram em mim.

À Professora Dra. Edméa Cássia Baptista, pela orientação, paciência, compreensão, disponibilidade, amizade e pela confiança depositada em mim durante o desenvolvimento deste trabalho.

Ao meu noivo Rafael Grecco Sanches, por sempre estar ao meu lado nos momentos difíceis, pela sua paciência, sua compreensão e seu incentivo na minha vida profissional.

À minha grande amiga Camila de Lima, pela participação em todos os momentos, por me apoiar e pelas risadas.

Aos professores do curso de mestrado em Engenharia Elétrica.

À Professora Dra. Edilaine M. Soler pela co-orientação.

Aos amigos do Laboratório de Otimização e Estudos Econômicos de Sistemas de Potência - LOESP.

Ao professor Dr. Eduardo N. Asada da EESC/USP, por ceder a licença do GAMS.

À Seção de Pós Graduação e ao Departamento de Matemática, direta e indiretamente, contribuíram com este trabalho.

À Reitoria da Universidade Estadual Paulista “Júlio de Mesquita Filho” pela credibilidade e apoio financeiro.

# SUMÁRIO

|  |             |
|--|-------------|
| <b>LISTA DE SÍMBOLOS .....</b>   | <b>III</b>  |
| <b>LISTA DE FIGURAS.....</b>   | <b>VI</b>   |
| <b>LISTA DE TABELAS.....</b>   | <b>VII</b>  |
| <b>RESUMO .....</b>  | <b>VIII</b> |
| <b>ABSTRACT.....</b>   | <b>IX</b>   |
| <b>INTRODUÇÃO.....</b>   | <b>1</b>    |
| <b>REVISÃO DE MÉTODOS DE OTIMIZAÇÃO .....</b>  | <b>6</b>    |
| 2.1 MÉTODOS DE BUSCA LINEAR .....  | 6           |
| 2.1.1 A escolha do tamanho do passo da busca linear .....  | 7           |
| 2.2 O MÉTODO PRIMAL-DUAL BARREIRA LOGARÍTMICA.....   | 11          |
| 2.3 MÉTODOS DE DIREÇÕES CONJUGADAS .....   | 15          |
| 2.3.1 O Método do Gradiente Conjugado.....   | 16          |
| 2.3.2 Método Geral dos Gradiente Conjugados.....   | 18          |
| 2.4 MÉTODOS REGIÃO DE CONFIANÇA .....  | 19          |
| 2.4.1 Ponto de Cauchy .....  | 25          |
| 2.5 KNITRO: PACOTE DE ALGORITMOS DE OTIMIZAÇÃO .....   | 31          |
| 2.5.1 Algoritmo Interior-Direct.....   | 33          |
| 2.5.2 Algoritmo Interior-CG .....  | 34          |
| 2.5.3 Algoritmo Active-Set.....  | 35          |
| 2.5.4 A opção crossover.....   | 36          |
| <b>MÉTODO DE PONTO INTERIOR COM BUSCA LINEAR E REGIÃO DE<br/>CONFIANÇA.....</b>  | <b>37</b>   |
| 3.1 DEFINIÇÃO DO PROBLEMA .....  | 37          |
| 3.2 O MÉTODO .....   | 37          |
| 3.3 A ESTRATÉGIA DE BUSCA LINEAR.....  | 38          |
| 3.4 A ESTRATÉGIA DE REGIÃO DE CONFIANÇA.....   | 43          |
| 3.4.1 O passo da região de confiança.....  | 46          |
| 3.5 MÉTODO GRADIENTE CONJUGADO .....   | 47          |
| 3.6 O MÉTODO DO GRADIENTE CONJUGADO PROJETADO .....  | 50          |
| 3.7 CRITÉRIOS DE PARADA .....  | 53          |
| 3.7.1 Programação não linear.....  | 54          |
| 3.7.2 O problema de barreira .....   | 55          |
| 3.8 O ALGORITMO DO MÉTODO DE PONTO INTERIOR IMPLEMENTADO NO KNITRO .....   | 58          |
| <b>ATUALIZAÇÕES DO PARÂMETRO DE BARREIRA .....</b>   | <b>61</b>   |
| 4.1 ESTRATÉGIA (I): O PARÂMETRO DE BARREIRA É DECRESCIDO MONOTONICAMENTE ...   | 62          |
| 4.2 ESTRATÉGIA (II): O PARÂMETRO DE BARREIRA É CALCULADO POR UMA REGRA<br>ADAPTATIVA COM BASE NA DIFERENÇA DE COMPLEMENTARIDADE .....                                  | 63          |
| 4.3 ESTRATÉGIA (III): É REALIZADA UMA INVESTIGAÇÃO SOBRE O PASSO (AFIM ESCALA)<br>PARA DETERMINAR DINAMICAMENTE O VALOR DO PARÂMETRO BARREIRA A CADA<br>ITERAÇÃO ..... | 64          |

|   |            |
|---|------------|
| 4.4 ESTRATÉGIA (IV): É UTILIZADA UMA REGRA MEHROTRA TIPO PREVISOR-CORRETOR PARA DETERMINAR O PARÂMETRO DE BARREIRA UTILIZANDO O PASSO CORRETOR .....          | 65         |
| 4.5 ESTRATÉGIA (V): É UTILIZADA UMA REGRA MEHROTRA TIPO PREVISOR-CORRETOR PARA DETERMINAR O PARÂMETRO DE BARREIRA EM QUE O PASSO CORRETOR É CONDICIONADO..... | 66         |
| 4.6 ESTRATÉGIA (VI): UMA FUNÇÃO QUALIDADE É MINIMIZADA A CADA ITERAÇÃO PARA DETERMINAR O PARÂMETRO DE BARREIRA .....  | 67         |
| <b>O PROBLEMA DE FLUXO DE POTÊNCIA ÓTIMO.....</b>   | <b>70</b>  |
| <b>RESULTADOS NUMÉRICOS .....</b>   | <b>82</b>  |
| 6.1 TESTES NUMÉRICOS .....  | 83         |
| 6.1.1 PROBLEMA MATEMÁTICO TESTE .....   | 83         |
| 6.1.2 SISTEMAS ELÉTRICOS DE POTÊNCIA .....  | 87         |
| 6.1.2.1 SISTEMA ELÉTRICO IEEE 14 BARRAS .....   | 87         |
| 6.1.2.1 SISTEMA ELÉTRICO IEEE 30 BARRAS .....   | 90         |
| 6.1.2.3 SISTEMA ELÉTRICO IEEE 118 BARRAS .....  | 93         |
| 6.2 ANÁLISE DOS RESULTADOS .....  | 97         |
| <b>CONCLUSÕES.....</b>  | <b>100</b> |
| <b>TRABALHOS PUBLICADOS .....</b>   | <b>102</b> |
| <b>REFERÊNCIAS.....</b>   | <b>104</b> |
| <b>APÊNDICE A.....</b>  | <b>114</b> |
| A.1 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 14 BARRAS – ESTRATÉGIA (I) .....   | 114        |
| A.2 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 30 BARRAS – ESTRATÉGIA (I) .....   | 115        |
| A.3 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 118 BARRAS – ESTRATÉGIA (I) .....  | 118        |
| A.4 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 14 BARRAS – ESTRATÉGIA (IV).....   | 128        |
| A.5 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 30 BARRAS – ESTRATÉGIA (IV).....   | 129        |
| A.6 ARQUIVO DE SAÍDA PARA O SISTEMA IEEE 118 BARRAS – ESTRATÉGIA (IV).....  | 132        |

## LISTA DE SÍMBOLOS

$f(x)$  – função objetivo;

$g(x)$  – vetor das restrições de igualdade;

$h(x)$  – vetor das restrições de desigualdade do tipo “menor ou igual”;

$x$  – vetor das variáveis primais;

$x_1$  e  $x_2$  – vetor das variáveis primais;

$s$  – vetor das variáveis de folga;

$\in$  – pertence a;

$R, E, N$  – conjunto dos números reais;

$\alpha$  – passo;

$d$  – direção;

$\nabla$  – vetor gradiente;

$B$  – matriz simétrica e não singular;

$\omega$  – minimizador global da função;

$c$  – constante entre 0 e 1;

$\rho$  – constantes entre 0 e 1;

$x^*$  – solução ótima;

$\mu$  – parâmetro de barreira;

$L$  – função Lagrangiana;

$\lambda$  – multiplicador de Lagrange;

$i, j, k, l, m, n, p$  – números inteiros positivos;

$W$  – matriz Hessiana;

$Q$  – matriz simétrica;

$g$  – gradiente;

$F(x)$  – matriz Hessiana no Método Geral dos Gradientes Conjugados;

$p$  – vetor passo;

$m_k$  – problema quadrático;

$\Delta$  – raio da região de confiança;

$sred$  – redução suficiente;

$pred$  – redução prevista;

$ared$  – redução atual;

$\eta$  – um número positivo;

$\nabla^2$  – matriz Hessiana;

$\tau$  – valor escalar empírico;

$p^C$  – ponto Cauchy;

$p^S$  – vetor da versão linear;

$p^N$  – passo inteiro de Cauchy;

$z = (x, s)$  – vetores das variáveis;

$c(z) = \begin{bmatrix} h(x) \\ g(x) + s \end{bmatrix}$  – vetor de restrições;

$\lambda = (\lambda_h, \lambda_g)$  – vetor dos multiplicadores de Lagrange;

$\lambda_h$  – multiplicadores de Lagrange associados às restrições de igualdade;

$\lambda_g$  – multiplicadores de Lagrange associados às restrições de desigualdade;

$S$  – matriz diagonal dada pelos vetores  $s$ ;

$\Lambda_g$  – matriz diagonal dada pelos vetores  $\lambda_g$ ;

$A_h$  – matriz Jacobiana associada à restrição  $h$ ;

$A_g$  – matriz Jacobiana associada à restrição  $g$ ;

$d = (d_z, d_\lambda)$  – vetor de direções;

$d_z$  – vetores de direções primais;

$d_\lambda$  – vetores de direções duais;

$I$  – matriz identidade;

$\alpha_z$  – tamanhos de passos associados ao vetor  $z$ ;

$\alpha_\lambda$  – tamanhos de passos associados ao vetor  $\lambda$ ;

$\phi$  – função mérito;

$\varphi$  – função barreira;

$\nu$  – parâmetro penalidade;

$D\phi$  – derivada da função mérito;

$d = (d_z^{CSO}, d_\lambda^{CSO})$  – vetor de direções das variáveis  $z$  e  $\lambda$

$\bar{d}_z = (\bar{d}_x, \bar{d}_s)$  – vetor de direções das variáveis  $x$  e  $s$ ;

$\bar{d}_\lambda = (\bar{d}_h, \bar{d}_g)$  – vetor de direções das restrições  $h$  e  $g$ ;

$\gamma_z^{CSO}$  e  $\gamma_\gamma^{CSO}$  – escalares positivos;

$T$  – conjunto chamado região de confiança;

$\psi$  – um número igual a 1 ou 0 (zero);

$\Sigma$  – vetor dado pelas matrizes  $S^{-1}$  e  $\Lambda_g$ ;

$r = (r_h, r_g)$  – vetor residual associados às restrições de igualdade  $h$  e desigualdade  $g$ ;

$v$  – passo normal;

$G$  – matriz definida positiva;

$P$  – operador de pré-condicionamento e projeção;

$D$  – matriz simétrica definida positiva;

$u$  – algum vetor pertencente a  $R^{n-1}$ ;

$Z$  – matriz definida positiva;

$\varepsilon^{opt}$  e  $\varepsilon^{feas}$  – tolerâncias para os testes de parada;

$\theta$  – parâmetro fixo igual a 1;

$\sigma$  – parâmetro de centralização;

$\xi$  – desvio do menor produto de complementaridade;

$\mu^{aff}$  – parâmetro de barreira afim;

$(d_z^{aff}, d_\lambda^{aff})$  – passo escala afim das variáveis primais e duais;

$d^{corr}$  – passo corretor;

$d\Lambda^{aff}$  e  $dS^{aff}$  – matrizes diagonais dos elementos  $d\lambda^{aff}$  e  $ds^{aff}$ ;

$q_n$  – função qualidade para o caso não linear;

$q_L$  – função qualidade para o caso linear;

## LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1 - Direção do Método de Região de Confiança. ....   | 22 |
| Figura 2 - Uma visão esquemática do pacote <i>Knitro</i> .....  | 33 |
| Figura 3 - Representação geométrica do problema (115) – (119). ....   | 83 |
| Figura 4 - Sistema elétrico IEEE 14 barras.....   | 88 |
| Figura 5 – Magnitude de tensões no sistema elétrico IEEE 14 barras para as estratégias de atualização do parâmetro de barreira .....          | 89 |
| Figura 6 - Sistema elétrico IEEE 30 barras.....   | 91 |
| Figura 7 – Magnitude de tensões no sistema elétrico IEEE 30 barras para as estratégias de atualização do parâmetro de barreira .....          | 92 |
| Figura 8 - Sistema elétrico IEEE 118 barras.....  | 94 |
| Figura 9 – Magnitude de tensões no sistema elétrico IEEE 118 barras para as estratégias de atualização do parâmetro de barreira .....         | 96 |
| Figura 10 – Relação entre quantidade de iterações com cada estratégia de atualização para os sistemas elétricos IEEE 14, 30 e 118 barras..... | 98 |

## LISTA DE TABELAS

|  |    |
|--|----|
| Tabela 1 – Resultado das diferentes estratégias de atualização do parâmetro de barreira disponível no <i>Knitro</i> para o problema teste matemático ..... | 85 |
| Tabela 2 – Estatísticas finais para o problema matemático teste.....   | 86 |
| Tabela 3 – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 14 barras.....  | 88 |
| Tabela 4 – Estatísticas finais para o sistema elétrico IEEE 14 barras. ....  | 89 |
| Tabela 5 – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 30 barras.....  | 91 |
| Tabela 6 – Estatísticas finais para o Sistema Elétrico 30 barras.....  | 92 |
| Tabela 7 – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 118 barras.....   | 95 |
| Tabela 8 - Estatísticas finais para o sistema elétrico 118 barras.....   | 95 |

## RESUMO

Ferreira, E. C. (2013). *Uma investigação da influência do parâmetro de barreira na solução do problema de fluxo de potência ótimo*. Dissertação (Mestrado) - Faculdade de Engenharia de Bauru, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru, 2013.

Muitos autores mostram a viabilidade do uso de pacotes de algoritmos de otimização, com métodos específicos, para resolução de problemas reais. Neste trabalho destacamos o pacote de algoritmos de otimização *Knitro*, o qual foi projetado para resolver problemas de programação não linear e de grande porte. A sua implementação é composta de um Método de Pontos Interiores e um Método de Região de Confiança. No Método de Pontos Interiores, o problema de programação não linear é substituído por uma sequência de subproblemas de barreira, os quais são controlados por um parâmetro de barreira. Caso o algoritmo de ponto interior não forneça garantia de convergência global, isto é, ocorra a presença de não-convexidade e deficiências no posto da matriz Hessiana ou Jacobiana, um Método de Região de Confiança é chamado para garantir a convergência global, em que o cálculo do passo é feito por meio de um algoritmo gradiente conjugado. Muitos autores destacam, em seus trabalhos, que a convergência do Método de Pontos Interiores está diretamente ligada às estratégias de atualização do parâmetro de barreira, bem como à sua inicialização. Várias estratégias de atualização do parâmetro de barreira estão disponíveis no pacote de algoritmos de otimização *Knitro*. Neste trabalho, propomos utilizar o software *Knitro* e analisar a influência das diferentes estratégias de atualização do parâmetro de barreira disponíveis, na solução de problemas matemáticos e dos problemas de Fluxo de Potência Ótimo (FPO), estudado na Engenharia Elétrica, área de Sistemas Elétricos de Potência. Experimentos numéricos são apresentados com um problema matemático teste e com os sistemas elétricos IEEE 14, 30 e 118 barras.

**Palavras-chave:** Método de Pontos Interiores, Método de Região de Confiança, *Knitro* e Fluxo de Potência Ótimo.

## ABSTRACT

Ferreira, E. C. (2013). *An investigation of the influence of the barrier parameter in solving the problem of optimal power flow*. Dissertation (Master's degree) - Faculdade de Engenharia de Bauru, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru, 2013.

Many authors show the feasibility of using packet optimization algorithms with specific methods for solving real problems. In this paper we emphasize the package optimization algorithms Knitro, which was designed to solve problems of nonlinear programming and large. The implementation consists of an Interior Point Method and of Trust Region Method. In the interior point method, the nonlinear programming problem is replaced by a sequence of barrier subproblems, which is controlled by a barrier parameter. If the interior point algorithm does not provide guarantee of global convergence, that is occurring the presence of non-convexity and deficiencies in the post of the Hessian matrix or Jacobian, a is invoked Trust Region method to ensure global convergence, in which the calculation of step is done using a conjugate gradient algorithm. Many authors emphasize in their work, the convergence of Interior Point Method is directly linked to the strategies of updating the barrier parameter, as well as its startup. Various strategies for updating the barrier parameter are available in the optimization algorithms of package Knitro. In this paper, we propose to use the software Knitro and analyze the influence of different strategies for updating the barrier parameter available in the solution of mathematical problems and problems of Optimal Power Flow (OPF), studied in Electrical Engineering, Electrical Systems power area. Numerical experiments are presented with test a mathematical problem and electrical systems IEEE 14, 30 and 118 buses.

**Keywords:** Interior Point Method, Trust Region Method, Knitro and Optimal Power Flow.

# CAPÍTULO 1

## INTRODUÇÃO

O atual contexto do setor elétrico brasileiro corresponde às reformas em sua estruturação, onde a principal delas é oriunda de um modelo que visa o monopólio do setor para um modelo onde a competitividade se apresenta como principal característica, agregando a estes novos paradigmas e filosofias sobre a problemática do planejamento e da operação dos sistemas elétricos a partir de sua geração, transmissão e distribuição de energia elétrica em seu território.

Diante da matriz energética brasileira, esta mudança objetiva o benefício dos consumidores a fim de tornar plausível um maior índice qualitativo dessa energia distribuída e a confiabilidade de seu sistema. Para tanto, o estudo de potência reativa se apresenta com grande valorização diante das concessionárias de energia elétrica, uma vez que, estas agregam uma capacidade limitada em seu sistema de transmissão para acomodar cargas adicionais e na manutenção de um perfil de tensão e de fluxo de potência aliado aos inúmeros cenários de operação. Porém, em algumas análises, nota-se que o ponto de operação do sistema não compreende essas condições e, os operadores do sistema devem promover os ajustes das variáveis de controle objetivando um novo ponto de operação que satisfaça às premissas estabelecidas sobre a rede. Nota-se que as variáveis de controle poder-se-iam ser ajustadas a fim de otimizá-la de modo que maximize ou minimize um critério de desempenho do sistema de potência que visualize as restrições de igualdade e desigualdade. A problemática de otimização é conhecida como fluxo de potência ótimo (FPO). Um fator singular diante dessa otimização é o FPO reativo, em que os controles ativos são fixados e suas variáveis de controle relacionadas à potência reativa, como é o caso da magnitude da tensão dos geradores e dos compensadores síncronos e estáticos de potência reativa, além dos *taps* dos transformadores, poderão ser otimizados em relação à função objetivo.

O FPO é um problema não linear, restrito, não convexo e de grande porte. Como ferramenta, é utilizado para determinar o melhor ponto de operação de um

sistema elétrico de potência, contribuindo para a melhoria do desempenho deste. É estudado na Engenharia Elétrica, área de Sistemas Elétricos de Potência. Seu modelo foi proposto por Carpentier (1962) e desde então muitas metodologias foram testadas em sua resolução, entre elas destacamos os Métodos de Pontos Interiores.

A formulação de problemas reais como problemas de otimização não linear, como é o caso do FPO, é encontrada em várias áreas do conhecimento tais como, engenharia, matemática, agronomia, economia, entre outras. Estes problemas são muitas vezes difíceis de serem resolvidos, devido a sua não linearidade. Para amenizar essa dificuldade, muitos pesquisadores desenvolveram pacotes de algoritmos de otimização para contribuir na resolução destes tipos de problemas.

Destacamos neste trabalho, o pacote de otimização não linear denominado *Knitro*, o qual utiliza um Método de Ponto Interior (*Interior-Point*), no qual encontram-se implementados um método de busca linear e um de Região de Confiança, ou um Método do Conjunto Ativo (*Active Set*), ou possivelmente ambos, dependendo das características do problema a ser resolvido. Neste trabalho utilizamos o *Interior-Point*.

Segundo Byrd et al. (2006), o *Knitro* foi projetado para resolver problemas de grande porte e de acordo com Waltz e Platenga (2010), embora seja projetado principalmente para otimização não-linear em geral, ele mostra-se eficiente em resolver as seguintes classes de problemas de otimização: problemas irrestritos; sistemas de equações não-lineares; problemas de mínimos quadrados, lineares e não-lineares; problemas de programação linear e problemas de programação quadrática, convexos e não convexos; problemas matemáticos com restrições de complementaridade; problemas de programação restrita não-linear convexos e não convexos; programação linear inteira mista de porte médio e problemas de programação não linear inteira mista e convexa de porte médio.

Sua primeira versão foi criada e introduzida em julho de 2001, por Nocedal e Waltz (2001), e tem sido melhorada desde então ([www.ziena.com](http://www.ziena.com)), estando na versão 8.1 em 2012.

No *Knitro* estão disponíveis seis estratégias de atualizações do parâmetro de barreira, a saber: (i) o parâmetro de barreira é decrescido monotonicamente; (ii) o parâmetro de barreira é calculado por uma regra adaptativa com base na diferença de complementaridade; (iii) é realizada uma investigação sobre o passo (afim-escala) para determinar dinamicamente o valor do parâmetro barreira a cada iteração; (iv) é

utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira levando em conta o passo corretor; (v) é utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira em que o passo corretor é condicionado; e (vi) uma função qualidade é minimizada em cada iteração para determinar o parâmetro de barreira. A investigação destas estratégias é o objetivo principal de estudo do nosso trabalho.

Vários autores provaram a eficiência do pacote de algoritmos *Knitro*, como Platenga e Waltz (2006), em que seus estudos analisam o desempenho de *Knitro*, na versão 5, em problemas de otimização não-lineares e de grande porte. Os resultados deste estudo mostram que o *Knitro* pode resolver problemas com mais de 100 variáveis e restrições satisfatoriamente.

O *Knitro* também é utilizado na indústria de energia elétrica para muitas aplicações, mais frequentemente para determinar as condições operacionais que atingem o problema de FPO, de acordo com o sítio. (<http://www.ziena.com/elecpower.htm>).

Platenga (2006), utilizou o pacote de algoritmos *Knitro* na resolução do problema de FPO. O autor afirma que o *Knitro* é um software que resolve problemas de otimização não-lineares e de grande porte. A sua robustez, eficiência e facilidade de uso o tornam uma ferramenta poderosa em Gestão de Energia de Sistemas (SGA), com rápidos cálculos operacionais e planejamento de longo alcance. O *Knitro* pode otimizar custos de geradores, perdas nas linhas de transmissão, e muito mais, enquanto satisfaz a não-linearidade das equações de fluxo e as restrições do sistema. Atualmente o pacote de algoritmos *Knitro* é usado em redes de transmissão contendo um enorme número de barras.

Como já citado, o *Knitro* é um pacote de algoritmos de otimização que tem como base um Método de Pontos Interiores que utiliza busca linear ou um Método de Região de Confiança.

O Método de Pontos Interiores foi introduzido por Frisch (1955) e por Carrol (1961), sendo popularizado por Fiacco e McCormick (1968), com a finalidade de resolver problemas com restrições de desigualdades não lineares por meio do uso do método da função barreira.

Devido a problemas apresentados por essa base de método de pontos interiores, tais como, o mau condicionamento da matriz Hessiana quando seu fator de barreira tende a zero; a dificuldade na escolha do fator de barreira e na escolha de

uma solução inicial; a não existência da derivada na solução e o aumento ilimitado da função barreira na vizinhança da fronteira, o entusiasmo no uso da função barreira diminuiu sensivelmente na década de 70.

O interesse pela utilização da metodologia de Pontos Interiores para a busca de soluções ótimas de problemas de otimização reapareceu quando, em 1984, Karmarkar (1984) publicou o seu método Projetivo para Programação Linear. Este trabalho provocou uma agitação nas atividades de pesquisa nesta área.

Após a introdução feita por Karmarkar, muitos métodos variantes de seu algoritmo original foram apresentados.

Já os Métodos de Região de Confiança constituem uma classe relativamente nova de algoritmos de otimização que minimizam um problema de otimização quadrática através de uma função objetivo não linear dentro de uma região chamada de região de confiança. Nesta região o problema quadrático pode ser confiado como uma boa aproximação para a função objetivo não linear original. Os Métodos de Região de Confiança diferem entre si na forma que modelam a função objetivo e tratam as restrições, de acordo com Nocedal e Wright (1999) e Conn et al. (2000), principalmente as restrições de desigualdades.

Na prática, os algoritmos escolhem o tamanho da região de confiança de acordo com o desempenho do algoritmo durante uma prévia iteração. Se o problema quadrático está reagindo bem, produzindo bons passos e tendo-se uma previsão do comportamento da função objetivo ao longo desses passos, o tamanho da região de confiança continua decrescente ao longo dos passos que são feitos. Por outro lado, o fracasso de um passo indica que o problema não é uma boa representação da função objetivo da região de confiança atual, então é reduzido o tamanho da região e reinicia-se o processo.

Propomos neste trabalho utilizar o pacote de algoritmos de otimização *Knitro* e analisar a influência das estratégias de atualização do parâmetro de barreira, na solução de um problema matemático teste e do problema de Fluxo de Potência Ótimo.

Este trabalho encontra-se dividido em 8 capítulos.

No capítulo 2, apresentamos uma revisão de alguns métodos de otimização utilizados para resolução de problemas não lineares.

No capítulo 3, descrevemos o Método de Ponto Interior com Busca Linear e Região de Confiança que está implementado no pacote de algoritmos *Knitro*.

No capítulo 4, apresentamos as seis estratégias de atualização do parâmetro de barreira disponíveis no pacote de algoritmos *Knitro*, as quais são utilizadas neste trabalho como objetivo de estudo.

No capítulo 5, apresentamos um histórico do problema de FPO, sua definição e sua formulação.

No capítulo 6, apresentamos os resultados numéricos do método para um problema matemático teste e para os sistemas elétricos IEEE 14, 30 e 118 barras.

No capítulo 7, apresentamos as conclusões e as perspectivas de continuação deste trabalho.

Finalmente, no capítulo 8, apresentamos a relação dos trabalhos publicados e apresentados em congressos e reuniões científicas.

## CAPÍTULO 2

### REVISÃO DE MÉTODOS DE OTIMIZAÇÃO

Neste capítulo apresentamos uma revisão sobre os métodos de Busca Linear, Método Primal Dual Barreira Logarítmica, Método de Direções Conjugadas e Método de Região de Confiança, fornecendo assim, uma visão ampla do desenvolvimento teórico de cada método que está implementado no pacote *Knitro*. Apresentamos também um histórico sobre o pacote de algoritmos de otimização *Knitro*, no qual descrevemos o software que será utilizado, de uma maneira geral.

#### 2.1 Métodos de Busca Linear

Seja o problema de otimização irrestrita:

$$\begin{aligned} &\text{minimizar } f(x) \\ &x \in \mathbb{R}^n. \end{aligned} \tag{1}$$

Os métodos irrestritos para o cálculo de uma solução para o problema (1) consistem em dado um ponto  $x^k$  na iteração  $k$ , determinar uma direção  $d^k$  e então decidir o quanto deve-se mover ao longo dessa direção, isto é:

$$x^{k+1} = x^k + \alpha_k d^k \tag{2}$$

em que  $\alpha_k$  é denominado tamanho do passo. Para o problema (1) é exigido que as direções sejam direções de descida para as quais  $d^T \nabla f(x) < 0$ , pois essa propriedade é a garantia de que a função  $f(x)$  pode ser reduzida ao longo desta direção.

Nestes métodos a direção de busca, muitas vezes tem a forma:

$$d = -B^{-1}\nabla f(x) \quad (3)$$

em que  $B$  é uma matriz simétrica e não singular. Por exemplo, no Método do Gradiente,  $B$  é simplesmente a matriz identidade, no Método de Newton  $B$  é a matriz Hessiana  $\left(\nabla^2 f(x)\right)$  e no Método Quasi-Newton,  $B$  é uma aproximação da matriz Hessiana que é atualizada a cada iteração por meio de uma fórmula pré-estabelecida.

Quando definimos a direção  $d$  por (3) e  $B$  é definida positiva, temos:

$$d^T \nabla f(x) = -\nabla f(x)^T B^{-1} \nabla f(x) < 0 \quad (4)$$

e, portanto,  $d$  é uma direção de descida.

Na seção seguinte detalhamos a escolha do tamanho do passo  $\alpha_k$ .

### 2.1.1 A escolha do tamanho do passo da busca linear

De acordo com Nocedal e Wright (1999), ao calcular o tamanho do passo  $\alpha_k$ , gostaríamos de obter uma redução significativa da função  $f(x)$  em (1), mas, ao mesmo tempo, não gastar muito tempo ao fazer essa escolha.

A escolha ideal seria o minimizador global da função  $\omega(\cdot)$  definida por:

$$\omega(\alpha) = f(x + \alpha d), \quad \alpha > 0. \quad (5)$$

Existem diversos métodos para o cálculo desse minimizador global, tais como: Busca Dicotômica, Secção Áurea, Newton, Bisseção, entre outros, os quais podem ser encontrados em Luenberger (1984) e Bazaraa (1993) e são denominados Métodos de Busca Linear exata. A utilização desses métodos geralmente exige um esforço computacional considerável, pois podem-se realizar diversas avaliações da função objetivo e também exigir informação de suas derivadas.

Ao invés de determinar um otimizador global para (5), podemos ficar satisfeitos em encontrar um minimizador local de precisão moderada. Desta forma

temos os métodos de busca linear inexata, os quais tem por objetivo a diminuição da função objetivo mediante a satisfação de uma condição. Neste trabalho utilizamos o Método de Busca Linear inexata denominado *backtracking*.

Destacamos duas condições a serem satisfeitas por esses métodos:

- **Condições de Wolfe**

A abordagem de diminuição da função objetivo, conhecida como redução suficiente, é uma condição da busca linear inexata que estipula que  $\alpha_k$  deverá reduzir suficientemente a função objetivo  $f(x)$ , e é dada pela seguinte desigualdade:

$$f(x^k + \alpha d^k) \leq f(x^k) + c_1 \alpha \nabla f(x^k)^T d^k \quad (6)$$

para qualquer constante  $c_1 \in (0,1)$ . Em outras palavras, a redução da  $f(x)$  deve ser proporcional tanto ao tamanho do passo quanto à derivada direcional  $\nabla f(x)^T d$ . A desigualdade (6) muitas vezes é denominada de condição de Armijo.

A condição de redução suficiente não garante por si só que o algoritmo faça um progresso razoável, pois é satisfeita para todos os valores suficientemente pequenos de  $\alpha$ . Para excluir passos muito curtos, apresentamos uma segunda exigência, chamada de condição de curvatura, no qual  $\alpha_k$  deve satisfazer:

$$\nabla f(x^k + \alpha_k d^k)^T d^k \geq c_2 \nabla f(x^k)^T d^k \quad (7)$$

para qualquer constante  $c_2 \in (c_1,1)$ , em que  $c_1$  é a constante de (6). Note que o lado esquerdo é simplesmente a derivada parcial, denominada  $\omega'(\alpha_k)$ , então a condição de curvatura garante que a inclinação de  $\omega(\alpha_k)$  é maior do que  $c_2$  vezes o gradiente de  $\omega'(0)$ . Isso faz sentido porque se a inclinação  $\omega'(\alpha)$  é fortemente negativa, temos uma indicação de que podemos reduzir  $f(x)$  significativamente movendo mais ao longo da direção escolhida. Por outro lado, se a inclinação é somente ligeiramente negativa ou mesmo positiva, é um sinal de que não se pode esperar muita diminuição

em  $f(x)$  nesta direção, então é conveniente terminar a busca linear. Valores típicos para  $c_2$  são 0,9 quando a direção de busca  $d^k$  é obtida por um Método Newton ou quasi-Newton, e 0,1 quando  $d^k$  é obtido a partir de um Método de Gradiente Conjugado não linear.

As condições de redução suficiente (6) e de curvatura (7) são conhecidas como as condições de Wolfe.

Um tamanho de passo pode satisfazer as condições de Wolfe sem estar particularmente próximo de um minimizador de  $\omega$ . Podemos, contudo, modificar a condição de curvatura para forçar  $\alpha_k$  encontrar-se em uma vizinhança de um minimizador local ou ponto estacionário de  $\omega$ . As condições de Wolfe, denominadas fortes, requerem que  $\alpha_k$  satisfaça:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + c_1 \alpha_k \nabla f(x^k)^T d^k, \quad (8)$$

$$|\nabla f(x^k + \alpha_k d^k)^T d^k| \leq c_2 |\nabla f(x^k)^T d^k|, \quad (9)$$

com  $0 < c_1 < c_2 < 1$ . A única diferença das condições de Wolfe, (6) e (7); (8) e (9), é que estas não permitem que a derivada  $\omega'(\alpha)$  seja positiva. Assim, são excluídos pontos que estão longe dos pontos fixos de  $\omega$ .

A prova de que existem tamanho de passos que satisfaçam as condições de Wolfe pode ser encontrada em Nocedal e Wright (1999).

- **Condições de Goldstein**

As condições de Goldstein asseguram que o tamanho do passo  $\alpha_k$  forneça uma redução suficiente da função objetivo como as condições de Wolfe, evitando que o passo  $\alpha_k$  seja muito pequeno. As condições de Goldstein podem ser expressas pelas desigualdades:

$$f(x^k) + (1-c)\alpha_k \nabla f(x^k)^T d \leq f(x^k + \alpha_k d^k) \leq f(x^k) + c\alpha_k \nabla f(x^k)^T d^k \quad (10)$$

em que  $c$  é uma constante tal que  $0 < c < \frac{1}{2}$ .

A segunda desigualdade em (10) corresponde à condição de redução suficiente em (6), enquanto que a primeira desigualdade é introduzida para controlar o tamanho do passo.

A primeira desigualdade de (10) é uma desvantagem da condição de Goldstein em relação às condições de Wolfe, pois ela pode excluir todos os minimizadores de  $\omega$ , no entanto, destacamos que ambas as condições são semelhantes em suas teorias de convergência.

### 2.1.1.1 A busca *Backtraking*

Como mencionado anteriormente, a condição de redução suficiente de Wolfe (6), não assegura, por si só, que o algoritmo progrida razoavelmente ao longo da direção de busca. No entanto, se o algoritmo de busca linear escolhe seus candidatos à tamanho de passo de forma apropriada, utilizando uma abordagem denominada *backtraking*, nós podemos dispensar a condição de Wolfe (7) e usar somente a condição (6) para terminar o processo. O algoritmo *backtraking* é visto a seguir:

---

**Algoritmo 1:** *Backtraking*

---

**Início**

Escolha  $\bar{\alpha} > 0, \rho \in (0,1), c \in (0,1)$ ;

Defina  $\alpha \leftarrow \bar{\alpha}$ ;

**Repita** até que  $f(x^k + \alpha d^k) \leq f(x^k) + c\alpha \nabla f(x^k)^T d^k$ ;

$\alpha \leftarrow \rho\alpha$ ;

**Fim**

Atualize  $\alpha_k = \alpha$ ;

**Fim**

---

Neste procedimento, o tamanho de passo  $\bar{\alpha}$  é inicialmente escolhido igual a 1 no Método de Newton e quasi-Newton, mas pode ter valores diferentes em outros algoritmos como no Método do Gradiente ou Gradiente Conjugado. Após um número finito de iterações, um tamanho de passo factível será encontrado, pois  $\alpha_k$  se tornará suficientemente pequeno para que se mantenha a condição de redução satisfeita. Na prática, é permitido variar em cada iteração da busca linear o fator de contração  $\rho$ . Precisamos garantir apenas que a cada iteração  $\rho \in [\rho^{\min}, \rho^{\max}]$ , para algumas constantes fixas  $0 < \rho^{\min} < \rho^{\max} < 1$ .

A abordagem *backtraking* garante que o tamanho de passo  $\alpha_k$  selecionado é um valor fixo em relação à escolha inicial  $\bar{\alpha}$  e é pequeno o suficiente para satisfazer a condição de redução (6).

## 2.2 O Método Primal-Dual Barreira Logarítmica

O primeiro Método de Pontos Interiores foi introduzido por Frisch (1955) e por Carrol (1961), e popularizado por Fiacco e McCormick (1968), com a finalidade de resolver problemas com restrições de desigualdades não lineares através do uso do Método da Função Barreira.

O interesse no uso da função barreira diminuiu na década de 70 devido a alguns problemas tais como: o mal condicionamento da matriz Hessiana da função barreira quando seu parâmetro tende a zero; a dificuldade na escolha do parâmetro de barreira e na escolha de uma solução inicial; a não-existência da derivada na solução e o aumento ilimitado da função barreira na vizinhança da fronteira.

Posteriormente, Karmakar (1984) apresentou um novo Método de Pontos Interiores para programação linear, afirmando que podia obter soluções cinquenta vezes mais rápidas do que o Método Simplex para problemas de grande porte. Este trabalho reativou o interesse em pesquisas nesta área, os quais podemos citar: o algoritmo Primal-Afim utilizado na resolução de problemas de Programação Linear com restrições de igualdade apresentado por Barnes (1986) e por Vanderbei et al. (1986); o algoritmo Dual-Afim para resolver problemas de Programação Linear na forma de desigualdade proposto por Adler et al. (1989); a contribuição de Megiddo &

Shub (1989) com a indicação que a trajetória que conduz à solução ótima fornecida pelos algoritmos Afins dependem da solução inicial; a incorporação da função Barreira Logarítmica ao problema de Programação Linear e a resolução deste através da metodologia de Pontos Interiores Primal-Afim e Dual-Afim, destacando-se os trabalhos de Megiddo (1987), Renegar (1988), Vaidya (1990) e Ye (1986); os métodos de trajetória central, propostos por Gonzaga (1989, 1990) e Monteiro & Adler (1989); o algoritmo Primal-Dual de Pontos Interiores proposto por Monteiro et al. (1990) e também por Kojima et al. (1989), os quais exploram uma função potencial Primal-Dual variante da função Barreira Logarítmica e o método da Barreira Logarítmica Primal-Dual Previsor-Corretor, em que a cada iteração é dado um passo previsor e um passo corretor, determinando direções de busca melhores que as apresentadas por Monteiro & Adler (1989) e Monteiro et al. (1990).

Os Métodos de Barreira transformam o problema restrito em um problema irrestrito e introduzem as restrições na função objetivo através de um parâmetro de barreira. Tais parâmetros geram barreiras que impedem as variáveis de violarem seus limites trabalhando no interior dessa região. Logo, parte-se de um ponto interior factível e geram-se novos pontos interiores factíveis. Uma das vantagens desse método é a obtenção de, pelo menos, uma solução factível, caso ocorra uma parada prematura do mesmo, pois esse método trabalha somente com problemas de desigualdade cujo interior é não-vazio.

Considere o problema de programação não linear restrita:

$$\begin{aligned} & \text{minimizar } f(x) \\ & \text{sujeito a: } g_i(x) = 0; \quad i = 1, 2, \dots, m \\ & \quad \quad \quad h_j(x) \leq 0; \quad j = 1, 2, \dots, p \end{aligned} \tag{11}$$

sendo:  $m < n$ ,  $x \in R^n$ ,  $g(x) \in R^m$ ,  $h(x) \in R^p$ , e as funções  $f$ ,  $g$  e  $h$  são de classe  $C^2$ .

A resolução do problema (11) pelo Método Primal Dual Barreira Logarítmica exige a transformação das restrições de desigualdades em restrições de igualdades por meio da introdução de variáveis de folga ou excesso positivas.

Portanto, o problema (11) modificado pode ser apresentado como:

$$\begin{aligned}
& \text{minimizar } f(x) \\
& \text{sujeito a: } g_i(x) = 0; \quad i = 1, 2, \dots, m \\
& \quad \quad \quad h_j(x) - s_j = 0; \quad j = 1, 2, \dots, p
\end{aligned} \tag{12}$$

sendo  $s \in \mathcal{R}^p$  denominado de vetor das variáveis de folga, com  $s_j > 0$ , para  $j = 1, \dots, p$ .

Associa-se o problema modificado (12) a uma função barreira logarítmica a qual garante a não negatividade dessas variáveis de folga:

$$\begin{aligned}
& \text{minimizar } f(x) - \mu \sum_{j=1}^p \ln(s_j) \\
& \text{sujeito a: } g_i(x) = 0; \quad i = 1, 2, \dots, m \\
& \quad \quad \quad h_j(x) + s_j = 0; \quad j = 1, 2, \dots, p
\end{aligned} \tag{13}$$

sendo  $\mu$  o parâmetro de barreira, o qual é um numero positive que tende a zero.

Quando o parâmetro de barreira tende a zero, a solução do problema apresentado em (13), aproxima-se da solução ótima,  $x^*$ , do problema (11).

Ao problema (13) é associado uma função Lagrangiana:

$$L = f(x) - \mu \sum_{j=1}^p \ln(s_j) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{j=1}^p \pi_j [h_j(x) + s_j] \tag{14}$$

em que  $\lambda_i, i = 1, 2, \dots, m$  e  $\pi_j, j = 1, 2, \dots, p$  são as coordenadas dos vetores dos multiplicadores de Lagrange associados às restrições de igualdade e desigualdade, respectivamente.

São aplicadas as condições necessárias de primeira-ordem em (14), gerando:

$$\nabla L(w) = 0 \tag{15}$$

sendo:  $w = (x, s, \lambda, \pi)^T$ .

A equação (15) representa um sistema de equações não-lineares, o qual é resolvido pelo Método de Newton, que gera um sistema do tipo  $Wd = -\nabla L(w)$ . Nesse sistema  $W$  é a matriz Hessiana da função Lagrangiana (14);  $\nabla L(w)$  é o vetor

gradiente; e o vetor das direções de busca  $d = (d_x, d_s, d_\lambda, d_\pi)^T$  é utilizado para atualizar as variáveis  $x$ ,  $s$ ,  $\lambda$  e  $\pi$  como segue:

$$\begin{aligned}
 x^{k+1} &= x^k + \alpha_p d_x \\
 s^{k+1} &= s^k + \alpha_p d_s \\
 \lambda^{k+1} &= \lambda^k + \alpha_d d_\lambda \\
 \pi^{k+1} &= \pi^k + \alpha_d d_\pi
 \end{aligned} \tag{16}$$

sendo escolhido o comprimento de passo  $\alpha_p \in (0,1]$  e  $\alpha_d \in (0,1]$  para preservar a positividade do vetor  $s$  e o sinal do vetor  $\pi$ .

---

**Algoritmo 2:** Método Primal Dual Barreira Logarítmica

---

**Início**

Dado o problema (11), construa a função Lagrangiana (14);

Dê estimativa inicial para  $\mu_k$  e  $d^k = (x^k, s^k, \lambda^k, \pi^k)$  que satisfaça as condições propostas;

Defina  $k = 0$ ;

**Repita** até que a norma do gradiente seja maior que uma precisão  $\varepsilon$

Obtenha o sistema  $W\Delta d = -\nabla L(w)$  e resolva-o;

Calcule os passos primais e duais e atualize  $d^k$  utilizando (16);

Se as condições de Karush–Kuhn–Tucker (KKT) são satisfeitas, pare.

**Fim**

Atualize  $\mu$

Faça  $k = k + 1$ .

**Fim**

---

Um ponto estritamente factível não é obrigatório, mas as condições  $s > 0$  e  $\pi > 0$  devem ser satisfeitas em todos os pontos. O processo de otimização termina quando as condições de KKT são satisfeitas. Destacamos que existem

diversas maneiras de atualizar e inicializar o parâmetro de barreira  $\mu$  e estas influenciam na convergência do método.

## 2.3 Métodos de Direções Conjugadas

Segundo Luenberger e Ye (2008), os Métodos de Direções Conjugadas, são intermediários aos Métodos do Gradiente e de Newton. A ideia era melhorar a convergência do Método do Gradiente e resolver os problemas de avaliar, armazenar e inverter a matriz Hessiana no Método de Newton.

Considere o seguinte problema quadrático:

$$\min f(x) = \frac{1}{2}x^T Qx - b^T x, \quad (17)$$

em que  $Q \in R^{n \times n}$  é uma matriz simétrica definida positiva.

Para compreensão do Método de Direções Conjugadas, vamos analisar o *Teorema 1* utilizando a definição de direções conjugadas.

**Definição 1:** Dada uma matriz simétrica  $Q$ , os vetores  $d^1$  e  $d^2$  são denominados  $Q$ -*Ortogonalis ou conjugados* em relação à  $Q$ , se  $(d^1)^T Qd^2 = 0$ .

**Teorema 1:** Seja  $\{d^i\}_{i=1}^{n-1}$  um conjunto de vetores  $Q$ -ortogonais não-nulos. Para qualquer  $x^0 \in E^n$ , a sequência  $x^k$  gerada de acordo com a equação:

$$x^{k+1} = x^k + \alpha_k d^k, \quad k \geq 0 \quad (18)$$

em que:

$$\alpha_k = \frac{-(g^k)^T d^k}{(d^k)^T Qd^k} \quad (19)$$

e

$$g^k = Qx^k - b \quad (20)$$

converge para a solução única,  $x^*$ , de  $Qx = b$  após  $n$  passos, ou seja,  $x^n = x^*$ .

Em (20),  $g^k$  é o gradiente de  $f(x)$  apresentada em (17).

A prova do teorema 1 encontra-se em Luenberger e Ye (2008).

O termo  $\alpha_k = \frac{-(g^k)^T d^k}{(d^k)^T Q d^k}$  é a grande diferença deste método, pois os valores

de  $\alpha_k$  podem ser determinados sem conhecermos o valor da solução ótima  $x^*$ .

### 2.3.1 O Método do Gradiente Conjugado

Segundo Wright e Nocedal (1999), o primeiro Método do Gradiente Conjugado Não-Linear foi introduzido por Fletcher e Reeves na década de 1960 e é uma das técnicas mais conhecidas para a solução de problemas de otimização não linear de grande porte. Ao longo dos anos, várias abordagens foram propostas, e algumas são amplamente usadas na prática. As principais características destes algoritmos são que eles não necessitam de armazenamento de matriz Hessiana e são mais rápidos que o Método do Gradiente.

De acordo com Luenberger e Ye (2008), o Método do Gradiente Conjugado é um método de direção conjugada, em que as direções não são especificadas previamente, mas são determinadas sequencialmente em cada passo da iteração. No passo  $k$  avalia-se o vetor gradiente negativo atual e acrescenta-se a ele uma combinação linear dos vetores das direções anteriores para obter um novo vetor de direções conjugadas ao longo do qual o método se move.

Existem três vantagens nesta forma de seleção da direção. Em primeiro lugar, o gradiente é sempre diferente de zero e linearmente independente de todos os vetores direção anteriormente obtidos. Em segundo lugar, a fórmula para o cálculo do novo vetor de direção é explícita e disponível. Esta particularidade faz com que o método seja mais complicado do que o Método do Gradiente. Em terceiro lugar, pelo fato das direções serem baseadas na direção dos gradientes, o processo faz, uniformemente bons progressos na direção da solução em cada passo. Esses progressos uniformes são importantes para generalização em problemas não quadráticos.

---

**Algoritmo 3:** Método do Gradiente Conjugado caso quadrático

---

**Início**

Defina  $k = 0$

Defina  $d^0 = -g^0 = b - Qx^0$ , para qualquer  $x^0 \in E^n$

Calcule:

$$\alpha_k = -\frac{(g^k)^T d^k}{(d^k)^T Q d^k}; \quad (21)$$

$$\beta^k = \frac{(g^{k+1})^T Q d^k}{(d^k)^T Q d^k}; \quad (22)$$

$$d^{k+1} = -g^{k+1} + \beta^k d^k; \quad (23)$$

$$x^{k+1} = x^k + \alpha_k d^k; \quad (24)$$

em que  $g^k = Qx^k - b$ .

$k = k + 1$

**Fim**

---

O primeiro passo do algoritmo é idêntico a um passo do gradiente, nos demais passos o método move-se em uma direção que é uma combinação linear do gradiente atual e do vetor direção anterior. A característica interessante do algoritmo são as fórmulas, (22) e (23), utilizadas para atualizar o vetor direção. A implementação deste método é mais complicada que a do Método do Gradiente, mas converge iterativamente em um número finito de passos.

### 2.3.2 Método Geral dos Gradiente Conjugados

Segundo Nocedal e Wright (1999), ao estender o Método dos Gradiente Conjugados para problemas não quadráticos, são realizadas as seguintes aproximações em  $x^k$ ;  $g^k$  é substituído por  $\nabla f(x^k)^T$  e a matriz  $Q$  por  $F(x^k)$ , a qual é a matriz Hessiana.

Quando o Método dos Gradientes conjugados é utilizado em problemas não-quadráticos, geralmente o processo não termina em  $n$  passos como no caso quadrático. Uma alternativa para isso é buscar novas direções de acordo com o algoritmo 4 e somente finalizar a busca após um critério de parada ser satisfeito, sendo esse um processo conhecido como “restart”.

Portanto o Método Geral dos Gradientes Conjugados é definido como segue:

---

**Algoritmo 4:** Método Geral dos Gradientes Conjugados

---

**Início**

Dado valor inicial para  $x^k$ , calcule  $g^k = \nabla f(x^k)^T$ ;

Defina  $d^k = -g^k$

Defina  $k = 0, 1, \dots, n-1$

Calcule  $x^{k+1} = x^k + \alpha_k d^k$ , onde  $\alpha_k = -\frac{g^T d^k}{d^{kT} F(x^k) d^k}$ ;

Defina  $g^{k+1} = \nabla f(x^{k+1})^T$ ;

**Repita** até que  $k = n-1$ ;

Calcule  $d^{k+1} = -g^{k+1} + \beta^k d^k$  onde  $\beta^k = \frac{(g^{k+1})^T F(x^k) d^k}{(d^k)^T F(x^k) d^k}$

Calcule novamente  $x^{k+1} = x^k + \alpha_k d^k$ ;

**Fim**

Defina  $x^0 \leftarrow x^n$

$k = k + 1$

**Fim**

---

Neste método não há necessidade de usar um algoritmo de busca linear, pois o passo está definido no próprio método, já que cada passo  $x^{k+1}$  é obtido deslocando  $x^k$  de um passo  $\alpha_k$  na direção  $d^k$  e isso é uma vantagem. Como desvantagem destaca-se a necessidade de calcular  $F(x^k)$  em cada ponto, gerando um esforço computacional maior e ainda, esse cálculo faz com que o método não seja globalmente convergente.

## 2.4 Métodos Região de Confiança

De acordo com Conn et al. (2000), o conceito de região de confiança amadureceu nos últimos 50 anos e está enraizado na estimação de parâmetros na área não-linear. O primeiro trabalho nesta área parece ser o de Levenberg (1944), que considera a adição de um múltiplo da identidade à matriz Hessiana como um procedimento de estabilização/amortecimento no contexto das soluções de problemas de mínimos quadrados não lineares.

Neste ponto de vista, foi desenvolvido por Morrison (1960), um trabalho em que a convergência do algoritmo de estimação é melhorada através da minimização de um problema quadrático de mínimos quadrados da função objetivo em uma esfera de raio constante. Morrison revelou que a solução de um sistema linear envolvendo a Hessiana do problema, aumentada por um múltiplo da matriz identidade, produz a solução do subproblema através de uma escolha adequada dos parâmetros de amortecimento associados, e também que o decréscimo do problema é monótono neste parâmetro. Uma técnica baseada na decomposição do autovalor é dada para calcular o mínimo da função escolhida dentro da esfera.

A mesma ideia fundamental é encontrada no artigo de Marquardt (1963), que indicou a ligação entre o amortecimento da matriz Hessiana e a redução do comprimento do passo e também mostrou que, minimizar uma função amortecida corresponde à minimizar a função original numa região restrita.

Levenberg (1944), Morrison (1960) e Marquardt (1963), fazem a observação que a Hessiana pode ser aproximada, quando temos uma função objetivo quadrática.

Outra importante contribuição conceitual é a de Griffith e Stewart (1961), que introduziu a ideia de minimizar sucessivamente a função objeto e as restrições ao redor da região atual. Seu problema é linear e a região é definida por limites sobre as

variáveis, conduzindo a um subproblema de programação linear. No entanto, eles não propõem ajustes ao tamanho da região de confiança, mas ao invés disso, a mantém constante durante os cálculos.

Um passo fundamental foi feito por Goldfeldt, Quandt e Trotter (1966), que introduziram um procedimento de atualização explícita para o tamanho do passo máximo. Eles viram esse parâmetro como um amortecimento da matriz Hessiana, o qual induz a uma restrição no tamanho do passo, ou como uma restrição do tamanho do passo calculado pelo amortecimento da Hessiana, isso não está claro no artigo. Entretanto, o procedimento de atualização é muito semelhante ao usado normalmente. Em particular, eles apresentam a noção de “arquivamento *versus* mudança prevista”. Não há dúvida de que a sua contribuição realmente definiu o cenário para os Métodos de Região de Confiança na forma que é conhecida atualmente. O artigo não cita contribuições anteriores de Levenberg, Morrison, Maquardt, Griffith e Stewart. O conceito de região de confiança foi posteriormente proposto por Powell (1970a) como uma ferramenta para assegurar a convergência de um método de otimização irrestrita, em que a fórmula Quasi-Newton de Powell-Symmetric-Broyden (PSB) foram usadas para construir a Hessiana de um problema quadrático. Powel refere-se ao trabalho de Maquardt como a inspiração para sua proposta. O trabalho inédito de Fletcher (1970) segue a mesma linha. Powel (1970b) também propôs a utilização de um dispositivo semelhante, no contexto de resolução de equações não-lineares utilizando uma atualização quasi-Newton de Broyden. Nestas contribuições, o procedimento de atualização é claramente focado no raio da região de confiança.

De 1970 em diante, o assunto foi tomado ativamente pelos pesquisadores, resultando em um aumento de resultados e ideias.

O termo região de confiança não foi mencionado por Powell (1970b), em seu trabalho pioneiro, nem por qualquer um dos seus antecessores. O termo de região de confiança e ponto de Cauchy foram citados pela primeira vez por Dennis, em um curso que ele ministrou em *U. S. National Bureau of Economic Research*, logo depois que ele ouviu Powell falar sobre sua técnica para resolver equações não-lineares. A primeira aparição oficial do termo região de confiança e ponto de Cauchy foi em Cellis et al. (1984). No entanto, esta terminologia não foi divulgada pelos pesquisadores por vários anos. Por exemplo, Winfield (1973) fala de uma região de validade por um problema quadrático, Fletcher (1980) utiliza o termo método de passo restrito para indicar que o passo é restrito a uma determinada região centrada na

iteração atual, e Toint (1988) refere-se a uma região de confiança. Moré (1983), em seu artigo, foi que influenciou a padronização da terminologia.

Destacamos vários outros autores que apresentaram prova de convergência global do método, artigos teóricos e com algoritmos, como pode ser visto em Conn et al. (2000).

Byrd, Gilbert e Nocedal (2000) que utilizaram um Método Região de Confiança baseado em técnicas de pontos interiores para programação não linear em que aplicaram técnicas de programação quadrática sequencial para uma sequência de problemas de barreira, e utilizaram o Método de Região de Confiança para assegurar a robustez da iteração e para permitir a utilização direta de derivadas de segunda ordem.

Nos Métodos de Região de Confiança, diferentemente dos Métodos de Busca Linear, para o cálculo da solução do problema (1), é determinado somente o vetor passo ao invés do cálculo da direção e do comprimento do passo, dessa forma:

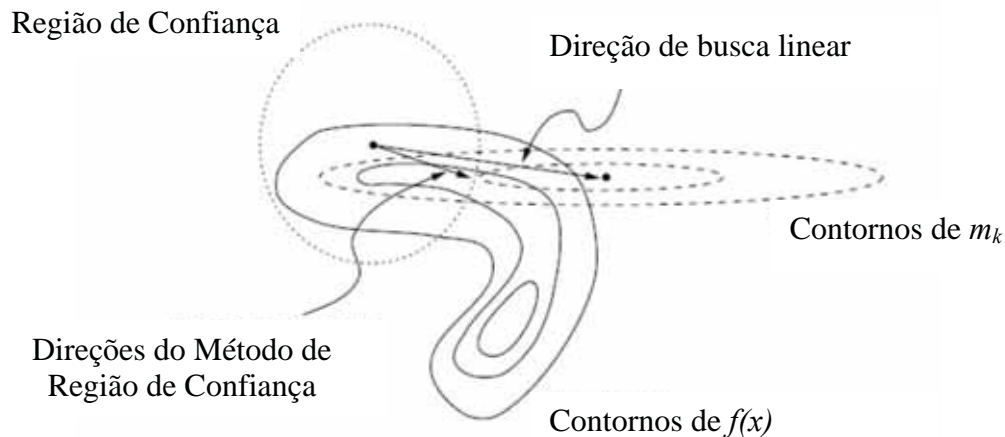
$$x^{k+1} = x^k + p, \quad (25)$$

em que  $p \in R^n$  é denominado vetor passo.

De acordo com Nocedal e Wright (1999), o Método de Região de Confiança define uma região ao redor da iteração atual em que a minimização de um problema quadrático seja uma boa aproximação desta região de confiança. Especificamente, é escolhido o tamanho da região e o comprimento de passo simultaneamente, como mostra a Figura 1. Se o vetor passo não é aceitável, então é reduzido o tamanho da região de confiança e tenta-se uma nova minimização. Em geral, o vetor passo muda sempre que o tamanho da região de confiança é alterado. O tamanho da região de confiança é muito importante e essencial para a efetividade de cada cálculo. Se a região de confiança é muito pequena, o algoritmo perde uma oportunidade de ter um vetor passo significativo que se mova perto do minimizador da função objetivo. Se a região é muito grande o vetor passo pode estar muito longe da região do minimizador da função objetivo, então tem-se que reduzir o tamanho da região de confiança e repetir o processo.

Na prática, os algoritmos escolhem o tamanho da região de confiança de acordo com o desenvolvimento do algoritmo durante uma iteração prévia. Se o

modelo está reagindo bem, ou seja, produzindo bons vetores passos e tendo-se uma previsão do comportamento da função objetivo ao longo desses vetores passos, o tamanho da região de confiança continua decrescente ao longo dos vetores passos que são feitos. Por outro lado, o fracasso de um vetor passo indica que o problema não é uma boa representação da função objetivo da região de confiança atual, então é reduzido o tamanho da região e uma nova tentativa é feita.



**Figura 1** - Direção do Método de Região de Confiança.  
**Fonte:** Nocedal e Wright (1999).

Considere o problema (1). Os Métodos de Região de Confiança geram uma sequência  $\{x_k\}_{k \in \mathbb{N}}$  com a ajuda da função quadrática dado por:

$$m_k(p) = f^k + \nabla(f^k)^T p + \frac{1}{2} p^T B_k p \quad (26)$$

em que  $f^k = f(x^k)$ ,  $\nabla f^k = \nabla f(x^k)$  e  $B_k$  é uma matriz simétrica.

Quando  $B_k$  é a matriz Hessiana  $\nabla^2 f^k$ , a função concorda com os três primeiros termos da expansão em série de Taylor de  $f$ . O algoritmo obtido fazendo  $B_k = \nabla^2 f^k$  é denominado de Método de Newton para região de confiança.

Para obter cada passo, procuramos a solução do subproblema:

$$\begin{aligned} \min_{p \in \mathbb{R}^n} \quad & m_k(p) = f^k + \nabla(f^k)^T p + \frac{1}{2} p^T B_k p \\ \text{s.a:} \quad & \|p\| \leq \Delta^k \end{aligned} \quad (27)$$

em que  $\Delta^k > 0$  é o raio da região de confiança. A solução  $(p^k)^*$  de (27) é o minimizador de (26) na região de confiança de raio  $\Delta^k$ . Portanto, precisamos resolver uma sequência de subproblemas (27) nos quais a função objetivo e a restrição, o qual pode ser reescrito por,  $p^T p \leq (\Delta^k)^2$  são quadráticas. Quando  $B_k$  é definida positiva e  $\|B_k^{-1} \nabla f_k\| \leq \Delta^k$ , a solução de (27) é simplesmente o minimizador irrestrito  $(p^k)^N = B_k^{-1} \nabla f_k$  do problema quadrático  $m_k(p)$ . Neste caso chamamos  $(p^k)^N$  de vetor passo inteiro. A solução de (27) não é tão óbvia em outros casos, mas pode ser encontrada sem muito esforço. De qualquer maneira, precisamos apenas de uma solução aproximada para obter convergência e um bom comportamento prático.

O raio da região de confiança  $\Delta^k$  é atualizado a cada iteração. A escolha deste é baseada na concordância entre a função  $m_k$ , apresentada em (26), e a função  $f$  nas iterações anteriores.

Dada um vetor passo  $p^k$ , definimos a razão:

$$\rho_k = \frac{f(x^k) - f(x^k + p^k)}{m_k(0) - m_k(p^k)} \quad (28)$$

em que o denominador é chamado de redução prevista (*pred*) pelo modelo, ou seja,

$$pred = m_k(0) - m_k(p^k) \quad (29)$$

e o numerador é chamado de redução atual (*ared*) da função  $f$ , isto é,

$$ared = f(x^k) - f(x^k + p^k). \quad (30)$$

Note que, como o vetor passo  $p^k$  é obtido minimizando o modelo (27) em uma região que inclui o vetor passo  $p = 0$ , a redução prevista será sempre não negativa. Desta forma, podemos concluir que: se  $\rho_k$  é negativo, o novo valor  $f(x^k + p^k)$  é maior do que o valor corrente  $f(x^k)$ , e então o vetor passo deve ser rejeitado; se  $\rho_k$  é próximo de 1, existe uma boa concordância entre o modelo  $m_k$  e a função  $f$ , sendo assim, é seguro expandir a região de confiança na próxima iteração; se  $\rho_k$  é positivo, mas não está próximo de 1, não alteramos a região de confiança e se  $\rho_k$  está próximo de zero ou negativo, a região de confiança é diminuída.

O Algoritmo 5 descreve o processo:

---

**Algoritmo 5:** Método de Região de Confiança

---

**Início**

Escolha valores iniciais para:  $\bar{\Delta} > 0$ ,  $\Delta^k \in (0, \bar{\Delta})$  e  $\eta \in \left[0, \frac{1}{4}\right)$

Defina  $k = 0$ ;

Obter a solução aproximada  $p_k$  da solução de (27);

Resolver (28);

**Se**  $\rho_k < \frac{1}{4}$

$$\Delta^{k+1} = \frac{1}{4} \|p^k\|$$

**Senão**

**Se**  $\rho_k > \frac{3}{4}$  e  $\|p^k\| = \Delta^k$

$$\Delta^{k+1} = \min(2\Delta^k, \bar{\Delta})$$

**Senão**

$$\Delta^{k+1} = \Delta^k ;$$

**Fim**

**Fim**

**Se**  $\rho_k > \eta$

$$x^{k+1} = x^k + p^k$$

**Senão**

$$x^{k+1} = x^k$$

**Fim**

$k = k + 1$

**Fim**

---

Note que  $\bar{\Delta}$  é uma cota global para os vetores passos e que o raio é aumentado somente se  $\|p_k\|$  alcança a fronteira da região de confiança. Se o vetor passo fica estritamente dentro da região de confiança  $\Delta^k$ , concluímos que o valor atual de  $\Delta^k$  não está interferindo no progresso do algoritmo e deixamos seu valor igual para a próxima iteração.

Vamos descrever duas estratégias para encontrar soluções aproximadas para o problema (27), que reduzem  $m_k$ , realizada pela abordagem chamada de ponto de Cauchy. Este ponto é simplesmente o minimizador de  $m_k$  ao longo da direção do vetor gradiente  $\nabla f^k$ , sujeito a fronteira da região de confiança.

A primeira estratégia é o Método *Dogleg* e seu uso é apropriado quando a matriz  $B_k$  é uma aproximação da matriz Hessiana e esta é definida positiva. A segunda estratégia, conhecida como Método CG de Steihaug, é mais indicada quando a matriz  $B_k$  é a matriz Hessiana exata,  $\nabla^2 f^k$ , e quando esta matriz é de grande dimensão e esparsa.

Para introduzirmos essa estratégia, apresentamos a definição do ponto de Cauchy.

### 2.4.1 Ponto de Cauchy

No Método de Região de Confiança, para se obter convergência global, é suficiente encontrar uma solução aproximada  $p^k$ , que se encontra dentro da região de confiança e que forneça uma redução suficiente na função objetivo do problema. Esta redução suficiente, que denotaremos como  $(p^k)^C$ , pode ser quantificada em termos do ponto de Cauchy.

Seja  $(p^k)^S$  o vetor que resolve a versão linear de (27), isto é,

$$\begin{aligned}
(p^k)^S &= \min_{p \in \mathbb{R}^n} f^k + \nabla(f^k)^T p \\
\text{s.a:} \quad &\Delta^k \geq \|p\|
\end{aligned} \tag{31}$$

e seja  $\tau > 0$  o escalar que minimiza  $m_k(\tau(p^k)^S)$  satisfazendo a fronteira da região de confiança, ou seja,

$$\begin{aligned}
\tau_k &= \underset{\tau > 0}{\text{minimizar}} \quad m_k(\tau(p^k)^S) \\
\text{sujeito a:} \quad &\|\tau(p^k)^S\| \leq \Delta^k
\end{aligned} \tag{32}$$

Logo,  $(p^k)^C = \tau_k(p^k)^S$

A solução de (31) é:

$$(p^k)^S = -\frac{\Delta^k}{\|\nabla f^k\|} \nabla f^k. \tag{33}$$

Para encontrar  $\tau_k$  explicitamente, são considerados separadamente os casos  $\nabla(f^k)^T B_k \nabla f^k \leq 0$  e  $\nabla(f^k)^T B_k \nabla f^k > 0$ . No primeiro caso, conforme  $\tau$  cresce, a função  $m_k(\tau(p^k)^S)$  decresce monotonicamente, sempre que  $\nabla f^k \neq 0$ , então  $\tau_k$  é simplesmente o maior valor que satisfaz a região de confiança, ou seja,  $\tau_k = 1$ . Para o segundo caso,  $m_k(\tau(p^k)^S)$  é uma função quadrática e convexa em  $\tau$ , então  $\tau_k$  é o

minimizador sem restrição da função quadrática  $\left( \frac{\|\nabla f^k\|^3}{(\Delta^k \nabla(f^k)^T B_k \nabla f^k)} \right)$ , ou o valor

de fronteira 1, ou seja, o mínimo entre esses dois valores. Resumindo, temos

$(p^k)^C = -\tau_k \frac{\Delta^k}{\|\nabla f^k\|} \nabla f^k$ , em que:

$$\tau_k = \begin{cases} 1 & , \text{ se } \nabla(f^k)^T B_k \nabla f^k \leq 0; \\ \min \left( \frac{\|\nabla f^k\|^3}{(\nabla(f^k)^T B_k \nabla f^k)}, 1 \right) & , \text{ caso contrário.} \end{cases} \quad (34)$$

O cálculo do passo de Cauchy,  $(p^k)^C$ , é de baixo custo computacional, pois nenhuma fatoração de matriz é necessária e é extremamente importante para decidir se uma solução aproximada do subproblema de região de confiança é aceitável. Especificamente, um Método de Região de Confiança será globalmente convergente se seus vetores passos obtiverem uma redução suficiente no problema  $m_k$ , ou seja, se a redução no problema  $m_k$  é pelo menos algum múltiplo fixo do decréscimo obtido pelo passo de Cauchy a cada iteração.

Note que se tomarmos o ponto de Cauchy como nosso passo, sempre estaremos implementando o Método do Gradiente com uma escolha particular do tamanho do passo. Mas este método não tem um bom desempenho mesmo se os passos ótimos foram usados a cada iteração. Por esta razão devemos aperfeiçoar o ponto de Cauchy para obter uma solução aproximada melhor do subproblema.

O ponto de Cauchy não depende fortemente da matriz  $B_k$ , que é usada apenas no cálculo do vetor passo. A convergência rápida pode ser esperada apenas se  $B_k$  tem a função de determinar tanto a direção quanto tamanho do passo.

Vários algoritmos que geram soluções aproximadas  $p^k$  para o subproblema (27) iniciam calculando o ponto de Cauchy e tentam melhorá-lo. A estratégia de aperfeiçoamento é frequentemente projetada para que o vetor passo inteiro  $(p^k)^N = -B_k^{-1} \nabla f^k$  sendo que  $B_k$  é sempre escolhida definida positiva e  $\|(p^k)^N\| \leq \Delta^k$ .

A seguir, iremos descrever algoritmos utilizados para o cálculo do ponto de Cauchy, o Método *Dogleg* e o Método do Gradiente Conjugado de Steihaug.

### 2.4.1.1 O Método *Dogleg*

O Método *Dogleg* requer a solução de um sistema linear envolvendo a matriz  $B$  a cada iteração. Se  $B$  é de grande dimensão, esta operação pode ter um alto custo computacional.

Sabemos que a solução do subproblema (27) depende do raio da região de confiança  $\Delta^k$ . Quando  $B_k$  é uma matriz definida positiva, o minimizador irrestrito do problema  $m_k$  é o vetor passo inteiro  $(p^k)^N = -B_k^{-1}\nabla f^k$ . Se este ponto é factível, ele será uma solução. Assim, a solução  $p^*(\Delta)$  é dada por  $p^*(\Delta) = p^*$ , quando  $\|p^N\| \leq \Delta$ .

Quando  $\Delta^k$  é pequeno, a restrição  $\|p^k\| \leq \Delta^k$  assegura que o termo quadrático no problema  $m_k$  tem pouco efeito na solução de (28). A solução verdadeira  $p(\Delta^k)$  é aproximadamente a mesma solução que obteríamos minimizando a função linear  $f^k + \nabla(f^k)^T p_k$  sobre  $\|p^k\| \leq \Delta$ , isto é,  $p^*(\Delta) \approx -\Delta \frac{\nabla f^k}{\|\nabla f^k\|}$ , quando

$\Delta$  é pequeno.

O Método *Dogleg* encontra uma solução aproximada substituindo a trajetória curva por um caminho constituído por dois segmentos. O primeiro segmento vai da origem ao minimizador irrestrito sobre a direção de descida máxima definida por  $p^C = -\frac{\nabla(f^k)^T \nabla f^k}{\nabla(f^k)^T B_k \nabla f^k} \nabla f^k$ , enquanto o segundo segmento vai de  $p^C$  até  $p^N$ .

Formalmente, denotamos esta trajetória por  $\tilde{p}(\tau)$  para  $\tau \in [0, 2]$  em que:

$$\tilde{p}(\tau) = \begin{cases} \tau p^C & , 0 \leq \tau \leq 1, \\ p^C + (\tau - 1)(p^N - p^C) & , 1 \leq \tau \leq 2. \end{cases} \quad (35)$$

O Método *Dogleg* escolhe  $p$  que minimiza o problema  $m$  ao longo do caminho, sujeito à região de confiança. De fato, não é necessário nem realizar uma

busca, pois o caminho *dogleg* intercepta a fronteira da região de confiança no máximo uma vez, e o ponto de intersecção pode ser calculado analiticamente.

As provas referentes a essa teoria podem ser encontradas em Nocedal e Wright (1999).

### 2.4.1.2 O Método do Gradiente Conjugado de Steihaug

O Método do Gradiente Conjugado de Steihaug é uma técnica baseada no algoritmo do Método do Gradiente Conjugado, um algoritmo iterativo para resolver sistemas lineares cujas as matrizes simétricas simétricas e definidas positivas. O algoritmo do Método do Gradiente Conjugado (CG) está descrito no capítulo 2 deste trabalho.

Destacamos, nesta seção, que a diferença entre o método CG padrão em relação a abordagem de Steihaug, é que o algoritmo termina quando sai da região de confiança, definida por  $\|p\| \leq \Delta$ , ou quando encontra uma direção de curvatura negativa em  $B$ .

O Método do Gradiente Conjugado de Steihaug pode ser formalmente apresentado como segue:

---

#### Algoritmo 6: Método do Gradiente Conjugado de Steihaug

---

##### Início

Defina  $\varepsilon > 0$ ,

$$p^k = 0,$$

$$r^k = g,$$

$$g = \nabla f^k,$$

$$d^k = -r^k$$

Defina  $k = 0$ ;

**Se**  $\|r^k\| < \varepsilon$

Retorne  $p = p^k$

**Para**  $k = 1, 2, \dots, n$ ;

**Se**  $(d^k)^T B d^k \leq 0$

Encontre  $\tau$  tal que  $p = p^k + \tau d^k$  minimiza  $m(p)$  e satisfaz  $\|p\| = \Delta$ ;

Retorna  $p$

Defina  $\alpha_k = \frac{(r^k)^T r^k}{(d^k)^T B d^k}$

Defina  $p^{k+1} = p^k + \alpha_k d^k$

**Se**  $\|p^{k+1}\| \geq \Delta$

Determine  $\tau \geq 0$  tal que  $p = p^k + \tau d^k$  satisfaz  $\|p\| = \Delta$ ;

Retorna  $p$ ;

Defina  $r^{k+1} = r^k + \alpha_k B d^k$

**Se**  $\|r^{k+1}\| < \varepsilon \|r^k\|$

Retorna  $p = p^{k+1}$

Defina  $\beta^{k+1} = \frac{(r^{k+1})^T r^{k+1}}{(r^k)^T r^k}$

Defina  $d^{k+1} = -r + \beta^{k+1} d^k$

$k = k + 1$

**Fim**

**Fim**

---

O Algoritmo 6 difere do algoritmo do Método CG, o qual pode ser encontrado em Nocedal e Wright (1999), pela presença dos dois primeiros comandos **Se** dentro do comando **Para**, sendo estes os critérios de parada para o algoritmo. O primeiro comando **Se** finaliza o algoritmo se sua direção de busca atual  $d^j$  for uma direção de curvatura nula ou curvatura negativa ao longo de  $B$ . O segundo comando **Se** finaliza o algoritmo se  $p^j$  violar o limite da região de confiança. Em ambos os casos,  $p$  é encontrado pela intersecção da direção de busca atual com o limite da região de confiança.

## 2.5 *Knitro*: Pacote de algoritmos de otimização

O pacote *Knitro* foi desenvolvido em sua primeira versão 1.0, inicialmente não comercializada, em julho de 2001, por Nocedal e Waltz (2001), no qual foi implementado um método de pontos interiores juntamente com um algoritmo do gradiente conjugado.

Em 2002, foi lançada a segunda versão, já comercializada, do pacote de algoritmos *Knitro* 2.0 pelos autores Waltz e Nocedal (2002), o qual utilizava o Método do Gradiente Conjugado para o cálculo do passo e uma versão dos algoritmos *Interior-Point* e *Direct* para resolução de problemas de grande porte. Nesta versão, a linguagem de programação utilizada foi o Fortran com suporte para os sistemas operacionais *Linux*, *Solaris* e *Windows*.

A versão 6.0 foi lançada pelos autores Waltz e Platenga (2009) e apresenta diferenças com relação as versões anteriores, como: novos recursos para resolver modelos de otimização com ou sem variáveis binárias e inteiras para problemas de programação linear e não linear; dois algoritmos para a programação inteira-mista não-linear; e oferece um melhor desempenho da Programação Sequencial Quadrática Linear no algoritmo *Active Set* em problemas simples de otimização com restrições, entre outras.

No *Knitro* estão disponíveis seis estratégias de atualizações do parâmetro de barreira, a saber: (i) o parâmetro de barreira é decrescido monotonicamente; (ii) o parâmetro de barreira é calculado por uma regra adaptativa com base na diferença de complementaridade; (iii) é realizada uma investigação sobre o passo (afim-escala) para determinar dinamicamente o valor do parâmetro barreira a cada iteração; (iv) é utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira levando em conta o passo corretor; (v) é utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira em que o passo corretor é condicionado; e (vi) uma função qualidade é minimizada em cada iteração para determinar o parâmetro de barreira. A investigação destas estratégias é o objetivo principal de estudo do nosso trabalho.

Ao longo dos anos, o pacote de algoritmos de otimização foi evoluindo ao chegar a sua versão *Knitro* 7.0, que foi criada em setembro de 2010 por Waltz e Platenga (2010), a qual apresenta diversas melhorias em relação à álgebra linear. Nele

encontra-se implementado um Método de Pontos Interiores, que utiliza para o cálculo do passo uma busca linear ou um passo de região de confiança. A linguagem de programação utilizada é C/C++ e também possui suporte para os sistemas operacionais *Linux*, *Solaris* e *Windows*.

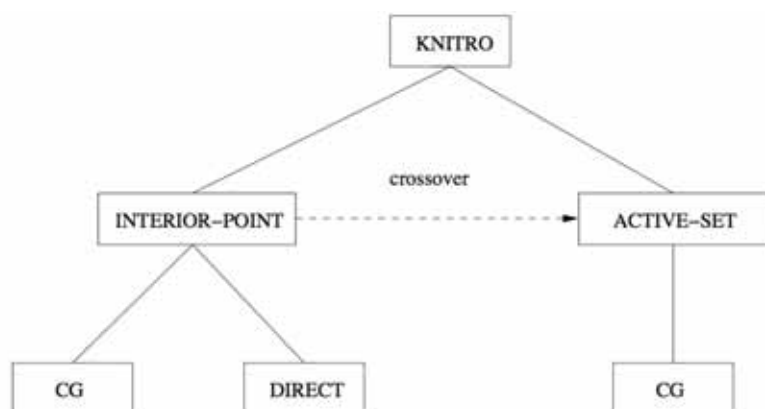
Sua atual versão é a *Knitro* 8.1, atualizada em dezembro de 2012 e comercializada por *Ziena Optimization LLC* (<http://www.ziena.com/index.htm>). Nesta versão houveram algumas modificações com a adição de novos recursos, os quais podemos citar: uma nova linguagem de programação, *Python*, utilizando a plataforma *Linux* ou *Unix*; capacidade de executar a subrotina *Basic Linear Algebra Subroutines* (BLAS) e paralelamente usar a biblioteca *Intel Math Kernel Library* BLAS fornecidos com *Knitro*, proporcionando acelerações significativas em problemas de grande porte; novas funções utilizando a biblioteca API para retornar ao usuário o gradiente final, Jacobiana e Hessiana, caso exista; a nova função da biblioteca API "*KTR\_chgvarbnds*", que permite aos usuários modificar os limites de variáveis no modelo após uma primeira solução e resolver o problema de otimização novamente; a possibilidade de utilizar o *solver* FICOXpress(R) juntamente com o algoritmo *Active-set* do *Knitro*; mais velocidade e melhorias no desempenho ([http://www.ziena.com/docs/Knitro81\\_UserManual.pdf](http://www.ziena.com/docs/Knitro81_UserManual.pdf)).

O pacote *Knitro* oferece suporte para *Windows* (32 bits e 64 bits), *Linux* (32-bit e 64-bit), *Mac OS X* (32-bit e 64-bit) e *Solaris* 10 (64 bits, 64 x 86) utilizando linguagem de programação C/C++, *Fortran*, *Java* ou *Microsoft Excel* e utilizando os ambientes *AMPL*<sup>®</sup>, *AIMMS*<sup>®</sup>, *GAMS*<sup>®</sup>, *Mathematica*<sup>®</sup> e *MATLAB*<sup>®</sup> para ser executado.

Atualmente, o *Knitro* é um pacote de algoritmos de otimização composto por três algoritmos diferentes para resolver problemas de otimização. Dois algoritmos são do tipo ponto interior (*Interior-Direct*) e um é do tipo conjunto ativo (*Active Set*). Estes algoritmos são conhecidos por terem características diferentes, por exemplo, os Métodos de Pontos Interiores seguem uma direção no interior da região factível enquanto os Métodos de Conjunto Ativo tendem a permanecer nas fronteiras. O pacote *Knitro* disponibiliza os algoritmos para ter uma maior flexibilidade na resolução de vários tipos de problemas e permite um cruzamento (*crossover*) durante o processo de solução de um algoritmo para outro (Waltz e Platenga, 2010).

O *Knitro* pode aplicar um Método de Ponto Interior (*Interior-Point*), o qual é utilizado neste trabalho, ou um Método do Conjunto Ativo (*Active Set*), ou

possivelmente ambos, dependendo das características do problema a ser resolvido. Dentro da abordagem *Interior-Point*, o *Knitro* fornece dois algoritmos implementando abordagens distintas de barreira, sendo elas *CG* e *Direct*, como podemos observar na Figura 2. A nomenclatura *CG* reflete o fato de que o passo algorítmico é calculado usando um Método de Região de Confiança juntamente com uma abordagem iterativa do Método Gradiente Conjugado, enquanto que a *Interior-Direct* implica que o passo é calculado usando uma busca linear por meio de uma fatoração direta de um sistema linear. O pacote oferece também a seleção automática de opções e configurações do mesmo (Byrd et al., 2006), que por padrão, o pacote *Knitro* tenta escolher o melhor algoritmo para o problema com base em características do mesmo (Waltz e Platenga, 2009).



**Figura 2** - Uma visão esquemática do pacote *Knitro*  
**Fonte:** Byrd et al. (2006)

De acordo com Byrd et al. (2006), um dos principais desafios no desenvolvimento do *Knitro* tem sido a integração eficaz dos algoritmos de pontos interiores e do conjunto ativo em um único pacote e o desenvolvimento de ferramentas que exploram o poder da abordagem integrada.

A seguir, descrevemos resumidamente cada um dos algoritmos que compõem o pacote de algoritmos *Knitro*.

### 2.5.1 Algoritmo *Interior-Direct*

O algoritmo *Interior-Direct* utiliza um Método de Ponto Interior, no qual o passo é calculado usando um procedimento de busca linear. Com o objetivo de obter a convergência global na presença de não convexidade e de singularidades da Hessiana ou Jacobiana, o primeiro passo pode ser substituído por um passo de região de confiança utilizando o algoritmo *Interior-CG* (Byrd et al., 2006). Desta forma, um ou mais passos são realizados no *Interior-CG* até que haja uma redução suficiente em uma função mérito proposta. A implementação do algoritmo *Interior-Direct* é descrito no capítulo 3 e é objetivo de investigação e aplicação deste trabalho.

O uso do algoritmo *Interior-Direct* pode resultar em melhorias substanciais em relação ao *Interior-CG* quando o problema é mal-condicionado, devido ao fato deste, poder realiza um número grande de iterações do Método do Gradiente Conjugado.

### **2.5.2 Algoritmo *Interior-CG***

Byrd et al. (2006) afirmam que o algoritmo *Interior-CG* implementado no *Knitro*, calcula passos utilizando um problema quadrático, regiões de confiança e o Método do Gradiente Conjugado Projetado.

Esta abordagem difere-se da maioria dos Métodos de Pontos Interiores propostos na literatura, pois o cálculo da direção não é feito através de um sistema linear envolvendo KKT e a matriz Hessiana. Ao invés disso, utiliza-se uma matriz de projeção e o Método do Gradiente Conjugado, para minimizar um problema quadrático do problema barreira.

Esta formulação permite maior liberdade na escolha da matriz Hessiana e fornece mecanismos para lidar com singularidades da Hessiana e da Jacobiana, porém sua iteração é mais complexa do que a abordagem de busca linear. O Método *Interior-CG* é descrito no capítulo 3, como parte do *Interior-Direct*.

Segundo o autor, o problema de barreira (29) pode ser resolvido usando um Método de Programação Quadrática Sequencial com regiões de confiança. A aplicação direta de técnicas de programação sequencial quadrática para o problema de barreira leva a passos ineficientes que tendem a violar a positividade das variáveis de folga, os quais são interrompidos pela restrição da região de confiança. Para resolver

este problema, é elaborado um Método de Programação Sequencial Quadrática voltado ao problema de barreira.

O Método *Interior-CG* é equivalente ao Método de Ponto Interior com Busca Linear (padrão), mas é significativamente diferente em dois aspectos. Em primeiro lugar, não é totalmente um Método Primal-Dual, em que os multiplicadores são calculados como uma função das variáveis primais ( $x,s$ ), que se opõe à formulação (41) em que as variáveis primais e duais são calculadas simultaneamente de seus valores anteriores. Em segundo lugar, o Método de Região de Confiança utiliza uma escala das variáveis que desencoraja a intersecção da fronteira da região factível. Isso faz com que o algoritmo gere passos que podem ser diferentes dos produzidos pelo Método de Busca Linear (Byrd et al., 2006).

### 2.5.3 Algoritmo *Active-Set*

No *Knitro* também é implementado um algoritmo de conjunto ativo para programação sequencial linear e quadrática, o qual é denominado de *Active-Set*. Este método é de natureza similar a um Método de Programação Sequencial Quadrática, mas usa subproblemas de programação linear para estimar o conjunto ativo em cada iteração.

Segundo Waltz e Platenga (2010), este algoritmo é fundamentalmente diferente dos Métodos de Pontos Interiores. O método é eficiente e robusto para problemas pequenos e de médio porte, mas tipicamente menos eficiente do que os algoritmos *Interior-Direct* e *Interior-CG* em relação a problemas de grande porte, o qual tem-se centenas de variáveis e restrições. O *Active-Set* é recomendado quando o usuário pode fornecer uma boa estimativa da solução inicial.

De acordo com Byrd et al. (2006), o algoritmo *Active-Set*, não segue uma abordagem de programação sequencial quadrática, porque o custo para resolver subproblemas de programação quadrática e restrita impõe uma limitação no tamanho dos problemas que podem ser resolvidos na prática. Além disso, a incorporação de informações da segunda derivada em Métodos de Programação Sequencial Quadrática provou ser mais complexa.

É usado então, um Método de Programação Sequencial Linear e Quadrática que calcula um passo em duas etapas. Primeiro, um problema de programação linear é

resolvido para identificar um conjunto ativo. Segundo, isto é seguido por uma fase que explora programação quadrática com restrições de igualdade, em que as restrições no conjunto ativo são impostas como igualdades. O passo total do algoritmo é uma combinação dos passos obtidos na programação linear e na fase de programação quadrática com restrições de igualdade.

#### **2.5.4 A opção *crossover***

Os Métodos de Pontos Interiores proporcionam apenas uma estimativa aproximada da solução e o ótimo do conjunto ativo. Em muitas aplicações práticas é importante saber com precisão quais as restrições são ativas, pois estas correspondem à presença ou atividade de certos componentes da solução. Além disso, muitas vezes é importante ter estimativas precisas dos multiplicadores de Lagrange. Isto pode ser feito através da mudança do Método *Interior-Point* para uma iteração do Método *Active-Set*. Este processo é frequentemente chamado de cruzamento (*crossover*). Embora as técnicas de cruzamento tenham recebido muita atenção no contexto de programação linear, citado por Megiddo (1989), é possível observar que nenhum dos códigos de Métodos de Pontos Interiores não-lineares oferecem uma opção para isso. Esse processo de cruzamento é essencial para a eficiência computacional e determinação de soluções, com o objetivo de ser útil em futuras aplicações (Byrd et al., 2006).

Segundo Byrd et al. (2006), na programação linear, a opção *crossover* envolve duas etapas: a primeira etapa é a identificação de restrições ativas e a segunda é a mudança de uma solução não-básica para uma próxima solução básica. Na programação não linear, não se pode esperar que o conjunto das restrições ativas corresponda a uma solução básica. Ao invés disso, o procedimento de *crossover* procura identificar um conjunto de restrições ativas com os gradientes das restrições linearmente independentes e calcula uma solução em que essas restrições são satisfeitas na igualdade e que satisfazem a estacionaridade da Lagrangiana usando essas mesmas restrições.

No próximo capítulo abordaremos o método proposto por Waltz et al. (2004), um Método de Ponto Interior com busca linear e região de confiança, o qual está implementado no pacote de algoritmos de otimização *Knitro*.

## CAPÍTULO 3

# MÉTODO DE PONTO INTERIOR COM BUSCA LINEAR E REGIÃO DE CONFIANÇA

Neste capítulo descrevemos o Método de Pontos Interiores com Busca Linear e Região de Confiança proposto em Waltz et al. (2004). Este método está implementado no pacote de algoritmos *Knitro* e será utilizado nos testes do capítulo 6.

### 3.1 Definição do problema

Considere um problema de otimização restrito dado por:

$$\begin{array}{ll} \text{minimizar} & f(x) \\ \text{sujeito a:} & h(x) = 0 \\ & g(x) \leq 0 \end{array}$$

(36)

em que:  $x \in R^n$ ,  $f: R^n \rightarrow R$ ,  $h: R^n \rightarrow R^l$  e  $g: R^n \rightarrow R^m$ , e  $f$ ,  $g$  e  $h$  são de classe  $C^2$ .

### 3.2 O método

Para resolver o problema (36), Waltz et al. (2004) propõe resolver uma sequência de problemas de barreira dados por:

$$\begin{aligned}
& \underset{z}{\text{minimizar}} \quad \varphi_\mu(z) \equiv f(x) - \mu \sum_{i=1}^m \ln s_i \\
& \text{sujeito a:} \quad h(x) = 0 \\
& \quad \quad \quad g(x) + s = 0
\end{aligned} \tag{37}$$

em que  $s > 0$  é um vetor de variáveis de folga,  $z = (x, s)^T$  e  $\mu > 0$  é o parâmetro de barreira.

Ao problema (37) é associada a seguinte função Lagrangiana:

$$L(z, \lambda, \mu) = \varphi_\mu(z) + \lambda_h^T h(x) + \lambda_g^T (g(x) + s) \tag{38}$$

em que  $\lambda_h \in R^l$  e  $\lambda_g \in R^m$  são os vetores dos multiplicadores de Lagrange e definimos  $\lambda = (\lambda_h, \lambda_g)^T$ .

O Método de Pontos Interiores consiste em encontrar uma solução aproximada para o problema (37), com um parâmetro de barreira  $\mu = \mu_k$ , esse parâmetro é atualizado (decrecido) e obtém-se  $\mu_{k+1}$ , e então o problema de barreira (37) com  $\mu = \mu_{k+1}$  é resolvido novamente.

A sequência de parâmetros  $\mu_k$ , gera uma sequência de problemas (37) e uma sequência de soluções desses problemas  $z^*(\mu_k)$ . Quando  $\mu_k \rightarrow 0$ , então  $z^*(\mu_k) \rightarrow z^*$ , em que  $z^*$  é um ótimo local do problema (36). No capítulo 4 são apresentadas seis estratégias de atualização para o parâmetro de barreira as quais serão analisadas neste trabalho.

### 3.3 A estratégia de busca linear

Para determinar a solução do problema (36), Waltz et al. (2004), aplicam as condições necessárias de primeira ordem na função Lagrangiana (38) o que resulta no seguinte sistema não linear:

$$\begin{bmatrix} \nabla f(x) + A_h(x)^T \lambda_h + A_g(x)^T \lambda_g \\ S \Lambda_g e - \mu e \\ h(x) \\ g(x) + s \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (39)$$

em que  $S$  e  $\Lambda_g$  são matrizes diagonais, cujas diagonais são dadas, respectivamente, pelos vetores  $s$  e  $\lambda_g$ ;  $A_h$  e  $A_g$  são as matrizes Jacobianas associadas às restrições  $h$  e  $g$ .

Resolvendo (39) pelo Método de Newton, obtemos o seguinte sistema:

$$\begin{bmatrix} W(z, \lambda; \mu) & A(x)^T \\ A(x) & 0 \end{bmatrix} \begin{bmatrix} d_z \\ d_\lambda \end{bmatrix} = - \begin{bmatrix} \nabla_z L(z, \lambda; \mu) \\ c(z) \end{bmatrix} \quad (40)$$

em que,

$d_z = \begin{bmatrix} d_x \\ d_s \end{bmatrix}$  e  $d_\lambda = \begin{bmatrix} d_h \\ d_g \end{bmatrix}$ ; são denominados direções primais e duais, respectivamente,

$$c(z) = \begin{bmatrix} h(x) \\ g(x) + s \end{bmatrix}, \quad A(x) = \begin{bmatrix} A_h(x) & 0 \\ A_g(x) & I \end{bmatrix},$$

$$\text{e } W(z, \lambda; \mu) = \begin{bmatrix} \nabla_{xx}^2 L(z, \lambda; \mu) & 0 \\ 0 & S^{-1} \Lambda_g \end{bmatrix}.$$

A nova solução é obtida através de:

$$z^+ = z + \alpha_z d_z, \quad \lambda^+ = \lambda + \alpha_\lambda d_\lambda; \quad (41)$$

em que  $\alpha_z$  e  $\alpha_\lambda$  são os tamanhos de passos associados ao vetor das variáveis  $z$  e  $\lambda$ , respectivamente.

Para o cálculo do tamanho dos passos  $\alpha_z$  e  $\alpha_\lambda$  em (41), são definidos:

$$\begin{aligned}\alpha_z^{\max} &= \max\{\alpha \in (0,1] : s + \alpha d_s \geq (1-\tau)s\} \\ \alpha_\lambda^{\max} &= \max\{\alpha \in (0,1] : \lambda_g + \alpha d_g \geq (1-\tau)\lambda_g\}\end{aligned}\quad (42)$$

com  $0 < \tau < 1$ , em particular no trabalho é adotado  $\tau = 0,995$  e o algoritmo realiza uma busca linear *backtraking* que calcula o tamanho do passo  $\alpha_z \in (0, \alpha_z^{\max}]$  e  $\alpha_\lambda \in (0, \alpha_\lambda^{\max}]$ , de modo a reduzir a função mérito a qual é definida por:

$$\phi_\nu(z) = \phi_\mu(z) + \nu \|c(z)\|, \quad (43)$$

com  $\nu$  sendo um parâmetro de penalidade. A função mérito (43) tem o objetivo de garantir o decréscimo da função objetivo. A norma utilizada em todo o trabalho é a norma Euclidiana.

Se em (42) os dois valores forem maior do que  $10^{-5}$ , então executa-se uma busca linear *backtraking*, que gera uma série do que denominamos comprimentos de passo  $\alpha_T$ , até que uma das seguintes condições seja satisfeita:

- Se o comprimento de passo  $\alpha_T$  for menor que  $10^{-5}$ , ou o número máximo de iterações for atingido, a busca linear é cancelada, a direção primal-dual  $d = (d_z, d_\lambda)$  é descartada e o Método de Região de Confiança, definido na seção 3.4 é chamado. Waltz et al. (2004) propõe o número máximo de iterações igual a 3;
- Se, um comprimento de passo  $\alpha_T$  satisfaz a condição de Armijo em (44),

$$\phi_\nu(z + \alpha_T \alpha_z^{\max} d_z) \leq \phi_\nu(z) + \eta \alpha_T \alpha_z^{\max} D\phi_\nu(z; d_z), \quad (44)$$

a busca linear é finalizada com sucesso e são definidos  $\alpha_z = \alpha_T \alpha_z^{\max}$  e

$$\alpha_\lambda = \alpha_T \alpha_\lambda^{\max}.$$

Caso contrário, isto é, se em (42) os dois valores forem menores de  $10^{-5}$ , o Método de Região de Confiança é chamado.

A não convexidade é medida pelos autores Waltz et al. (2004) através da condição de que o número de autovalores negativos da matriz dos coeficientes em (40), seja menor do que a soma do número de restrições de igualdade e de desigualdade.

No caso de não convexidade, é utilizado um Método de Região de Confiança o qual resolve um problema quadrático para calcular a direção, que possui propriedades de convergência global.

A busca linear *backtraking* é realizada do seguinte modo:

- Para  $j = 0$  definimos:

$$\alpha_T = 1; \quad (45)$$

- Para  $j = 1$ , se a iteração anterior foi uma iteração de busca linear, definimos:

$$\alpha_T = \frac{1}{2}; \quad (46)$$

- Caso contrário, se for uma iteração de região de confiança, é definido:

$$\alpha_T = \min \left( \frac{1}{2}, \frac{\Delta}{\|d_z\|} \right), \quad (47)$$

em que  $\Delta$  é o raio região de confiança atual.

- Para  $j > 1$ , o novo comprimento de passo é atualizado por:

$$\alpha_T \leftarrow \frac{\alpha_T}{2}. \quad (48)$$

A função mérito (43) pode rejeitar passos que fazem um bom progresso em direção à solução. Esta deficiência é superada no método mediante a aplicação de um passo de correção de segunda ordem (CSO), que é um passo tipo-Newton e que tem por objetivo melhorar a factibilidade.

O passo CSO é aplicado quando o primeiro comprimento do passo,  $\alpha_T = 1$  em (44), é rejeitado e se a razão para a rejeição puder ser atribuída somente a um aumento na norma das restrições, isto é, o segundo termo da função mérito (43). Mais especificamente, se a condição Armijo (44) não é satisfeita para  $\alpha_T = 1$ , e se  $\varphi_\mu(z + \alpha_z^{\max} d_z) \leq \varphi_\mu(z)$ , então, antes de calcular um comprimento de passo menor, tentamos um passo CSO, o qual determina as direções  $d^{CSO} = (d_z^{CSO}, d_\lambda^{CSO})$  através da solução do sistema:

$$\begin{bmatrix} W & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} d_z^{CSO} \\ d_\lambda^{CSO} \end{bmatrix} = \begin{bmatrix} -\nabla \varphi_\mu(z) - \alpha_z^{\max} W d_z - A^T [\lambda + \alpha_\lambda^{\max} d_\lambda] \\ -c(z + \alpha_z^{\max} d_z) \end{bmatrix} \quad (49)$$

em que  $A = A(x)$  e  $W = W(z, \lambda; \mu)$ , estão definidas em (40). Dessa forma definimos as direções:

$$\bar{d}_z = \alpha_z^{\max} d_z + d_z^{CSO}, \quad \bar{d}_\lambda = \alpha_\lambda^{\max} d_\lambda + d_\lambda^{CSO}, \quad (50)$$

em que:  $\bar{d}_z = (\bar{d}_x, \bar{d}_s)$  e  $\bar{d}_\lambda = (\bar{d}_h, \bar{d}_g)$ .

Em seguida, determinam-se os maiores escalares positivos  $\gamma_z^{CSO}, \gamma_\lambda^{CSO} \in (0, 1]$ , tal que:

$$s + \gamma_z^{CSO} \bar{d}_s \geq (1 - \tau) s \quad (51)$$

$$\lambda_g + \gamma_\lambda^{CSO} \bar{d}_g \geq (1 - \tau) \lambda_g \quad (52)$$

em que  $\tau$  é a mesma constante utilizada em (42), e calculamos a nova solução:

$$z^{CSO} = z + \gamma_z^{CSO} \bar{d}_z, \quad \lambda^{CSO} = \lambda + \gamma_\lambda^{CSO} \bar{d}_\lambda. \quad (53)$$

Se em (53) obtivermos  $\phi_v(z^{CSO}) < \phi_v(z)$ , então aceitamos o passo de correção de segunda ordem. Caso contrário, descartamos  $d^{CSO}$  e continuamos a busca linear *backtraking* ao longo do passo primal-dual  $(d_z, d_\lambda)$ , selecionando um novo comprimento de passo, caso  $j = 1$ .

Waltz et al. (2004) destacam que o cálculo do passo CSO não exige grandes cálculos computacionais, pois no lado esquerdo de (49), as matrizes  $A$  e  $W$  são as mesmas utilizadas em (40), isto é, elas não são avaliadas no ponto  $z + \alpha_z^{\max} d_z$ .

O lado direito de (49) não exige nenhuma avaliação adicional da restrição porque  $c(z + \alpha_z^{\max} d_z)$  foi avaliada em (44),  $\alpha_T = 1$ . Uma vez que a fatoração da matriz dos coeficientes foi realizada em (40), o custo total do passo de CSO se limita a resolver (49) e realiza uma avaliação adicional da função mérito para testar a aceitação do ponto da CSO.

### 3.4 A estratégia de região de confiança

No capítulo 2, vimos que os Métodos de Região de Confiança determinam um vetor passo ao invés de direção e passo como nos Métodos de Busca Linear.

Em seu trabalho, Byrd et al. (1998) denominam o vetor passo de  $d_z = (d_x, d_s)^T$  e estabelecem que ele deve ser limitado a um conjunto  $T_k$ , chamado de região de confiança. É definido  $T_k$ , de modo a atingir dois objetivos:

- Primeiro, restringe-se o vetor passo a uma região em que o problema quadrático:

$$\min_{d_x, d_s} \nabla f(x^k)^T d_x + \frac{1}{2} d_x^T \nabla_{xx}^2 L(x_k, s_k, \lambda_h, \lambda_g) d_x - \mu e^T S_k^{-1} d_s + \frac{1}{2} d_s^T \Sigma_k d_s \quad (54)$$

com  $\Sigma_k = S_k^{-1} \Lambda_{g_k}$ , é uma boa aproximação da função Lagrangiana, com  $S_k = \text{diag}(s_i^k)$  e  $\Lambda_{g_k} = \text{diag}(g_i^k)$ .

$$L(x, s, \lambda_h, \lambda_g) = f(x) - \mu \sum_{i=1}^m \ln s_i + \lambda_h^T h(x) + \lambda_g^T (g(s) + s) \quad (55)$$

e as equações lineares,

$$A_h(x^k)^T d_x + h(x^k) = r_h \quad (56)$$

e

$$A_g(x^k)^T d_x + d_s + g(x^k) + s^k = r_g$$

são boas aproximações para as restrições.

A região de confiança é obtida através de uma fronteira da forma  $\|(d_z)\| \leq \Delta^k$ , em que  $\Delta^k$  é o raio da região de confiança, e este é atualizado a cada iteração como segue:

$$\Delta^{k+1} = \begin{cases} \max\{7 \|d_z\|, \Delta\}; & \text{se } sred \geq 0,9 \\ \max\{2 \|d_z\|, \Delta\}; & \text{se } 0,3 \leq sred < 0,9 \\ \Delta; & \text{se } \eta \leq sred < 0,3 \end{cases} \quad (57)$$

em que  $0 < \eta < 1$ , (no trabalho é adotado  $\eta = 10^{-8}$ ), e *sred* representa uma redução suficiente da função mérito. Esta redução é dada por:

$$sred \equiv \frac{ared(d_z)}{pred(d_z)} \geq \eta, \quad (58)$$

em que

$$ared(d_z) = \phi_v(z) - \phi_v(z + d_z) \quad (59)$$

e

$$pred(d_z) = -\nabla \phi_v(z)^T d_z - \frac{\psi}{2} d_z^T W d_z + \nu (\|c(z)\| - \|c(z) + A(x)d_z\|) \quad (60)$$

são denominadas como redução atual e redução prevista da função mérito,

respectivamente e  $\psi = \begin{cases} 1 & \text{se } d_z^T W d_z > 0 \\ 0 & \text{caso contrário} \end{cases}$ .

O parâmetro de penalidade é indicado por  $\nu$  e sua atualização encontra-se em Waltz et al. (2004).

É imposto sobre o vetor passo a restrição que  $\|(d_x, d_s)\| \leq \Delta^k$ . Pelo fato que as variáveis de folgas não devem aproximar-se de zero, introduz-se uma matriz escala  $S_k^{-1}$ , que penaliza os passos  $d_s$  perto da fronteira da região factível. Esta região de confiança é denominada região de confiança escalada, e será definida como:

$$\|(d_x, S_k^{-1} d_s)\| \leq \Delta^k, \quad (61)$$

permitindo que  $\Delta^k$  seja maior que 1.

• O segundo objetivo da região de confiança é garantir que as variáveis de folga permaneçam positivas. Para isso, é imposto que  $s^k + d_s \geq (1 - \tau)s^k$ , em que  $\tau \in (0, 1)$  e nos testes é tomado por  $\tau = 0,995$ . A combinação da desigualdade,  $s^k + d_s \geq (1 - \tau)s^k$ , pode ser reformulada como  $d_s \geq -\tau s^k$ , a qual com (61), obtêm-se a forma final da região de confiança  $T_k$ , dada por:

$$\|(d_x, S_k^{-1} d_s)\| \leq \Delta^k \text{ e } d_s \geq -\tau s^k, \quad (62)$$

cujos autores destacam ser a mais apropriada para o algoritmo.

Desta forma estabelece-se o problema quadrático para região de confiança dado por:

$$\min_{d_x, d_s} \nabla f(x^k)^T d_x + \frac{1}{2} d_x^T \nabla_{xx}^2 L(x^k, s^k, \lambda_h^k, \lambda_g^k) d_x - \mu e^T S_k^{-1} d_s + \frac{1}{2} d_s^T \sum_k d_s \quad (63)$$

$$s.a: \quad A_h(x^k) d_x + h(x^k) = r_h \quad (64)$$

$$A_g d_x - d_s + g(x^k) - s^k = r_g \quad (65)$$

$$\|(d_x, S_k^{-1} d_s)\| \leq \Delta^k \quad (66)$$

$$d_s \geq -\tau s \quad (67)$$

$$\text{em que } \sum_k = S_k^{-1} \Lambda_{g_k}, \quad (68)$$

sendo  $\tau = 0,995$  e  $S_k = \text{diag}(s_i^k)$  e  $\Lambda_{g_k} = \text{diag}(g_i^k)$ .

Tendo o problema quadrático (54) e a região de confiança (62) definidos, tem-se que especificar a escolha do vetor residual  $r = (r_h, r_g)$  de (56) e será descrito na seção 3.5.

### 3.4.1 O passo da região de confiança

No método, segundo Waltz et. al. (2004), se o passo de busca linear é rejeitado, o algoritmo chama um passo de região de confiança.

É exigido que o raio  $\Delta^k$  seja calculado de acordo com o problema atual. O autor destaca que essa estratégia é importante quando dois passos de região de confiança são separados por uma longa sequência de passos de busca linear com o objetivo de preservar as propriedades de convergência global dos Métodos de Região de Confiança e o tamanho do  $\Delta^k$  é considerado na busca linear *backtracking* como visto em (47).

Seguindo a estratégia proposta por Waltz et. al. (2004), se o passo  $d_z^k$  mais recente foi de busca linear e bem sucedido,  $\Delta^{k+1}$  é definido por:

$$\Delta^{k+1} = 2\alpha_{zk} \left\| d_z^k \right\|, \quad (69)$$

caso contrário, se o passo mais recente foi um passo da região de confiança ou um passo de busca linear rejeitado, o raio da região de confiança é atualizado de acordo com a estratégia de atualização de região de confiança apresentada em (57).

Uma estratégia utilizada para evitar a repetição do cálculo de passos de busca linear que sejam rejeitados, é a de calcular as iterações seguintes com o Método de Região de Confiança até que um passo de sucesso seja obtido.

Testes numéricos mostram que esta estratégia simples mantém a informação de região de confiança atualizada e evita mudanças no tamanho do raio  $\Delta^k$ .

### 3.5 Método Gradiente Conjugado

De acordo com os autores Byrd et al. (2006), o algoritmo implementado no *Knitro* calcula passos usando um problema quadrático e regiões de confiança, como apresentado em (63) – (68). Esta formulação permite uma grande liberdade na escolha da matriz Hessiana e fornece um mecanismo para lidar com singularidades das Jacobiana e Hessiana. O preço para essa flexibilidade é uma interação mais complexa do que na abordagem de busca linear.

Primeiro, o problema de barreira (37) é um problema de otimização com restrições de igualdade e pode ser resolvido usando um Método de Programação Quadrática Sequencial (PQS) com regiões de confiança. A simples aplicação de técnicas de PQS para a solução do problema barreira (39) leva a passos ineficientes que tendem a violar a positividade das variáveis de folga e são frequentemente interrompidos pela restrição da região de confiança de acordo com os autores Byrd et al. (2006). Para superar este problema, o autor propõe o método PQS voltado especificamente para o problema de barreira.

Este método de PQS é definido por: na iteração atual  $(x^k, s^k)$ , e para um dado parâmetro de barreira  $\mu$ , primeiro calculam-se as estimativas dos multiplicadores de Lagrange  $(\lambda_h, \lambda_g)$  e, em seguida, calcula-se um vetor passo  $d = (d_x, d_s)$ , que tem por objetivo resolver o subproblema (63) – (68).

Byrd et al. (2006), afirmam que o subproblema (63) – (68) é difícil de minimizar exatamente devido a presença da restrição não linear em (66) e os limites em (67).

As coordenadas do vetor residual,  $r = (r_h, r_g)$ , são escolhidas como um número muito pequeno para que haja compatibilidade nas restrições e passos factíveis, onde os índices  $h$  e  $g$  correspondem às restrições de igualdade e desigualdade, respectivamente.

Para solução do subproblema (63) – (68), o cálculo do vetor passo  $d_x$ , através do pacote de algoritmos *Knitro*, é uma combinação de um passo normal  $v$  que tenta satisfazer as restrições lineares de (66), (67), com  $r = 0$ , a restrição de

região de confiança e um passo tangencial no espaço tangente das restrições de modo a encontrar a otimalidade.

Para calcular o passo normal  $v = (v_x, v_s)$ , é formulado o seguinte subproblema:

$$\min_v \left\| A_h v_x + h \right\|^2 + \left\| A_g v_x - v_s + g - s \right\|^2 \quad (70)$$

$$s.a: \left\| (v_x, S^{-1}v_s) \right\| \leq 0,8\Delta \quad (71)$$

$$v_s \geq \frac{-\tau_s}{2} \quad (72)$$

Para simplificar as restrições usa-se a escala definida por:

$$\tilde{v} = (v_x, \tilde{v}_s) = (v_x, S^{-1}v_s). \quad (73)$$

O vetor das estimativas dos multiplicadores de Lagrange  $\lambda = (\lambda_h, \lambda_g)^T$  são escolhidos para minimizar a norma Euclidiana das restrições:

$$\begin{aligned} \nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g &= 0 \\ S\lambda_g - \mu e &= 0 \end{aligned} \quad (74)$$

e são dadas por:

$$\lambda^k = \begin{bmatrix} \lambda_h \\ \lambda_g \end{bmatrix} = \lambda^{LS}(x^k, s^k, \mu) = (\hat{A}_k^T \hat{A}_k)^{-1} \hat{A}_k^T \begin{bmatrix} -\nabla f(x^k) \\ \mu e \end{bmatrix} \quad (75)$$

com

$$\hat{A}_k = \begin{bmatrix} A_h(x^k) & A_g(x^k) \\ 0 & S_k \end{bmatrix},$$

ou seja,  $\lambda^k$  é expressa pela solução quadrados mínimos  $\lambda^{LS}$  para o sistema (74).

Em (75)  $\lambda_g$  pode ser negativo, porém o método não força a positividade dos multiplicadores, mas garante que o problema quadrático continue convexo nas variáveis de folga. Logo o  $i$ -ésimo elemento da diagonal de  $\Sigma_k$  é definido por:

$$\sigma_k^i = \begin{cases} \frac{\lambda_g^i}{s^i}; & \text{se } \lambda_g^i > 0 \\ \frac{\mu}{(s^i)^2}; & \text{caso contrário.} \end{cases} \quad (76)$$

É calculada uma solução aproximada do problema (70) – (72) utilizando uma adaptação da abordagem *dogleg*, descrita no capítulo 2, que minimiza (70) ao longo de um trajeto linear composto de um passo de gradiente na norma usada em (71) e a norma mínima do passo de Newton com relação à mesma norma. A escala,  $S^{-1}v_s$ , na norma tende a limitar a extensão em que os limites nas variáveis de folga são violados.

Uma vez que o passo normal  $v$  é calculado, definimos os vetores  $r_h$  e  $r_g$  em (65) e (66), como os resíduos no cálculo de passo normal, ou seja,

$$\begin{aligned} r_h &= A_h v_x + h, \\ r_g &= A_g v_x - v_s + (g - s). \end{aligned} \quad (77)$$

e o subproblema (63) – (68) pode, portanto, ser reescrito como:

$$\min_{d_x, d_s} \nabla f(x)^T d_x - \mu e^T S^{-1} d_s + \frac{1}{2} (d_x^T \nabla_{xx}^2 L d_x + d_s^T \Sigma d_s) \quad (78)$$

$$s.a: A_h d_x = A_h v_x \quad (79)$$

$$A_g d_x - d_s = A_g v_x - v_s \quad (80)$$

$$\|d_x, S^{-1} d_s\| \leq \Delta \quad (81)$$

que é denominado como um subproblema tangencial.

Para encontrar uma solução aproximada de  $d$  em (78), primeiro introduz-se a escala  $\tilde{d} \leftarrow S^{-1} d_s$ , que transforma (81) em uma inequação elipsoidal. Em seguida, aplica-se o Método do Gradiente Conjugado Projetado, descrito no capítulo 3 para o

problema quadrático transformado. Durante a solução do Método do Gradiente Conjugado Projetado, é usada a estratégia Steihaug, descrito no capítulo 2, que monitora a satisfação da restrição da região de confiança (81), ou seja, o limite desta região é atingido ou é detectada curvatura negativa, isto é, um critério de parada seja satisfeito.

### 3.6 O Método do Gradiente Conjugado Projetado

Segundo os autores Byrd et al. (2006), uma das principais subrotinas compartilhadas pelos algoritmos executados no *Knitro*, é uma iteração do Método do Gradiente Conjugado Projetado. O subproblema tangencial requer a solução de um problema de programação quadrática com restrição de igualdade. Para solucionar este problema é usada uma iteração do Método do Gradiente Conjugado Projetado, que é adequado para problemas de grande porte e pode lidar com o caso de curvatura negativa, sem a necessidade de modificações na matriz Hessiana (Coleman, 1994; Gould et al., 2001; Keller et al., 2000; Luksan e Vlcek, 1998; Polyak, 1969).

O Método do Gradiente Conjugado Projetado é apresentado a seguir.

Considere o problema de programação quadrática:

$$\min_x \frac{1}{2} x^T G x + h^T x \quad (82)$$

$$s.a: Ax = b \quad (83)$$

e é assumido que  $G$  é uma matriz definida positiva no espaço nulo de  $A$ . Uma forma de resolver (82) e (83) é eliminando as restrições (84) e aplicando o Método do Gradiente Conjugado para o problema reduzido. Uma estratégia equivalente é aplicar uma forma especial da iteração de CG para o sistema de KKT associado ao problema (83), que é dado por:

$$\begin{bmatrix} G & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -h \\ b \end{bmatrix} \quad (84)$$

Embora a matriz dos coeficientes não seja definida positiva, podemos aplicar o Método do Gradiente Conjugado em (84), introduzindo um pré-condicionamento e uma projeção no Método do Gradiente Conjugado, tal que seja resolvido de forma eficiente o problema reduzido (82). O operador de pré-condicionamento/projeção é denotado, neste trabalho, por  $P$  e sua definição é dada após o algoritmo.

---

**Algoritmo 7:** Método do Gradiente Conjugado Projetado Pré-condicionado

---

**Início**

Escolha um ponto inicial  $x^0$  satisfazendo  $Ax^0 = b$ .

Defina  $x = x^0$ ,

Calcule  $r = Gx + h$ ,

$$z = Pr;$$

$$p = -z;$$

**Repita** os passos seguintes, até que  $\|z\|$  seja menor do que uma determinada tolerância:

$$\alpha = \frac{r^T z}{p^T Gp}$$

$$x \leftarrow x + \alpha p$$

$$r^{k+1} = r + \alpha Gp$$

$$z^{k+1} = Pr^{k+1}$$

$$\beta = \frac{(r^{k+1})^T z^{k+1}}{r^T z}$$

$$p \leftarrow -z^{k+1} + \beta p$$

$$z \leftarrow z^{k+1}$$

$$r \leftarrow r^{k+1}$$

**Fim**

**Fim**

---

Esta iteração é exatamente a do Método do Gradiente Conjugado padrão (Nocedal e Wright, 1999) para a solução de sistemas simétricos definidos positivos e pré-condicionados. A diferença é que, normalmente,  $P$  é uma matriz simétrica definida positiva, em que neste trabalho, representa uma projeção e pré-condicionamento da matriz, expressa a seguir.

Dado um vetor  $r$ , calculamos  $z = Pr$  como a solução do sistema:

$$\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z \\ w \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix} \quad (85)$$

em que  $D$  é uma matriz simétrica definida positiva no espaço nulo de  $A$ , e  $w$  é um vetor auxiliar. Para (85) ser uma operação de pré-condicionamento prático,  $D$  deve ser uma matriz esparsa, de modo que a solução de (85) é significativamente menos cara computacionalmente do que a solução de (84).

Por construção  $z = Pr$  está no espaço nulo de  $A$ , e por isso todas as direções de busca são geradas pelo Método do Gradiente Conjugado Projetado (PCG). Visto que inicialmente  $Ax^0 = b$ , em todas as iterações subsequentes  $x$  também satisfaz as restrições lineares. Para visualizar a iteração relativa ao de método do CG reduzido em que eliminam-se as restrições (83) e aplica-se o método do CG para um problema de dimensão  $n-l$ , note que todas as iterações do algoritmo de PCG pode ser expressa como  $x = x^0 + Zu$ , para algum vetor  $u \in R^{n-l}$ , e as colunas da matriz  $Z$ ,  $n \times (n-l)$  formam uma base para o espaço nulo de  $A$ . Neste espaço nulo, a solução do problema de programação quadrática (82) é dada pelo vetor  $u$  que satisfaz:

$$\left( Z^T G Z \right) u = Z^T \left( G x^0 + h \right) \quad (86)$$

Pode ser mostrado que nas iterações  $x$  as soluções geradas pelo algoritmo PCG são dadas por  $x = x^0 + Zu$ , onde  $u$  é calculado pela iteração do Método do Gradiente Conjugado Pré-Condicionado aplicado ao sistema (86), utilizando a matriz  $Z^T D Z$  como um pré-condicionador. Portanto, o algoritmo PCG é uma iteração padrão do CG pré-condicionado quando  $G$  e  $D$  são definidas positivas no espaço nulo de  $A$ .

Existem duas vantagens do algoritmo PCG em relação ao algoritmo CG reduzido. Em primeiro lugar, não é necessário calcular a base do espaço nulo e, conseqüentemente, não existe nenhum risco que o mau condicionamento em  $Z$  deteriore a taxa de convergência da iteração CG. Além disso, no algoritmo *Interior-*

CG em que primeiro escalamos as variáveis de folga utilizando  $\tilde{d} \leftarrow S^{-1}d_s$ , de modo que a matriz  $A$  em (82) e (83) tem a forma:

$$\begin{bmatrix} A_h & 0 \\ A_g & -S \end{bmatrix}, \quad (87)$$

portanto, não existe mau condicionamento causado pelas variáveis de folga quando estas se aproximam de zero.

A segunda vantagem é que a matriz projeção em (85) também pode ser usada para calcular o passo normal e os multiplicadores de Lagrange, assim, o custo extra de cada um destes cálculos é a repetição deste processo envolvendo os fatores desta matriz de projeção.

Nos algoritmos *Interior-CG*, problemas quadráticos da forma (82) são resolvidos sujeitos a restrições de região de confiança  $\|x\| \leq \Delta$  e, além disso,  $G$  pode não ser definida positiva no espaço nulo de  $A$ . O algoritmo PCG é adaptado para este caso, seguindo a abordagem de Steihaug vista no capítulo 2: O algoritmo PCG termina se a região de confiança é alcançada ou se uma curvatura negativa é encontrada.

O *Knitro* define  $D = I$  em (85), de modo que o pré-condicionador remove apenas o mau condicionamento associado com a matriz  $A$ .

O algoritmo PCG assume que um ponto inicial factível  $x^0$  é fornecido. A fatoração do sistema em (85) permite calcular tal ponto, resolvendo:

$$\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} w \\ x^0 \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad (88)$$

que é, na verdade, a solução de norma mínima na norma ponderada por  $D$ .

### 3.7 Critérios de parada

Nesta seção apresentamos os critérios de parada para programação não linear e para o problema de barreira.

### 3.7.1 Programação não linear

Segundo os autores Waltz et. al (2004), o teste de parada deve ser invariante sob o escalonamento das variáveis, da função objetivo e das restrições. Atingir essa invariância de escala completa é difícil e não é desejável nos casos em que determinados valores aproximam-se de zero. Os seguintes testes tentam alcançar um equilíbrio entre praticidade e invariância de escala.

As tolerâncias para o teste de parada para programação não linear (PNL),  $\epsilon^{opt}$  e  $\epsilon^{feas}$ , em que as nomenclaturas *opt* e *feas* significam otimalidade e factibilidade, respectivamente, são fornecidas pelo usuário. Nos testes feitos pelos autores, é utilizada a tolerância igual a  $10^{-6}$ . O algoritmo termina se em uma iteração  $(x, s, \lambda)$  satisfaz as seguintes condições:

$$\|\nabla f(x) + A_h(x)^T \lambda_h + A_g(x)^T \lambda_g\|_\infty \leq \max\{1, \|\nabla f(x)\|_\infty\} \epsilon^{opt} \quad (89)$$

$$\|S \lambda_g\|_\infty \leq \max\{1, \|\nabla f(x)\|_\infty\} \epsilon^{opt} \quad (90)$$

$$\|(h(x), g(x)^+)\|_\infty \leq \max\{1, \|(h(x^0), g(x^0)^+)\|_\infty\} \epsilon^{feas} \quad (91)$$

em que  $g(x)^+ = \max\{0, g(x)\}$  e  $x^0$  é o ponto inicial. Quando o máximo não é alcançado por 1, o fator de escala em (89) – (91) faz este teste invariante para as escalas em  $f$ ,  $g$  e  $h$  e às mudanças lineares da variável  $x$ . O fator 1 é necessário para a garantia do teste quando  $\|\nabla f(x)\|$  é zero ou quase zero. O teste de complementaridade (90) é baseado no fato de que a escala de  $s$  é dependente da escala de  $g$  e  $h$  e a magnitude do  $\lambda_g$  é proporcional à  $\frac{\|\nabla f\|}{\|A\|}$ . Assim (90) é invariante com a escala de  $f$ ,  $g$  e  $h$ . Um fator de escalonamento mais complexo

empregando  $\|A\|$  tornaria invariante a mudança linear nas variáveis, porém este fator é utilizado em (89) e (90) para simplificar.

O fator de escala de factibilidade é difícil de escolher. Se as restrições são lineares,  $\|(h(0), g(0)^+)\|_\infty$  é um fator de normalização adequado porque mede a magnitude dos vetores do lado direito de todas as restrições. Mas desde que  $(h(0)$  ou  $g(0)^+)$  não pode ser definida em alguns problemas, usamos o ponto inicial  $x^0$ .

### 3.7.2 O problema de barreira

Nesta seção, apresenta-se o teste de parada para o problema da barreira. A experiência citada pelos autores Waltz et. al (2004), é que a escolha deste teste de parada afeta significativamente a eficiência e robustez do Método de Ponto Interior. Para o atual valor de  $\mu$ , são escolhidas tolerâncias  $\epsilon_\mu^{opt}$  e  $\epsilon_\mu^{feas}$  e impostas nas condições de parada dos problemas de barreiras como segue:

$$\|\nabla f(x) + A_h(x)^T \lambda_h + A_g(x)^T \lambda_g\|_\infty \leq \max\{1, \|\nabla f(x)\|_\infty\} \epsilon_\mu^{opt} \quad (92)$$

$$\|S \lambda_g - \mu e\|_\infty \leq \max\{1, \|\nabla f(x)\|_\infty\} \epsilon_\mu^{opt} \quad (93)$$

$$\|(h(x), g(x) + s)\|_\infty \leq \max\{1, \|(h(x^0), g(x^0)^+)\|_\infty\} \epsilon_\mu^{feas} \quad (94)$$

Note que os fatores de escala são idênticos aos utilizados no teste de parada para programação não linear (89) – (91) e que o lado esquerdo destes dois testes diferem apenas na utilização do termo adicional  $-\mu e$  em (93) e a utilização de  $g(x) + s$  em vez de  $g(x)^+$  em (94).

As tolerâncias  $\epsilon_\mu^{opt}$  e  $\epsilon_\mu^{feas}$  serão escolhidas para serem proporcionais ao  $\mu$  na maioria do tempo e, assim, tornar-se mais forte à medida que  $\mu$  diminui. Pretende-se evitar que os valores iniciais de  $\mu$ ,  $\epsilon_\mu^{opt}$  e  $\epsilon_\mu^{feas}$  assumam valores pequenos, porque isso pode levar a erros de arredondamento significativos. Portanto,

são determinados os valores de  $\mu$ ,  $\varepsilon_\mu^{opt}$  e  $\varepsilon_\mu^{feas}$  para que satisfaçam o teste de parada do problema de barreira automaticamente implicando na satisfação do teste de parada PNL.

Uma vez que os fatores de escala à esquerda em (89) e (92) são idênticos, qualquer ponto que satisfaz (92) automaticamente satisfaz (90) para todos os valores  $\varepsilon_\mu^{opt} \leq \varepsilon_\mu^{feas}$ . Também, a partir de (93), temos que, para um dado par complementar  $s_i[\lambda_g]_i$  que satisfaça o teste de parada do problema de barreira  $s_i[\lambda_g]_i \leq \mu + \max\{1, \|\nabla f(x)\|_\infty\} \varepsilon_\mu^{opt}$  e, a fim de satisfazer também o teste de parada PNL (90), deve satisfazer  $s_i[\lambda_g]_i \leq \max\{1, \|\nabla f(x)\|_\infty\} \varepsilon_i^{opt}$ .

Portanto uma iteração  $(x, s)$  que satisfaz (93) também satisfaz (90) se:

$$\varepsilon_\mu^{opt} \leq \varepsilon^{opt} - \frac{\mu}{\left(\max\{1, \|\nabla f(x)\|_\infty\}\right)}. \quad (95)$$

Assim, um ponto que satisfaz a condição de parada de KKT da barreira (92), (93) também satisfaz as condições de parada de KKT do PNL (89), (90) para todos os valores de  $\varepsilon_\mu^{opt}$  que satisfaçam:

$$\varepsilon_\mu^{opt} \leq \varepsilon^{opt} - \mu. \quad (96)$$

Nota-se que a condição (96) só é válida quando o lado direito de (96) é positivo, ou seja,  $\mu \leq \varepsilon^{opt}$ .

Para factibilidade, uma vez que  $s > 0$ , temos que  $\|g(x)^{k+1}\|_\infty \leq \|g(x) + s\|_\infty$ .

Portanto a partir de (91) e (94) segue que para todo

$$\varepsilon_\mu^{feas} \leq \varepsilon^{feas} \quad (97)$$

o teste de factibilidade PNL (91) será satisfeito se o teste de factibilidade de barreira (94) for satisfeito.

Sujeito aos valores mínimos dados por (96) e (97), definimos  $\varepsilon_{\mu}^{opt}$  e  $\varepsilon_{\mu}^{feas}$  igual a  $\theta\mu$ , em que  $\theta$  é um parâmetro fixo do algoritmo,  $\theta = 1$ . As tolerâncias do teste de parada do parâmetro de barreira são determinadas pelas seguintes fórmulas:

$$\varepsilon_{\mu}^{opt} = \max \left\{ \theta\mu, \varepsilon^{opt} - \mu \right\} \quad (98)$$

$$\varepsilon_{\mu}^{feas} = \max \left\{ \theta\mu, \varepsilon^{feas} \right\} \quad (99)$$

Esta opção não assegura a resolução completa do subproblema de barreira, mas garante que não será adotado qualquer limite inferior no parâmetro de barreira  $\mu$ . Embora um valor levemente pequeno de  $\mu$  não resolva o subproblema de barreira devido aos limites das tolerâncias estabelecidas em (98) e (99), ainda deseja-se evitar que  $\mu$  decresça rapidamente, pois isso pode levar a iteração ao fracasso.

A partir de (98) e (99), a satisfação do teste de parada do parâmetro de barreira implica na satisfação do teste de parada PNL se  $\mu \leq \min \left\{ \frac{\varepsilon^{opt}}{(1+\theta)}, \frac{\varepsilon^{feas}}{(\theta)} \right\}$ , mas para garantir que  $\mu$  não seja restrito indevidamente, é imposto um valor mínimo de  $\mu$  com base na tolerância PNL:

$$\mu_{\min} = \frac{\min \left\{ \varepsilon^{opt}, \varepsilon^{feas} \right\}}{100}. \quad (100)$$

### 3.8 O algoritmo do Método de Ponto Interior implementado no *Knitro*

---

**Algoritmo 8:** Método de Ponto Interior proposto por Waltz et. al (2004)

---

#### **Início**

Seja  $l$  o número de restrições de igualdade e  $m$  o número de restrições de desigualdade;

Escolha valores iniciais para  $z^0 = (x^0, s^0)$ , para os parâmetros  $0 < \eta$  e  $0 < \delta < 1$ , em que  $\delta = 10^{-5}$ ,  $\eta = 10^{-8}$ , um número inteiro para o número máximo de iterações ( $imax=3$ ) e os valores para as tolerâncias  $\varepsilon_{\mu}^{opt} = 10^{-6}$  e  $\varepsilon_{\mu}^{feas} = 10^{-6}$ ;

Defina os valores iniciais para os multiplicadores de Lagrange  $\lambda_k$  por (75);

Defina o raio da região de confiança  $\Delta^k > 0$  através de (57) ou (69);

Defina o parâmetro de barreira  $\mu_k > 0$  ( $\mu_0 = 10^{-1}$ );

Defina  $k = 0$

**Repita** até que um critério de parada, dado por (89) – (91) para o problema de programação não-linear (36) seja satisfeito;

**Repita** até que o teste de parada para o problema de barreira em (92) – (94) seja satisfeito;

Resolver o sistema primal-dual (40) e registre o número de auto valores negativos da matriz dos coeficientes (*neig*);

Defina *Busca\_Linear* = *Falso*.

**Se**  $neig \leq l + m$

Resolva (40) para obter as direções de busca  $d = (d_z, d_{\lambda})$

Calcule  $\alpha_z^{\max}, \alpha_{\lambda}^{\max}$  utilizando as equações em (42)

**Se**  $\min\{\alpha_z^{\max}, \alpha_{\lambda}^{\max}\} > \delta$

Defina  $j = 0$ ,  $\alpha_j = 1$ .

**Enquanto**  $(j \leq imax), (\alpha_j > \delta)$  e *Busca\_linear* == *Falso*

**Se**  $\phi_\nu(z + \alpha_T \alpha_z^{\max} d_z) \leq \phi_\nu(z) + \eta \alpha_T \alpha_z^{\max} D\phi_\nu(z; d_z)$

Defina  $\alpha_z = \alpha_T \alpha_z^{\max}$ ,  $\alpha_\lambda = \alpha_T \alpha_\lambda^{\max}$ .

Defina  $z^{k+1}$ ,  $\lambda^{k+1}$  utilizando a equação (41).

Calcule o raio  $\Delta^{k+1}$

Defina *Busca\_Linear* = Verdadeiro

**Senão**

Defina  $j = j + 1$

Escolha um valor menor para  $\alpha_T$  utilizando (45),

(46), (47) e (48);

**Fim**

**Fim**

**Fim**

**Fim**

**Se** *Busca\_Linear* == Falso

Calcule  $(z^{k+1}, \lambda^{k+1})$  utilizando o procedimento *Dogleg*.

Calcule os multiplicadores de Lagrange utilizando (74), (75) e (76);

Calcule  $\nabla_{x^k}^2 L(x^k, s^k, \lambda_h^k, \lambda_g^k)$  e  $\Sigma_k$  usando (76);

Calcule o passo tangencial  $w_k$  utilizando o processo PCG;

Atualize o parâmetro de penalidade  $\nu_k$  de acordo com Waltz et al. (2004);

Calcule *pred*( $d_k$ ) e *ared*( $d_k$ ) através de (59) e (60);

**Se** *ared*( $d_k$ )  $\geq \eta$  *pred*( $d_k$ )

Defina  $x^{k+1} = x^k + d_x$ ,  $s^{k+1} = s^k + d_s$  e atualize  $\Delta^{k+1}$  em (57) e (69)

**Senão** utilize o procedimento CSO para obter  $d^{CSO} = (d_z^{CSO}, d_\lambda^{CSO})$ ;

**Se**  $d^{CSO} \neq 0$  e *ared*( $\alpha_z^{\max} + d_z + d^{CSO}$ )  $\geq \eta$  *pred*( $\alpha_z^{\max} + d_z$ ) e

$s + \alpha d_s + d^{CSO} \geq (1 - \tau)s$ ;

Defina  $x^{k+1} = x^k + d_x + d^{CSO}$ ,  $s^{k+1} = s^k + d_s + d^{CSO}$ ,

$\lambda^{k+1} = \lambda^k + d_\lambda + d^{CSO}$  e  $\Delta^{k+1} = \Delta^k$ ;

**Senão**

Defina  $x^{k+1} = x^k$ ,  $s^{k+1} = s^k$ ,  $\Delta^{k+1} \in [0,1\Delta^k; 0,5\Delta^k]$ ;

**Fim**

**Fim**

Defina  $\mu_{k+1} = \mu_k$

Defina  $k = k + 1$

**Fim**

**Fim**

**Fim**

---

No próximo capítulo apresentaremos as estratégias de atualização para o parâmetro de barreira que estão disponíveis no pacote de algoritmos *Knitro*.

## CAPÍTULO 4

### ATUALIZAÇÕES DO PARÂMETRO DE BARREIRA

Neste capítulo iremos apresentar as estratégias de atualização do parâmetro de barreira disponíveis no pacote de otimização *Knitro*.

O pacote oferece seis opções ao usuário para alteração das estratégias de atualização do parâmetro de barreira e uma opção na qual a escolha de uma das estratégias de atualização, entre as seis disponíveis, é realizada automaticamente pelo *Knitro*.

As estratégias de atualização do parâmetro de barreira abordadas na literatura são divididas em adaptativa e monótona. As estratégias adaptativas permitem mudanças no parâmetro de barreira a cada iteração, e muitas vezes são eficientes na prática, mas geralmente não possuem propriedades de convergência global. Vários autores utilizaram esse tipo de estratégia, entre eles podemos citar El-Bakry et al. (1996), Gay et al. (1998), Ulbrich et al. (2004), entre outros.

As estratégias monótonas tem por filosofia fixar o parâmetro de barreira até que uma solução aproximada do problema de barreira seja determinado. A mais conhecida é a abordagem de Fiacco-McCormick. Ela tem sido utilizada em vários algoritmos não lineares de pontos interiores, dos quais podemos citar Betts et al. (2003), Byrd et al. (1999), e foi também aplicada no pacote de otimização *Knitro*. De acordo com El-Bakry et al. (1996), Wachter e Biegler (2005) a estratégia de Fiacco-McCormick fornece uma estrutura que garante a convergência global, porém é muito sensível à escolha do ponto inicial  $x^0$  e do valor inicial do parâmetro de barreira, ao dimensionamento do problema e apresenta problemas quando as soluções se aproximam prematuramente da fronteira da região factível.

Nocedal et. al (2008) destacam que nos pacotes de algoritmos de otimização como o *Knitro*, são necessárias estratégias de atualização mais dinâmicas com o intuito de melhorar a eficiência dos Métodos de Pontos Interiores á resolução de problemas não lineares.

Os algoritmos utilizados no *Knitro* apenas garantem a convergência de primeira ordem de pontos estacionários, pois impor a convergência de segunda ordem desses pontos exigiria uma estimativa do menor autovalor da matriz Hessiana, o que é computacionalmente caro no caso de matrizes de grande dimensão. Os passos gerados pelo método promovem a convergência para pontos de mínimo por garantir propriedades de descida para o problema de barreira.

Como já foi citado, no *Knitro* estão disponíveis seis diferentes estratégias de atualização do parâmetro de barreira, as quais foram utilizadas neste trabalho (Waltz e Nocedal, 2003) e são descritas nas próximas seções.

#### **4.1 Estratégia (i): O parâmetro de barreira é decrescido monotonicamente**

A sequência de parâmetros de barreira  $\{\mu_k\}$  deve convergir para zero, e se possível rapidamente. Regras para decrescer o parâmetro de barreira têm sido estudadas por diversos pesquisadores, porém estas regras empregam vários parâmetros que podem ser difíceis de selecionar na prática.

Na regra monótona de Fiacco-McCormick o parâmetro de barreira é fixado, e uma série de passos primais-duais são calculados até que as condições de otimalidade do problema de barreira sejam satisfeitos com uma determinada precisão, e em seguida o parâmetro de barreira é decrescido.

No *Knitro*, segundo os autores Waltz et. al (2004), é utilizada a seguinte estratégia para atualizar  $\mu$ . Se o problema barreira mais recente foi resolvido em menos de três iterações, é definido:

$$\mu_{k+1} = \frac{\mu_k}{100}, \quad (101)$$

caso contrário:

$$\mu_{k+1} = \frac{\mu_k}{5}. \quad (102)$$

Esta estratégia foi escolhida tendo em vista os testes computacionais realizados pelos autores Waltz et al. (2004).

## 4.2 Estratégia (ii): O parâmetro de barreira é calculado por uma regra adaptativa com base na diferença de complementaridade

Segundo Nocedal et. al (2008), existem diversos processos adaptativos propostos especificamente para Métodos de Pontos Interiores á problemas não lineares alguns deles propostos por El-Bakry et al. (1996), Gay et al. (1998) e Ulbrich et al. (2004). Porém a estratégia adaptativa mais comum para a escolha do parâmetro de barreira  $\mu$  é defini-lo proporcional ao valor da complementaridade atual, isto é,

$$\mu = \sigma \frac{\lambda_g^T s}{n} \quad (103)$$

em que  $\sigma > 0$  é um parâmetro de centralização,  $n$  é o número de variáveis do problema,  $\lambda_g^T$  é o vetor dos multiplicadores de Lagrange associado à restrição de desigualdade e  $s$  é o vetor das variáveis de folga.

Existem diversas heurísticas para a escolha do parâmetro de centralização  $\sigma$  em (103). Uma das heurísticas utilizada no *Knitro* é a mesma estratégia empregada no pacote de software LOQO (Vanderbei e Shanno, 1999), a qual foi escolhida devido ao seu notável sucesso na prática. O parâmetro  $\sigma$  é definido como:

$$\sigma = 0,1 \min \left( 0,05 \frac{1-\xi}{\xi}, 2 \right)^3, \text{ em que } \xi = \frac{\min_i \{ \lambda_g^{(i)} s^{(i)} \}}{\frac{\lambda_g^T s}{n}} \quad (104)$$

Note que  $\xi$  mede o desvio do menor produto de complementaridade  $\lambda_g^{(i)} s^{(i)}$  em relação à média. Quando  $\xi = 1$  (todos os produtos individuais são iguais aos da média), temos que  $\sigma = 0$  e o algoritmo toma um passo “agressivo”. A regra de (104) escolhe sempre  $\sigma \leq 0,8$ , de modo que mesmo que o valor de  $\mu$  possa aumentar de

uma iteração para outra e então, ele nunca irá ser escolhido maior do que o valor atual da complementaridade  $\frac{\lambda_g^T s}{n}$ .

### 4.3 Estratégia (iii): É realizada uma investigação sobre o passo (afim escala) para determinar dinamicamente o valor do parâmetro barreira a cada iteração

Em (104), uma outra maneira de calcular  $\sigma$  é baseada no Método Previsor-Corretor de Mehrotra (MPC) (Mehrotra, 1992), para a programação linear. Este método determina o valor de  $\sigma$  usando o cálculo de um passo preliminar, isto é, um passo afim escala. Nocedal et. al (2008) descrevem a estratégia de Mehrotra para o caso de programação não-linear como segue:

Primeiro, calculamos um passo escala afim:

$$\left( d_z^{aff}, d_\lambda^{aff} \right) \quad (105)$$

definindo  $\mu = 0$  em (40), isto é,

$$\begin{bmatrix} W & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} d_z^{aff} \\ d_\lambda^{aff} \end{bmatrix} = - \begin{bmatrix} \nabla L(z, \lambda, 0) \\ c(z) \end{bmatrix} \quad (106)$$

em que  $\nabla L(z, \lambda, 0)$  é o gradiente da função Lagrangiana.

Em seguida, calculam-se  $\alpha_z^{aff}$  e  $\alpha_\lambda^{aff}$  e para serem os maiores comprimento de passo em  $(0,1]$ , que podem ser tomados ao longo da direção (105) antes de violar as condições de não-negatividade  $(s, \lambda) \geq 0$ . Fórmulas explícitas para estes valores são dados por:

$$\begin{aligned} \alpha_z^{\max} &= \max \left\{ \alpha^{aff} \in (0,1] : s + \alpha_s^{aff} d_s^{aff} \geq (1-\tau) s \right\} \\ \alpha_\lambda^{\max} &= \max \left\{ \alpha^{aff} \in (0,1] : \lambda_g + \alpha_{\lambda_g}^{aff} d_{\lambda_g}^{aff} \geq (1-\tau) \lambda_g \right\} \end{aligned} \quad (107)$$

com  $\tau = 1$ .

A seguir, define-se  $\mu^{aff}$  a ser o valor da complementaridade que seria obtida por um passo completo até a fronteira, isto é:

$$\mu^{aff} = \frac{\left( \lambda_g + \alpha_{\lambda_g}^{aff} d_{\lambda_g}^{aff} \right)^T \left( s + \alpha_s^{aff} d_s^{aff} \right)}{n} \quad (108)$$

e o parâmetro de centralização é escolhido a partir de

$$\sigma = \left( \frac{\mu^{aff}}{\frac{\lambda_g^T s}{n}} \right)^3. \quad (109)$$

Esta heurística de  $\sigma$  é baseada na experiência com problemas de programação linear e revelou-se eficaz também para a programação quadrática e convexa. Nota-se que quando é feito um bom progresso ao longo da direção escala afim, temos  $\mu^{aff} \ll \frac{\lambda_g^T s}{n}$ , de modo que  $\sigma$  é obtido a partir desta fórmula. Em outros casos,  $\sigma$  pode ser escolhido maior do que 1.

#### **4.4 Estratégia (iv): É utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira utilizando o passo corretor**

Outra estratégia adotada é a do MPC de Mehrotra (Mehrotra, 1992) considerando o cálculo do passo com três fases, como apresentado em Wright (1997). Primeiro, o algoritmo calcula o passo escala-afim (105) e o utiliza para determinar o parâmetro de barreira dado em (103), com  $\sigma$  calculado por (109). Em seguida, o algoritmo calcula um passo primal-dual a partir de (40) utilizando esse valor de  $\mu$ .

Finalmente, um passo corretor,  $d^{corr}$ , é calculado pela solução de (40) com o lado direito dado por:

$$-\left(0, d\Lambda^{aff} dS^{aff} e, 0\right)^T \quad (110)$$

onde  $d\Lambda^{aff}$  e  $dS^{aff}$  são as matrizes diagonais dos elementos dados por  $d\lambda_g^{aff}$  e  $ds^{aff}$ , respectivamente. O passo MPC completo equivale à soma dos passos primal-dual e corretor.

Nocedal et al. (2008) concluem em seu trabalho que para problemas não lineares a inclusão do passo corretor não necessariamente é benéfica. Isso contrasta com a experiência em programação linear e programação quadrática e convexa, onde o passo corretor é conhecido por acelerar a iteração do Método de Ponto Interior, sem influenciar na sua robustez.

Destaca também que o passo corretor pode ser prejudicial na programação linear e em casos de programação quadrática se uma estratégia eficiente de ponto inicial não é utilizada. O cálculo do passo corretor não faz parte da seleção do parâmetro de barreira segundo Nocedal et al. (2008), ele simplesmente é um mecanismo para a melhoria da qualidade do passo.

Uma vez que a estratégia de Mehrotra requer o cálculo de um passo afim escala (105), esta estratégia é mais cara do que a regra (104). Observamos que nas iterações em que o algoritmo região de confiança é chamado, por exemplo, quando a matriz Hessiana não é definida positiva, o parâmetro de barreira é calculado pela regra (104) ao invés da (105). Isso é feito porque a heurística de Mehrotra é cara para implementar no algoritmo de região de confiança, o qual utiliza uma iteração do gradiente conjugado.

#### **4.5 Estratégia (v): É utilizada uma regra Mehrotra tipo previsor-corretor para determinar o parâmetro de barreira em que o passo corretor é condicionado**

A quinta estratégia utilizada pelo *Knitro* é o uso do passo corretor de forma condicional, isto é, o passo corretor é utilizado apenas se não provocar um aumento na complementaridade por um fato maior do que 2.

Nocedal et al. (2008) destacam que a estratégia de MPC condicional é motivada pela observação de que os efeitos nocivos do passo corretor manifestam-se num aumento significativo da complementaridade. Nesta estratégia são utilizadas as equações (103) e (109) e como na estratégia (iv), quando o algoritmo recai sobre o algoritmo do Método de Região de Confiança, o parâmetro de barreira é calculado usando a regra (104), e o passo corretor não é usado.

#### 4.6 Estratégia (vi): Uma função qualidade é minimizada a cada iteração para determinar o parâmetro de barreira

As estratégias (iii), (iv) e (v) dependem dos parâmetros heurísticos (109) e (104). Apresentamos agora uma estratégia em que  $\mu$  é selecionado levando em consideração uma função de qualidade a qual deve ser minimizada. Assumimos que em (103),  $\sigma \geq 0$  é o parâmetro de centralização conhecido, e definimos  $\Delta(\sigma)$  como a solução das equações primais-duais (40) em função de  $\sigma$ . Sejam  $\alpha_z^{\max}(\sigma)$ ,  $\alpha_\lambda^{\max}(\sigma)$  o comprimento do passo que satisfazem (107) para o passo  $d = d(\sigma)$ , e são definidos os pontos de sondagem:

$$\begin{aligned} z(\sigma) &= z + \alpha_z^{\max}(\sigma)d_z(\sigma), \\ \lambda(\sigma) &= \lambda + \alpha_\lambda^{\max}(\sigma)d_\lambda(\sigma). \end{aligned}$$

O objetivo é escolher o valor de  $\sigma$  que proporcione uma melhora significativa na solução do problema de programação não-linear (36). Por exemplo, podemos escolher  $\sigma$  de modo a minimizar a seguinte função qualidade não linear com base no erro de KKT:

$$\begin{aligned} q_n(\sigma) &= \left\| \nabla f(x(\sigma)) - A_h(x(\sigma))^T \lambda_h(\sigma) + A_g(x(\sigma))^T \lambda_g(\sigma) \right\|^2 + \left\| c(z(\sigma)) \right\|^2 + \\ &+ \left\| S(\sigma) \Lambda_g(\sigma) e \right\|^2 \end{aligned} \quad (111)$$

A avaliação da  $q_N$  é cara computacionalmente, uma vez que requer a avaliação das funções dos problemas e de suas derivadas para cada valor de  $\sigma$ . Podemos evitar isso, usando uma função de qualidade linear. Se assumirmos que  $f$  e  $c$  são funções lineares, temos que (112) pode ser expressa como:

$$q_L(\sigma) = (1 - \alpha_\lambda^{\max}(\sigma))^2 \left\| \nabla f(x) - A_h(x)^T \lambda_h + A_g(x)^T \lambda_g \right\|^2 + (1 - \alpha_z^{\max}(\sigma))^2 \|c(x)\|^2 + (112)$$

$$+ \left\| (S + \alpha_s^{\max}(\sigma)d_s(\sigma))(\Lambda_g + \alpha_\lambda^{\max}(\sigma)d_\lambda(\sigma))e \right\|^2$$

em que  $d\Lambda_g(\sigma)$  é uma matriz diagonal com  $d\lambda(\sigma)$ , na diagonal, e similarmente para  $dS(\sigma)$ . Destacamos que a função  $q_L$  mede o erro de KKT exatamente nos pontos de sondagem  $(z(\sigma), \lambda(\sigma))$  para problemas de programação linear.

Note que  $d(\sigma) = d(0) + \sigma(d(1) - d(0))$ . Portanto,  $d(\sigma)$  pode ser facilmente calculado para qualquer valor de  $\sigma$ , uma vez que o sistema linear (40) foi resolvido por duas vezes para se obter  $d(0)$  e  $d(1)$ .

Tendo calculado  $d(\sigma)$ , o custo na avaliação de  $q_L$  consiste no cálculo do comprimento de passo máximo  $\alpha_z^{\max}(\sigma)$ ,  $\alpha_\lambda^{\max}(\sigma)$  e o último termo em (112), o que requer algumas operações vetoriais.

A função qualidade  $q_L$  é definida utilizando normas ao quadrado para penalizar alguns componentes grandes no erro de KKT. Note que  $q_L(\sigma)$  não é uma função convexa de  $\sigma$ , em geral. Além disso, devido à complicada dependência dos comprimentos de passos  $\alpha_z^{\max}(\sigma)$ ,  $\alpha_\lambda^{\max}(\sigma)$  sobre o parâmetro  $\sigma$ , não é possível obter uma expressão analítica para os minimizadores de  $q_L$ . No entanto, observou-se que, na prática, esta função é normalmente unimodal.

Portanto, é implementado um esquema de busca unidimensional para calcular um mínimo aproximado de  $q_L$ . Ele usa um procedimento denominado trisseção de ouro, como, pode ser visto em Luenberger (1984), e é ignorado o fato de que a  $q_L$  pode não ser necessariamente unimodal. Primeiramente é escolhido

$\sigma^{\min}$  e  $\sigma^{\max}$ , que definem um limite mínimo e máximo no valor  $\sigma$ , e definem os dois intervalos  $[\sigma^{\min}, 1]$  e  $[1, \sigma^{\max}]$ . Na aplicação do autor, o valor  $\sigma^{\min} = \max\left(\gamma, \mu^{\min} \frac{\lambda_g^T s}{n}\right)$ , em que é definido  $\mu^{\min} = 10^{-9}$  como um valor mínimo admissível do parâmetro de barreira,  $\gamma$  é um número pequeno, por exemplo,  $10^{-8}$  ou  $10^{-9}$  e  $\sigma^{\max} = 1000$ . Primeiramente é avaliada a função qualidade (112) para  $\sigma = 1$  e para alguns valores de  $\sigma$  ligeiramente inferiores a 1 (por exemplo 0,99). Se  $q_L(0,99) \leq q_L(1)$ , então, é feito o procedimento trisseção de ouro no intervalo de  $[\sigma^{\min}, 1]$ , caso contrário, pesquisamos no intervalo  $[1, \sigma^{\max}]$ . É importante que seja possível tomar valores maiores do que um para  $\sigma$ , de modo que o algoritmo possa recuperar-se de reduções excessivamente agressivas do parâmetro de barreira. O procedimento de trisseção termina quando 12 avaliações das funções qualidade são realizadas, ou se o tamanho de intervalo  $[a, b]$  torna-se menor do que  $b \times 10^{-2}$ .

As vantagens esperadas da abordagem da função qualidade são duas: primeiro, definimos um procedimento que associa a escolha do parâmetro de barreira a uma redução aceitável no erro KKT (linearizado). Portanto, esperamos que esta abordagem convirja em menos iterações comparadas com as abordagens previamente propostas com base em fórmulas heurísticas. Em segundo lugar, a escolha do parâmetro de barreira leva em conta a fração da fronteira dos comprimentos de passo em (107).

As funções (111) e (112) são baseadas nos passos após a aplicação da fração da regra de fronteira. Assim, as restrições implícitas,  $\lambda_g > 0$  e  $s > 0$ , são levadas em conta na escolha de  $\mu$ . Isto é semelhante às fórmulas de atualização Mehrotra e desencoraja escolhas do parâmetro de barreira que gera passos que rapidamente violem os limites  $\lambda_g > 0$  e  $s > 0$  e necessitam ser eliminados.

No próximo capítulo serão apresentados o histórico do problema de fluxo de potência e a sua formulação.

## **CAPÍTULO 5**

### **O PROBLEMA DE FLUXO DE POTÊNCIA ÓTIMO**

Neste capítulo apresentamos um histórico a respeito do problema de fluxo de potência ótimo, a definição do problema, as aplicações e sua formulação matemática.

#### **5.1 HISTÓRICO DO PROBLEMA DE FLUXO DE POTÊNCIA ÓTIMO**

O problema de fluxo de potência ótimo tem por objetivo determinar o melhor ponto de operação de um sistema elétrico de potência, contribuindo para a melhoria do desempenho deste, e ainda, satisfazendo às restrições regidas por particularidades operacionais e físicas da rede elétrica.

O problema de FPO não é simples devido à várias características que possui, sendo elas, a não linearidade, a presença de restrições canalizadas e de desigualdades, a não convexidade e por ser um problema de grande porte. Devido a essas características, vários trabalhos surgiram na tentativa de solucioná-lo, os quais utilizam diferentes metodologias.

O estudo do problema de FPO teve início na década de 60, mais precisamente em 1962, quando Carpentier propôs um modelo geral para o problema de FPO, incorporando as equações de fluxo de potência ao problema de Despacho Econômico (DE). Associou a este problema uma função Lagrangiana transformando-o em um problema de programação não linear irrestrito. Aplicou as condições necessárias de KKT e obteve um sistema não linear, resolvendo-o pelo método de Gauss-Seidel, encontrando, portanto, a solução do problema.

Após isso, muitos autores publicaram trabalhos na tentativa de solucionar o problema de FPO. Neste trabalho destacamos alguns deles.

Dommel & Tinney (1968) propuseram o Método do Gradiente Reduzido, o qual dividia as variáveis do problema em variáveis de controle e variáveis dependentes ou funcionais. Neste método, são atribuídos valores iniciais às variáveis de controle e as restrições de desigualdade são tratadas por parâmetros de penalidade.

As variáveis dependentes são calculadas através da resolução das restrições de igualdade do problema de FPO por meio do Método de Newton.

O Método da Função Penalidade foi introduzido por Sasson et al. (1973). Neste método são introduzidas funções que penalizam a função objetivo caso as restrições de igualdade ou desigualdade do problema sejam violadas, transformando o problema de FPO em um problema irrestrito. O parâmetro de penalidade é aumentado a cada iteração, um problema irrestrito é resolvido e este processo é repetido até que todas as restrições do problema sejam resolvidas satisfatoriamente.

Rashed e Kelly (1974), introduziram um método que utilizava o Método de Newton, o Método de Penalidade e dos multiplicadores de Lagrange. Neste método são incorporadas à função objetivo, as restrições de igualdade por meio de multiplicadores de Lagrange e as restrições de desigualdade por meio de fatores de penalidade, resultando em uma função auxiliar. As atualizações dos fatores de penalidade são feitas por um parâmetro positivo e das variáveis pelo Método de Newton, no qual o passo é calculado por meio da matriz Hessiana.

Sun et al. (1984) propuseram o Método de Newton com Penalidade. Seguindo a mesma linha de Rashed e Kelly (1974), esta abordagem para solucionar o problema de FPO, tem como base a formulação de Newton e o desacoplamento do problema original em dois subproblemas, denominados subproblema de potência ativa e subproblema de potência reativa. Após incorporar à função objetivo as restrições de igualdade, por meio de multiplicadores de Lagrange, e as restrições de desigualdade, por meio de multiplicadores de Lagrange e fatores de penalidade, origina-se uma função auxiliar. Essas restrições são divididas em duas classes: a classe das restrições penalizadas e a classe das restrições consideradas ativas na solução, sendo estas incorporadas na função objetivo por meio de multiplicadores de Lagrange. Ao satisfazer as condições de KKT, o ponto ótimo do problema é encontrado e as equações do fluxo de potência tradicional, são satisfeitas dentro de uma determinada tolerância.

Entre as diversas técnicas aplicadas na resolução do problema de FPO, o Método de Pontos Interiores tem sido considerado um dos mais eficientes devido ao seu ótimo desempenho e robustez para a solução de problemas de programação não linear.

Para solucionar o problema de Despacho Ótimo de Reativos, Granville (1994) utilizou um Método de Pontos Interiores denominado Método Primal-Dual

Barreira Logarítmica, o qual incorpora variáveis de folga e/ou excesso nas restrições de desigualdade, transformando-as em restrições de igualdade. É adicionada uma função barreira logarítmica e um parâmetro de barreira à função objetivo garantindo que as variáveis de folga e/ou excesso permaneçam positivas. O parâmetro de barreira tende a zero e é atualizado a cada iteração. Uma dificuldade encontrada neste método é a escolha do valor inicial do parâmetro de barreira, pois é um dado fundamental para o sucesso do método.

Wu et al. (1994) sugerem um método denominado Método Previsor-Corretor que diferencia-se do Método Primal-Dual, pois são introduzidos termos não lineares nas condições de otimalidade.

Torres & Quintana (1998) propuseram a resolução do problema de FPO por meio do Método de Pontos Interiores usando coordenadas retangulares de tensão, que dessa forma, a função objetivo e as restrições são quadráticas. Características de funções quadráticas, como por exemplo, a matriz Hessiana constante e a expansão em série de Taylor até a segunda ordem sem erro de truncamento, facilitam a incorporação de informações de alta ordem em um procedimento previsor-corretor, o que proporciona uma melhora no desempenho do Método de Pontos Interiores.

Momoh et al. (1999a) e Momoh et al. (1999b) apresentaram uma revisão a respeito de técnicas estudadas e usadas para resolução do problema de FPO existentes até o ano de 1993. De acordo com os autores, essas técnicas podem ser classificadas em seis classes: programação não linear; solução das condições de otimalidade baseadas em Newton; programação linear; versão híbrida de programação inteira e Métodos de Pontos Interiores.

Nejdawi et al. (2000) propuseram um algoritmo de programação quadrática, sendo este formado por um ciclo de linearização externo e um ciclo de otimização interno. O ciclo de otimização interno resolve um problema de programação quadrática reduzido e com restrições relaxadas, mantendo o problema com uma dimensão pequena. As iterações do ciclo externo são comparadas ao FPO de Newton e as do ciclo interno, como eficientes iterações de pontos interiores. Os autores observaram, através de testes computacionais para os sistemas IEEE de 30, 57, 118 e 300 barras, que os números de iterações internas e externas para as diferentes dimensões do problema não sofreram grandes diferenças e este algoritmo é considerado muito eficiente pelos autores.

Costa et al. (2000), por meio de um estudo comparativo, apresentaram três abordagens, baseadas no Método de Newton, para solucionar o problema de FPO. Sendo elas: penalidade e conjunto ativo; primal-dual; e barreira logarítmica primal-dual. Fizeram uma revisão das metodologias e compararam suas principais características, apresentado testes computacionais para os sistemas IEEE 30, 57 e 118 barras.

Quintana et al. (2000) analisaram e classificaram as principais publicações em Método de Pontos Interiores, em relação ao desenvolvimento matemático e implementação dos algoritmos que obtiveram sucesso nas aplicações em operações do sistema de energia. Os autores também analisaram os mais conhecidos códigos de software em pontos interiores, os quais, a maioria estão disponíveis gratuitamente na Internet.

Oliveira & Filho (2003), propuseram o Método de Pontos Interiores Primal-Dual e Previsor-Corretor para um modelo de FPO DC, onde um problema de fluxo em redes com restrições adicionais são representadas pelas leis de *Kirchoff*. A estrutura da matriz resultante é explorada reduzindo o sistema linear a ser resolvido a um sistema com dimensão igual ao número de barras ou ao número de laços independentes, cuja matriz é invariante no decorrer do processo, permitindo que o método tenha uma iteração de velocidade maior. Conseqüentemente, um sistema linear deve ser resolvido, cuja matriz varia a cada iteração e a dimensão deste sistema se reduz ao número de geradores. O autor apresenta resultados numéricos para os sistemas testes do IEEE e sistemas brasileiros de grande porte. Para esses casos, o Método de Pontos Interiores se mostra bastante eficiente e robusto tendo uma convergência rápida.

Almeida (2005) analisa os casos de difícil resolução do problema de FPO a partir de um problema de FPO parametrizado e de suas condições de otimalidade, que são primeiramente derivadas supondo a formulação original do problema. Esta formulação contém restrições de igualdade e de desigualdade, que posteriormente são particularizadas à formulação usada nos Métodos Primais-Duais de Pontos Interiores, utilizando um equivalente do Sistema Sul Brasileiro para ilustrar tais casos críticos.

Carvalho (2005) utiliza os Métodos de Pontos Interiores Primal-Dual e previsor corretor desenvolvendo-os para o problema de minimização de custos na geração e perdas na transmissão do pré-despacho DC (fluxo de carga em corrente contínua) de um sistema de potência hidroelétrico, com base no modelo de fluxo em

redes e no princípio do mínimo esforço. É estudado em seu trabalho a estrutura matricial resultante da simplificação do problema proposto pela inclusão do princípio do mínimo esforço.

Oliveira (2005) propõe um Método de Pontos Interiores Primal-Dual e Previsor-Corretor para minimizar os custos na geração e perdas na transmissão no problema de pré-despacho de um sistema de potência hidroelétrico, baseando-se no modelo de fluxo em redes e no princípio do mínimo esforço. O autor também propõe um Método Primal-Dual perturbado com base na definição de uma heurística que determinará a escolha do parâmetro de perturbação, o qual mostrou-se eficiente na prática convergindo em menos iterações comparado com uma implementação já existente do modelo de fluxo de redes.

Baptista et al. (2006) apresentaram uma nova abordagem para minimizar as perdas em sistemas elétricos de potência, a qual considera a aplicação do Método Primal-Dual Barreira Logarítmica à magnitude da tensão e variáveis transformadores *taps*. As restrições de desigualdade são tratadas pelo Método da Lagrangiana aumentada. A função Lagrangiana é associada a todas as restrições e as condições necessárias de primeira ordem são obtidas através do método de Newton, da atualização das variáveis duais e fatores de penalidade. Os resultados dos testes são apresentados para mostrar o bom desempenho desta abordagem.

Thomaz (2007) apresenta em seu trabalho um Método de Pontos Interiores Primal-Dual desenvolvido para o problema de FPO corrente alternada ativo e reativo. O autor adotou a representação das tensões através de coordenadas cartesianas uma vez que, neste modelo, a matriz Hessiana do problema é constante e a expansão até segunda ordem em Taylor é exata. Antes da aplicação do método, reduz-se o número de variáveis do problema não alterando a estrutura esparsa do mesmo. A matriz resultante possui característica simétrica, a qual é explorada de forma eficiente reduzindo o esforço computacional do método por iteração. São apresentados resultados computacionais, utilizando sistemas de teste IEEE e um sistema real brasileiro.

Lage et al. (2008) apresentam um estudo sobre o Despacho Ótimo ativo e reativo com restrições de fluxo de linha. O Método Primal-Dual Barreira Logarítmica foi utilizado para resolver o problema. Os resultados obtidos nos sistemas de 8 e 118 barras, mostram que a técnica proposta é capaz de melhorar o funcionamento do sistema.

Lage et al. (2009) apresentam um Método Barreira/Penalidade Modificada. Neste método, as restrições de desigualdade são transformadas em restrições de igualdades pela introdução de variáveis de folga, que são tratadas por uma função barreira modificada ou a função penalidade quadrática. Em seguida, as condições de primeira ordem e método de Newton são aplicados na solução do problema, e para validar a proposta do método, sistemas elétricos de potência de 3, 14, 30, 118, 162 e 300 barras foram usados como estudos de caso, e os resultados obtidos mostraram a eficiência do método.

Song et al. (2009) aplicam um Método Ponto Interior não-linear para FPO reativo usando funções penalidade de arredondamento. A função penalidade aplicada é na forma de cosseno e é incluída para se harmonizar as funções barreira com as variáveis de controle discretas em problemas de otimização de sistemas de potência. Os resultados dos testes com sistemas de teste 14 barras mostram a aplicabilidade do algoritmo proposto.

Sousa et al. (2009) propõem uma nova abordagem chamada de Função Lagrangiana Barreira Modificada para resolver o problema de Fluxo de Potência Ótimo. Nesta abordagem, as restrições de desigualdade são tratadas pelo Método da Função Barreira Modificada. A viabilidade da abordagem proposta é demonstrada com testes comparativos com o Método de Ponto Interior, utilizando vários sistemas de teste IEEE e duas redes derivadas da geração brasileira / sistema de transmissão. Os resultados mostram que o método Função Lagrangiana Barreira Modificada é computacionalmente mais atraente do que o Método de Pontos Interiores em termos de velocidade, número de iterações e condicionamento numérico.

Probst (2010) utiliza um método de pontos interiores previsor-corretor, onde as tensões são representadas em coordenadas cartesianas ao invés de coordenadas polares, pois estas, sendo quadráticas, permitem correções não lineares nas condições de factibilidade primal e dual e não apenas nas de complementaridade como nos métodos tradicionais de programação não linear.

Khiat e Marano (2010) apresentam um estudo de um Método de Pontos Interiores com base no Algoritmo Primal Dual Barreira Logarítmica para programação não-linear. A minimização de perda total de potência ativa é formulada como um FPO Reativo, incluindo restrições de igualdade e de desigualdade que representam as condições de segurança do sistema de energia. Mostra-se que este método de otimização tem características muito atraentes para resolver essa classe de

problemas e resultados obtidos com várias simulações para um verdadeiro sistema de transmissão de energia são apresentados.

Chung et al. (2011) propõem um Método de Ponto Interior previsor-corretor decomposto para resolver o problema de fluxo de potência dinâmica, que é um problema de otimização não linear de grande porte. O sistema KKT é decomposto em vários subsistemas com base na sua estrutura especial de bloco, em que o tamanho de cada um dos subsistemas depende somente do tamanho da rede. No processo iterativo, variáveis de folga e multiplicadores de Lagrange de restrições dinâmicas são previstos primeiramente e corrigidos, e então, outras variáveis são previstas e corrigidas em cada intervalo de tempo. Os parâmetros, tais como o comprimento do passo e parâmetro de barreira, são independentemente estimados em cada subsistema. Além disso, uma estratégia de iteração de desigualdade é introduzida para evitar a computação desnecessária. A eficácia do método proposto foi demonstrada nos sistemas IEEE 14 e 118 barras com até 24 intervalos de tempo. Verificou-se que, em comparação com um Método de Ponto Interior Primal Dual Puro, o Método de Ponto Interior Previsor-Corretor Decomposto proposto é mais atrativo, especialmente quando as pressões dinâmicas tornam-se ativas.

Soler et al. (2012) apresentam um método para lidar com as variáveis discretas do problema de fluxo de potência ótimo. A função de penalidade é apresentada e devido à sua inclusão para a função objetivo, uma sequência de problemas de programação não linear com apenas as variáveis contínuas são obtidos e as soluções destes problemas convergem para uma solução do problema misto. Os problemas de programação não-lineares obtidos são resolvidos por um Método Primal-Dual Barreira Logarítmica e testes numéricos são apresentados para os sistemas IEEE 14, 30, 118 e 300 indicando que o método é eficiente.

Pinheiro (2012) propôs em seu trabalho o Método Previsor-Corretor Primal-Dual de Pontos Interiores, com Barreira Logarítmica Modificada com estratégia de ajuste cúbico e o Método Previsor-Corretor Primal-Dual de Pontos Interiores, com Barreira Logarítmica Modificada com estratégias de ajuste cúbico e de convergência global. O autor apresentou resultados obtidos para os sistemas elétricos de Dommel e Tinney (1968) de 9 barras, IEEE 14, 30, 57, 118 e 162 barras, e os comparou entre os dois métodos propostos.

## **5.2 DEFINIÇÃO DO PROBLEMA DE FPO**

O problema FPO é uma ferramenta que tem por finalidade fornecer a melhor condição de operação de um sistema elétrico sob um determinado objetivo. O objetivo pode ser, por exemplo, a condição de operação tal que o sistema tenha a menor quantidade de perdas elétricas possível. O problema de FPO não é simples, pois a solução encontrada deve respeitar os limites operativos dos equipamentos que compõem a rede elétrica assim como atender outras restrições inerentes à operação de um sistema elétrico.

### **5.2.1 APLICAÇÕES DO FLUXO DE POTÊNCIA ÓTIMO**

O FPO tem aplicação em vários problemas de planejamento da expansão e operação e de operação em tempo-real, tais como:

- Despacho econômico e seguro (operação em tempo-real, simulação do despacho em estudos de planejamento da operação e expansão);
- Redespacho preventivo e corretivo (operação em tempo-real);
- Minimização de perdas;
- Alocação de fontes de potência reativa (planejamento da expansão do suporte de reativos);
- Avaliação da confiabilidade composta de sistemas geração e transmissão;
- Planejamento da expansão de sistemas de transmissão;
- Tarifação de serviços de transmissão;
- Determinação de preços nodais de energia.

O FPO é um termo genérico que corresponde a uma ampla classe de problemas dentre os quais se destaca o problema de FPO reativo, que é o modelo de problema de FPO utilizado neste trabalho.

Na próxima seção apresentamos a formulação matemática do problema de FPO reativo.

## 5.2.2 FORMULAÇÃO MATEMÁTICA DO PROBLEMA DE FPO

O FPO é geralmente formulado como um problema de Programação Não Linear, de acordo com o seguinte formato padrão:

$$\begin{aligned} \min & f(x) \\ \text{s.a: } & g(x) \\ & h(x) \leq 0 \\ & \bar{x} \leq x \leq \underline{x} \end{aligned} \tag{113}$$

Em que:

- $f(x)$  : função objetivo;
- $g(x) = 0 \in \mathcal{R}^m$  : restrições de igualdade;
- $h(x) \leq 0 \in \mathcal{R}^p$  : restrições de desigualdade;
- $x = (V, \theta) \in \mathcal{R}^n$  : vetor de variáveis do problema;
- $\underline{x}$  e  $\bar{x}$  : vetor dos limites inferiores e superiores das variáveis  $x$ , respectivamente.

O vetor  $x$  das variáveis de controle e de estado, representa o vetor de magnitude das tensões ( $V$ ), os ângulos das tensões ( $\theta$ ). A função objetivo,  $f(x)$  pode ser as perdas de potência ativa na transmissão, o custo na geração, entre outras. Neste trabalho, adota-se as perdas de potência ativa na transmissão.

As restrições de igualdade  $g(x)$  são as equações do fluxo de potência ativas e reativas obtidas impondo-se as Leis de *Kirchoff* em cada barra da rede. As restrições de desigualdade  $h(x)$  representam as restrições funcionais como: limites sobre a potência reativa nas barras de controle de reativos e sobre a potência ativa na barra *slack*, limites nos fluxos ativos e reativos nas linhas de transmissão, entre outras.

As variáveis do problema de FPO podem ser caracterizadas como:

- ❖ Variáveis dependentes:
  - tensão nas barras de carga do sistema;
  - ângulo em todas as barras do sistema menos a de referência angular;

- potência reativa nas barras com controle de reativo e referência;
- potência ativa na barra de referência.

❖ Variáveis de controle:

- tensão nas barras de controle de reativos do sistema;
- *tap* dos transformadores;
- bancos de capacitores.

Segundo Monticelli (1983), o problema de FPO é escrito com a utilização das equações do fluxo de potência, de uma função objetivo e das restrições, obtendo-se o seguinte problema de FPO reativo:

$$\begin{aligned}
 & \min \sum_{(k,m) \in NL} f_{km}(V, \theta) \\
 & s.a: \Delta P_i(V, \theta) = 0, \quad i = 1, 2, \dots, \text{NBCCR} \\
 & \quad \Delta Q_j(V, \theta) = 0, \quad j = 1, 2, \dots, \text{NBC} \\
 & \quad \underline{Q}_n \leq Q_n(V, \theta) \leq \overline{Q}_n, \quad n = 1, 2, \dots, \text{NBCR} \\
 & \quad \underline{V}_p \leq V_p(V, \theta) \leq \overline{V}_p, \quad p = 1, 2, \dots, \text{NB}
 \end{aligned} \tag{114}$$

em que:

- NBCCR é o número de barras de carga e de controle de reativos;
- NBC é o número de barra de cargas;
- NBCR é o número de barras de controle de reativos;
- NB é o número de barras do sistema elétrico;
- NL é o número total de linhas de transmissão;
- $V$  e  $\theta$  são os vetores da magnitude e fase de tensão, respectivamente;
- $f_{km}(V, \theta)$ , neste trabalho, é a função de perdas de potência ativa na transmissão na linha  $km$ ,

$$\text{em que: } f_{km}(V, \theta) = \sum_{(k,m) \in NL} g_{km} (V_k^2 + V_m^2 - 2V_k V_m \cos \theta_{km});$$

- As equações de balanço do sistema elétrico são dadas por:
  - ✓ Potência ativa para as barras de carga e controle reativo:

$$\Delta P_k(V, \theta) = P_k^G - P_k^C - \sum_{m \in \Omega} P_{km}(V, \theta),$$

em que:  $P_{km}(V, \theta) = V_k^2 g_{km} - V_k V_m (g_{km} \cos \theta_{km} + b_{km} \text{sen} \theta_{km})$ ;

✓ Potência reativa para as barras de carga:

$$\Delta Q_k(V, \theta) = Q_k^G - Q_k^C - \sum_{m \in \Omega} Q_{km}(V, \theta);$$

em que:  $Q_{km}(V, \theta) = -V_k^2 (b_{km} + b_{km}^{sh}) + V_k V_m (b_{km} \cos \theta_{km} - g_{km} \text{sen} \theta_{km})$ ;

• Limite na geração de potência reativa para as barras de controle reativo:

$$Q_{km}(V, \theta) = \sum_{m \in \Omega} -(V_k)^2 (b_{km} + b_{km}^{sh}) + V_k V_m (b_{km} \cos \theta_{km} - g_{km} \text{sen} \theta_{km});$$

•  $g_{km}, b_{km}$  e  $b_{km}^{sh}$ , são a condutância, susceptância e a susceptância *shunt* da linha, respectivamente;

•  $P_k^G$  e  $P_k^C$  são as potências ativas geradas e potência ativas consumidas, respectivamente;

•  $Q_k^G$  e  $Q_k^C$  são as potências reativas geradas e potências reativas consumidas, respectivamente;

•  $\overline{Q}_k$  e  $\underline{Q}_k$  são os limites máximos e mínimos de geração de potência reativa, respectivamente;

•  $\overline{V}_k$  e  $\underline{V}_k$  são os limites máximos e mínimos das magnitudes das tensões, respectivamente;

•  $\Omega$  é o conjunto das barras vizinhas à barra  $k$ , incluindo ela mesma.

Destacamos que a variável de controle *tap* de transformador foi considerada de forma fixa neste trabalho.

O problema a ser estudado neste trabalho é o problema de FPO reativo. O aspecto reativo em um sistema de energia está relacionado principalmente ao controle da tensão nas barras. No problema de FPO reativo, as variáveis associadas com a potência ativa estão fixas e as variáveis de controle estão associadas com a potência reativa. A função objetivo em (114) relaciona-se às perdas ativa nas linhas de transmissão, essa função é não separável e não permite simplificações. De acordo com Monticelli e Liu (1992), esse fato pode dificultar a solução do problema de FPO.

Existem muitas metodologias de otimização utilizadas para solução do problema (114). Dentre elas destacamos a metodologia de Pontos Interiores.

No próximo capítulo serão apresentados os resultados numéricos da aplicação do pacote de algoritmos de otimização não linear, denominado *Knitro*, no qual está implementado um Método de Ponto Interior que utiliza Busca Linear ou Região de Confiança para o cálculo do passo, aplicados a um problema matemático teste e a vários sistemas elétricos de potência encontrados na literatura.

## CAPÍTULO 6

### RESULTADOS NUMÉRICOS

Neste capítulo, apresentamos os resultados obtidos por meio da utilização do pacote de algoritmos *Knitro*, apresentado no capítulo 2, a um problema matemático proposto por Bazaraa et al. (1993) e adaptado por Ferreira et al. (2012), e ao problema de FPO. Os sistemas elétricos de potência usados foram IEEE 14, 30 e 118 barras. Os dados destes sistemas foram obtidos na página <http://www.ee.washington.edu/research/pstca/> (acessado em janeiro de 2013). O pacote *Knitro* foi utilizado com interface do GAMS – *General Algebraic Modeling System* ([www.gams.com](http://www.gams.com)) para resolver o problema matemático teste e o problema de FPO. A versão *free* do software está disponível na internet, podendo ser utilizada apenas para problemas com poucas variáveis, no entanto, para realização completa de todos os testes deste trabalho, devido a sua dimensão, foi necessário a utilização da versão, *Knitro* 6.0, cuja licença foi cedida pelo Laboratório de Análise de Sistemas de Energia Elétrica (LASEE) da Escola de Engenharia de São Carlos - USP.

A implementação computacional para a entrada de dados do problema de FPO foi realizada utilizando o software Matlab R2010a (<http://www.mathworks.com>) em um microcomputador com processador Intel i5 M430 de 2.27GHz, 4 Gb de memória RAM do Laboratório de Otimização e Estudos Econômicos de Sistemas de Potência (LOESP), da Faculdade de Engenharia, da Unesp de Bauru.

Todos os testes foram realizados em um microcomputador com processador Intel i3 de 3.20 GHz, com 4 GB de memória RAM.

No *Knitro* alguns valores iniciais são pré-definidos e outros podem ser definidos pelo usuário. Em todos os testes o valor inicial para o parâmetro de barreira é escolhido manualmente por  $\mu = 10^{-1}$ , já as tolerâncias dos testes de parada para o problema de programação não linear e para o problema de barreira são  $\varepsilon^{opt} = 10^{-6}$  e  $\varepsilon^{feas} = 10^{-6}$ , respectivamente, são os valores padrões do pacote de algoritmos *Knitro*.

## 6.1 Testes numéricos

Nesta seção apresentam-se os testes numéricos e os resultados obtidos.

### 6.1.1 Problema matemático teste

Seja o problema matemático teste adaptado por Ferreira et al. (2012):

$$\min (x_1 - 2)^4 + (x_1 - 2x_2)^2 \quad (115)$$

$$s.a : \text{sen}x_1 - x_1 + 2 - x_2 = 0 \quad (116)$$

$$x_1^2 - x_2 \leq 0 \quad (117)$$

$$x_2 - \text{sen}x_1 - 0,5 \geq 0 \quad (118)$$

$$x_2 - \text{sen}x_1 - 2,5 \leq 0 \quad (119)$$

A representação geométrica das curvas de níveis da função objetivo (115), das restrições do problema (116) – (119) e o ponto ótimo do problema é dada pela Figura 3.

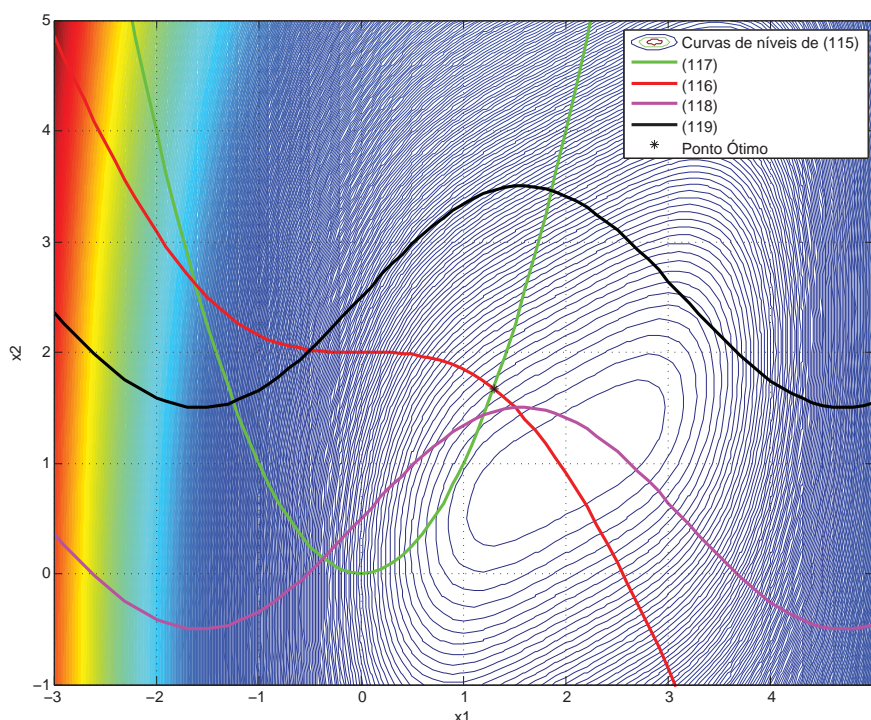


Figura 3 - Representação geométrica do problema (115) – (119).

A solução inicial utilizada é  $x_0 = (0, 2)^T$ ,  $\lambda_0$  é calculado por (74) – (76) e  $s$  é escolhido como um valor positivo.

Para resolver o problema (115) – (119), propõe-se resolver o seguinte problema de barreira, cujas variáveis de folga  $s_j > 0$ :

$$\begin{aligned} \min \varphi_\mu(x_1, x_2) &= (x_1 - 2)^4 + (x_1 - 2x_2)^2 - \mu \sum_{j=1}^3 \ln(s_j) \\ \text{s.a:} \quad \text{sen}x_1 - x_1 + 2 - x_2 &= 0 \\ x_1^2 - x_2 + s_1 &= 0 \\ -x_2 + \text{sen}x_1 + 0,5 - s_2 &= 0 \\ x_2 - \text{sen}x_1 - 2,5 + s_3 &= 0 \end{aligned} \quad (120)$$

Associa-se ao problema (120) uma função Lagrangiana, dada por:

$$\begin{aligned} L(x_1, x_2, s_1, s_2, s_3, \lambda_0, \lambda_1, \lambda_2, \lambda_3) &= (x_1 - 2)^4 + (x_1 - 2x_2)^2 + \lambda_0(\text{sen}x_1 - x_1 + 2 - x_2) + \\ &+ \lambda_1(x_1^2 - x_2 + s_1) + \lambda_2(-x_2 + \text{sen}x_1 + 0,5 + s_2) + \lambda_3(x_2 - \text{sen}x_1 - 2,5 + s_3) - \\ &- \mu \ln(s_1) - \mu \ln(s_2) - \mu \ln(s_3) \end{aligned} \quad (121)$$

Aplicam-se as condições de otimalidade à função Lagrangiana (121), e obtém-se o seguinte sistema de equações não lineares:

$$\begin{aligned} 2x_1 - 4x_2 + \lambda_0 \cos x_1 - 1 + 4(x_1 - 2)^3 + 2\lambda_1 x_1 + \lambda_2 \cos x_1 - \lambda_3 \cos x_1 &= 0 \\ \lambda_3 - \lambda_1 - \lambda_2 - \lambda_0 - 4x_1 + 8x_2 &= 0 \\ \lambda_1 - \frac{\mu}{s_1} &= 0 \\ \lambda_2 - \frac{\mu}{s_2} &= 0 \\ \lambda_3 - \frac{\mu}{s_3} &= 0 \\ \text{sen}x_1 - x_2 - x_1 + 2 &= 0 \\ x_1^2 + s_1 - x_2 &= 0 \\ s_2 - x_2 + \text{sen}x_1 + \frac{1}{2} &= 0 \\ s_3 + x_2 - \text{sen}x_1 - \frac{5}{2} &= 0 \end{aligned} \quad (122)$$

O sistema (122) é resolvido pelo método de Newton, e o vetor gradiente e a matriz Hessiana associados à (122), utilizados no sistema de Newton, são dados por:

$$\begin{bmatrix} 2\lambda_1 + 12(x_1 - 2)^2 - \lambda_0 \text{sen} x_1 - & -4 & 0 & 0 & 0 & \cos x_1 - 1 & 2x_1 & \cos x_1 & -\cos x_1 \\ -\lambda_2 \text{sen} x_1 + \lambda_3 \text{sen} x_1 + 2 & & & & & & & & \\ -4 & 8 & 0 & 0 & 0 & -1 & -1 & -1 & 1 \\ 0 & 0 & \frac{\mu}{s_1^2} & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{\mu}{s_2^2} & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{\mu}{s_3^2} & 0 & 0 & 0 & 1 \\ \cos x_1 - 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2x_1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos x_1 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\cos x_1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

A Tabela 1 apresenta os resultados para as diferentes estratégias de atualização do parâmetro de barreira, disponíveis no pacote de algoritmos de otimização *Knitro* e destacadas no capítulo 4. São apresentados os valores finais obtidos para as variáveis  $x_1$  e  $x_2$ , para a função objetivo e a quantidade de iterações realizados em cada estratégia.

**Tabela 1** – Resultado das diferentes estratégias de atualização do parâmetro de barreira disponível no *Knitro* para o problema teste matemático

| Estratégias    | $x_1$      | $x_2$      | Função Objetivo  | Total de Iterações |
|----------------|------------|------------|------------------|--------------------|
| (i)            | 1,29203728 | 1,66936036 | 4,44012572235106 | 7                  |
| (ii)           | 1,29203724 | 1,66936040 | 4,44012578403590 | 5                  |
| (iii)          | 1,29203728 | 1,66936035 | 4,44012571235210 | 6                  |
| (iv)           | 1,29203733 | 1,66936037 | 4,44012571235205 | 5                  |
| (v)            | 1,29203726 | 1,66936037 | 4,44012565770918 | 5                  |
| (vi)           | 1,29203701 | 1,66936055 | 4,44012572206548 | 5                  |
| <i>default</i> | 1,29203728 | 1,66936035 | 4,44012571235205 | 5                  |

Na Tabela 2 apresentam-se os resultados das estatísticas finais que mostram o número de iterações feitas utilizando o Método Região de Confiança com passo CG, número de avaliações da função objetivo, o número de avaliações do gradiente, o número de avaliações da Hessiana, o tempo total de processamento em segundos e o tempo total gasto nas avaliações também em segundos, para cada estratégia de atualização do problema de barreira disponível no pacote de algoritmos de otimização *Knitro*.

**Tabela 2** – Estatísticas finais para o problema matemático teste.

| <b>Estratégias</b>                   | <b>(i)</b> | <b>(ii)</b> | <b>(iii)</b> | <b>(iv)</b> | <b>(v)</b> | <b>(vi)</b> | <b>default</b> |
|--------------------------------------|------------|-------------|--------------|-------------|------------|-------------|----------------|
| <b>Número de iterações CG</b>        | 0          | 0           | 0            | 0           | 1          | 0           | 0              |
| <b>Avaliações da função objetivo</b> | 8          | 6           | 7            | 6           | 10         | 6           | 6              |
| <b>Avaliações do gradiente</b>       | 8          | 6           | 7            | 6           | 6          | 6           | 6              |
| <b>Avaliações da Hessiana</b>        | 7          | 5           | 6            | 5           | 5          | 5           | 5              |
| <b>Tempo total</b>                   | 0,076      | 0,051       | 0,074        | 0,084       | 0,057      | 0,081       | 0,039          |
| <b>Tempo gasto nas avaliações</b>    | 0,001      | 0,000       | 0,000        | 0,001       | 0,000      | 0,000       | 0,001          |

Analisando os resultados obtidos na Tabela 1, nota-se que o processo de otimização utilizando a estratégia (i) convergiu em sete iterações, sendo este o maior número obtido em comparação com as demais estratégias. Já as estratégias (ii), (iv), (v) e (vi) obtiveram um total de cinco iterações, sendo esse o menor número comparando com as outras estratégias. E a estratégia (iii) obteve seis iterações. O melhor valor para a função objetivo foi obtido com a estratégia (v) com diferença na sétima casa decimal.

Observamos na Tabela 2 que apenas a estratégia (v) realizou 01 iteração CG. Quanto ao número de avaliações da função objetivo as estratégias (ii), (iv) e (vi) obtiveram o menor número e a estratégia (v) o maior número. As estratégias (ii), (iv), (v) e (vi) obtiveram o menor número de avaliações do gradiente e avaliações da Hessiana, e a estratégia (i) o maior número. Comparando o tempo total de processamento, a estratégia que resolveu o problema mais rápido foi a estratégia (ii) e a que utilizou mais tempo para resolução foi a estratégia (iv). As estratégias (i) e (iv) gastaram 0,001 segundo de tempo nas avaliações.

## 6.1.2 Sistemas Elétricos de Potência

Em todos os testes realizados com os sistemas elétricos de 14, 30 e 118 barras, a função objetivo a ser minimizada foi a de perdas de potência ativa nas linhas de transmissão, tendo como restrições de igualdade as equações do fluxo de potência, como variáveis canalizadas as magnitudes de tensão e as restrições canalizadas de geração de reativos nas barras de controle de reativo. Em todos os sistemas elétricos utilizados a potência base foi de 100 MVA.

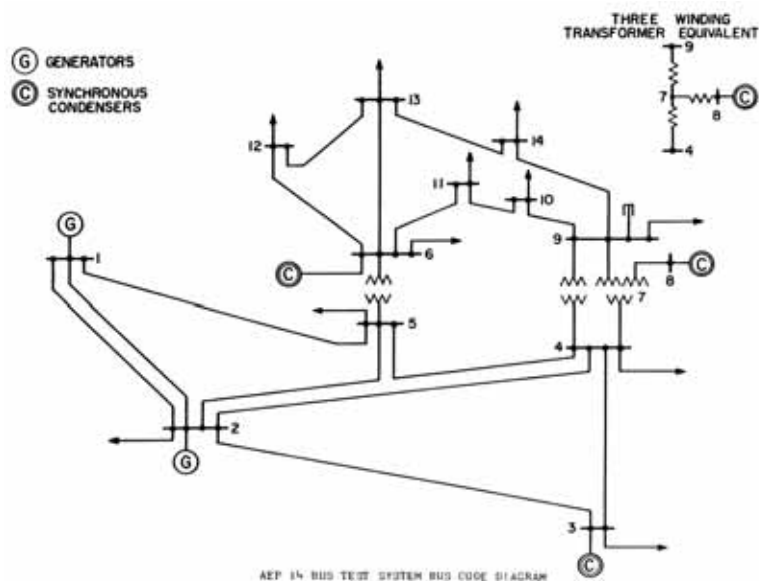
Nos testes realizados com os sistemas elétricos de 14 e 30 barras, os limites mínimos e máximos das magnitudes de tensão das barras do sistema foram  $\underline{V}_p = 0,95 \text{ p.u}$  e  $\overline{V}_p = 1,10 \text{ p.u}$ , respectivamente, e para o sistema elétrico 118 barras, os limites mínimos e máximos das magnitudes de tensão das barras do sistema elétrico foram  $\underline{V}_p = 0,90 \text{ p.u}$  e  $\overline{V}_p = 1,10 \text{ p.u}$ , respectivamente. Nos testes realizados com os sistemas IEEE 14, 30 e 118 barras os valores dos *taps* são fixos e os valores iniciais para as tensões e ângulos são de acordo com os dados destes sistemas, os quais foram obtidos na página <http://www.ee.washington.edu/research/pstca/> (acessado em janeiro de 2013).

### 6.1.2.1 Sistema elétrico IEEE 14 barras

O sistema IEEE 14 barras possui as seguintes características:

- 1 barra de geração (*slack*);
- 4 barras de controle reativo;
- 9 barras de carga;
- 17 linhas de transmissão;
- 3 transformadores com *tap* fixo.

O diagrama do sistema elétrico IEEE 14 barras é apresentado na Figura 4.



**Figura 4** - Sistema elétrico IEEE 14 barras.

Fonte: <http://www.ee.washington.edu/research/pstca/pf14/14bus600.tif>

Para o sistema IEEE 14 barras a formulação matemática do FPO contempla 22 restrições de igualdade, 42 restrições de desigualdade e 27 variáveis.

A Tabela 3 apresenta os resultados das Perdas (MW) e a quantidade de iterações para as diferentes estratégias de atualização do parâmetro de barreira, presentes no pacote de algoritmos de otimização *Knitro*.

**Tabela 3** – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 14 barras

| Estratégias    | Perdas (MW)      | Total de Iterações |
|----------------|------------------|--------------------|
| (i)            | 12,3615894863511 | 12                 |
| (ii)           | 12,3614451559757 | 10                 |
| (iii)          | 12,3614690886585 | 8                  |
| (iv)           | 12,3614758449893 | 7                  |
| (v)            | 12,3614726719065 | 6                  |
| (vi)           | 12,3614747401626 | 8                  |
| <i>default</i> | 12,3614758449893 | 7                  |

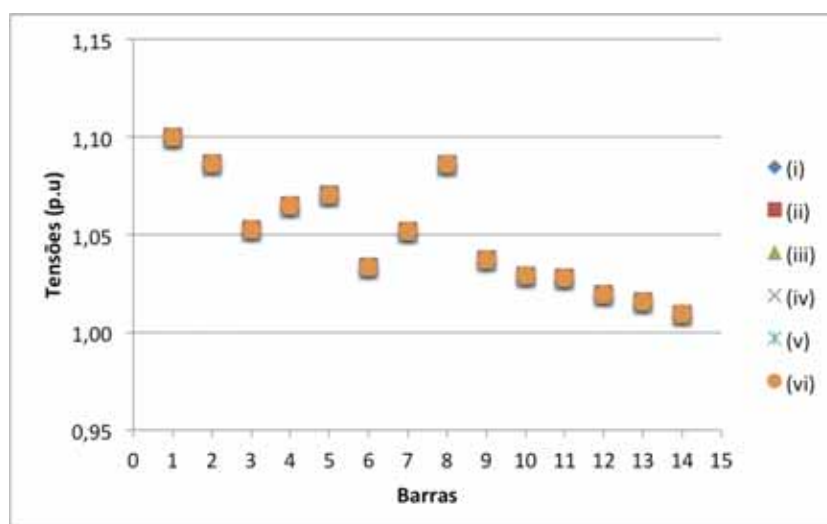
A Tabela 4 apresenta as estatísticas finais que mostra o número de iterações realizadas utilizando o Método Região de Confiança com passo CG, o número de avaliações da função objetivo, o número de avaliações do gradiente, o número de avaliações da Hessiana, o tempo total de processamento em segundos e tempo total gasto nas avaliações também em segundos, para cada estratégia de atualização do parâmetro de barreira disponível no pacote de algoritmos de otimização *Knitro*.

**Tabela 4** – Estatísticas finais para o sistema elétrico IEEE 14 barras.

| <b>Estratégias</b>                   | <b>(i)</b> | <b>(ii)</b> | <b>(iii)</b> | <b>(iv)</b> | <b>(v)</b> | <b>(vi)</b> | <b>default</b> |
|--------------------------------------|------------|-------------|--------------|-------------|------------|-------------|----------------|
| <b>Número de iterações CG</b>        | 0          | 0           | 0            | 0           | 0          | 0           | 0              |
| <b>Avaliações da função objetivo</b> | 13         | 11          | 9            | 8           | 7          | 9           | 8              |
| <b>Avaliações do gradiente</b>       | 13         | 11          | 9            | 8           | 7          | 9           | 8              |
| <b>Avaliações da Hessiana</b>        | 12         | 10          | 8            | 7           | 6          | 8           | 7              |
| <b>Tempo total</b>                   | 0,051      | 0,066       | 0,098        | 0,099       | 0,087      | 0,093       | 0,075          |
| <b>Tempo gasto na avaliação</b>      | 0,007      | 0,020       | 0,009        | 0,013       | 0,005      | 0,010       | 0,015          |

Os valores das tensões, dos ângulos, das restrições e dos multiplicadores de Lagrange obtidos nos testes para o sistema IEEE 14 barras utilizando as estratégias (i) e (iv), são apresentados no Apêndice A.

Na Figura 5 tem-se a magnitude das tensões na última iteração das estratégias de atualização do parâmetro de barreira. Observa-se que uma magnitude de tensão está ativa no seu limite superior e as demais tensões estão dentro dos limites definidos no ponto de solução do sistema. Nota-se que todas as tensões para cada barra tem os mesmos valores em todas as estratégias de atualização.



**Figura 5** – Magnitude de tensões no sistema elétrico IEEE 14 barras para as estratégias de atualização do parâmetro de barreira

Analisando os resultados da Tabela 3, observamos que a estratégia que resolve o problema de otimização com o menor número de iterações, é a estratégia (v) e a estratégia que resolve com o maior número de iterações é a estratégia (i).

Podemos observar na Tabela 4 que a estratégia (v) obteve o menor número de avaliações da função, avaliações do gradiente e avaliações da Hessiana, e a estratégia (i) o maior número nas três categorias. A estratégia (iv) obteve o maior tempo computacional e a estratégia (i) o menor tempo computacional, e a estratégia (v) obteve o menor tempo em segundos gasto nas avaliações e a estratégia (ii) o maior tempo.

Destacamos que nenhuma das estratégias de atualização do parâmetro de barreira obteve iterações do CG, o que significa que nenhuma destas utilizaram o Método de Região de Confiança para o cálculo do passo.

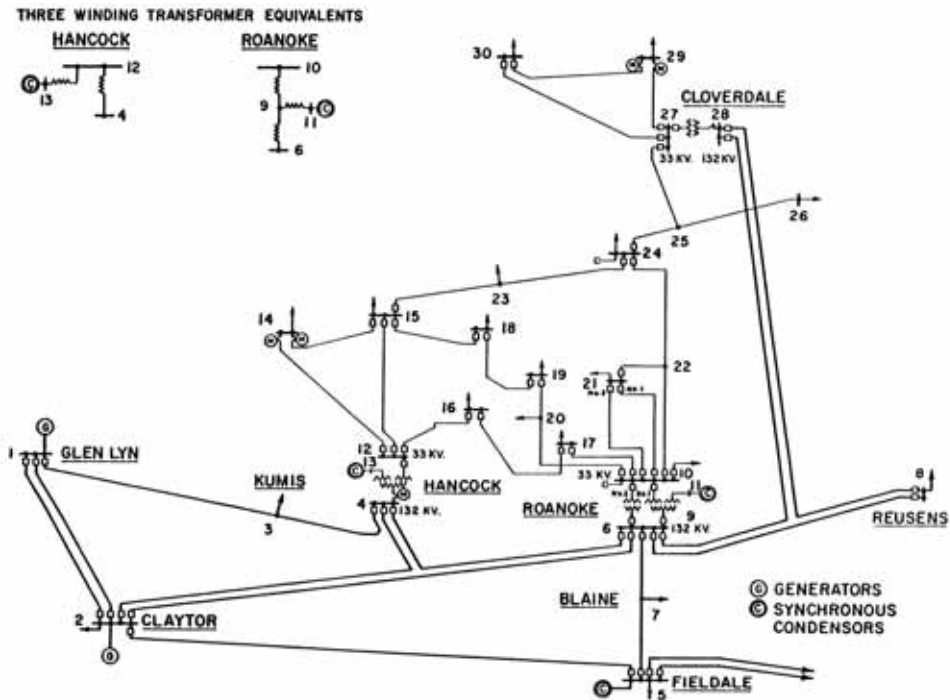
Para os testes no sistema elétrico IEEE 14 barras, o pacote *Knitro* escolheu para a estratégia *default* a estratégia (iv) para atualizar o parâmetro de barreira.

### 6.1.2.1 Sistema elétrico IEEE 30 barras

O sistema elétrico IEEE 30 barras tem as seguintes características:

- 1 barra de geração (*slack*);
- 5 barras de controle de reativo;
- 24 barras de carga;
- 41 linhas de transmissão;
- 4 transformadores com *tap* fixo;

O diagrama do sistema elétrico IEEE 30 barras é apresentado na Figura 6.



**Figura 6** - Sistema elétrico IEEE 30 barras.

Fonte: <http://www.ee.washington.edu/research/pstca/pf30/30bus600.tif>

O modelo do problema de FPO para o sistema elétrico IEEE 30 barras possui 53 restrições de igualdade, 78 restrições de desigualdade e 59 variáveis.

A Tabela 5 apresenta os resultados das Perdas (MW) e a quantidade de iterações para as diferentes estratégias de atualizações do parâmetro de barreira, presentes no pacote de algoritmos de otimização *Knitro*.

**Tabela 5** – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 30 barras

| Estratégias    | Perdas (MW)      | Total de Iterações |
|----------------|------------------|--------------------|
| (i)            | 16,2599352470274 | 14                 |
| (ii)           | 16,2597178497246 | 10                 |
| (iii)          | 16,2597067090325 | 10                 |
| (iv)           | 16,2596921785080 | 7                  |
| (v)            | 16,2596996707274 | 7                  |
| (vi)           | 16,2597040258763 | 10                 |
| <i>default</i> | 16,2596921785080 | 7                  |

A Tabela 6 apresenta as estatísticas finais mostrando o número de iterações realizadas quando se utiliza o Método Região de Confiança com passo CG, o número de avaliações da função objetivo, o número de avaliações do gradiente, o número de avaliações da Hessiana, o tempo total de processamento em segundos e tempo total

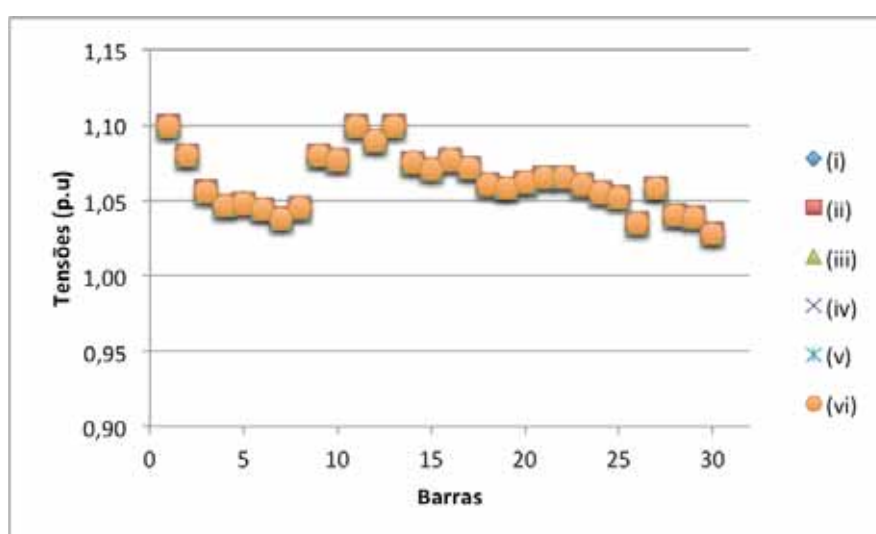
gasto na avaliação também em segundos, para cada estratégia de atualização do parâmetro de barreira disponível no pacote de algoritmos de otimização *Knitro*.

**Tabela 6** – Estatísticas finais para o Sistema Elétrico 30 barras.

| <b>Estratégias</b>                   | <b>(i)</b> | <b>(ii)</b> | <b>(iii)</b> | <b>(iv)</b> | <b>(v)</b> | <b>(vi)</b> |
|--------------------------------------|------------|-------------|--------------|-------------|------------|-------------|
| <b>Número de iterações CG</b>        | 0          | 0           | 0            | 0           | 0          | 0           |
| <b>Avaliações da função objetivo</b> | 15         | 11          | 11           | 8           | 8          | 11          |
| <b>Avaliações do gradiente</b>       | 15         | 11          | 11           | 8           | 8          | 11          |
| <b>Avaliações da Hessiana</b>        | 14         | 10          | 10           | 7           | 7          | 10          |
| <b>Tempo total</b>                   | 0,075      | 0,099       | 0,050        | 0,076       | 0,103      | 0,065       |
| <b>Tempo gasto na avaliação</b>      | 0,026      | 0,031       | 0,013        | 0,019       | 0,019      | 0,013       |

Os valores das tensões, ângulos, restrições e multiplicadores de Lagrange obtidos nos testes utilizando a estratégia (i) e (iv) são apresentados no Apêndice A.

Na Figura 7 tem-se a magnitude das tensões na última iteração das estratégias de atualização do parâmetro de barreira. Observa-se que três magnitudes de tensão estão ativas no seu limite superior e as demais tensões estão dentro dos limites definidos no ponto de solução do sistema. Nota-se que todas as tensões para cada barra tem os mesmos valores em todas as estratégias de atualização.



**Figura 7** – Magnitude de tensões no sistema elétrico IEEE 30 barras para as estratégias de atualização do parâmetro de barreira

Analisando a Tabela 5, observamos que a estratégia (iv) e (v) foram as que obtiveram o menor número de iterações e a estratégia (i) o maior número.

Podemos observar na Tabela 6 que as estratégias (iv) e (v) obtiveram o menor número de avaliações da função, avaliações do gradiente e avaliações da Hessiana e a estratégia (i) obteve o maior número nos mesmos requisitos. A estratégia (iii) obteve o menor tempo total de processamento e tempo gasto na avaliação juntamente com a estratégia (vi), e a estratégia (ii) o maior tempo gasto na avaliação e a estratégia (v) o maior tempo total de processamento e a estratégia utilizada no modo *default* foi a estratégia (iv).

Destacamos que nenhuma das estratégias de atualização do parâmetro de barreira obteve iterações do CG, o que significa que nenhuma destas utilizaram o Método de Região de Confiança para o cálculo do passo.

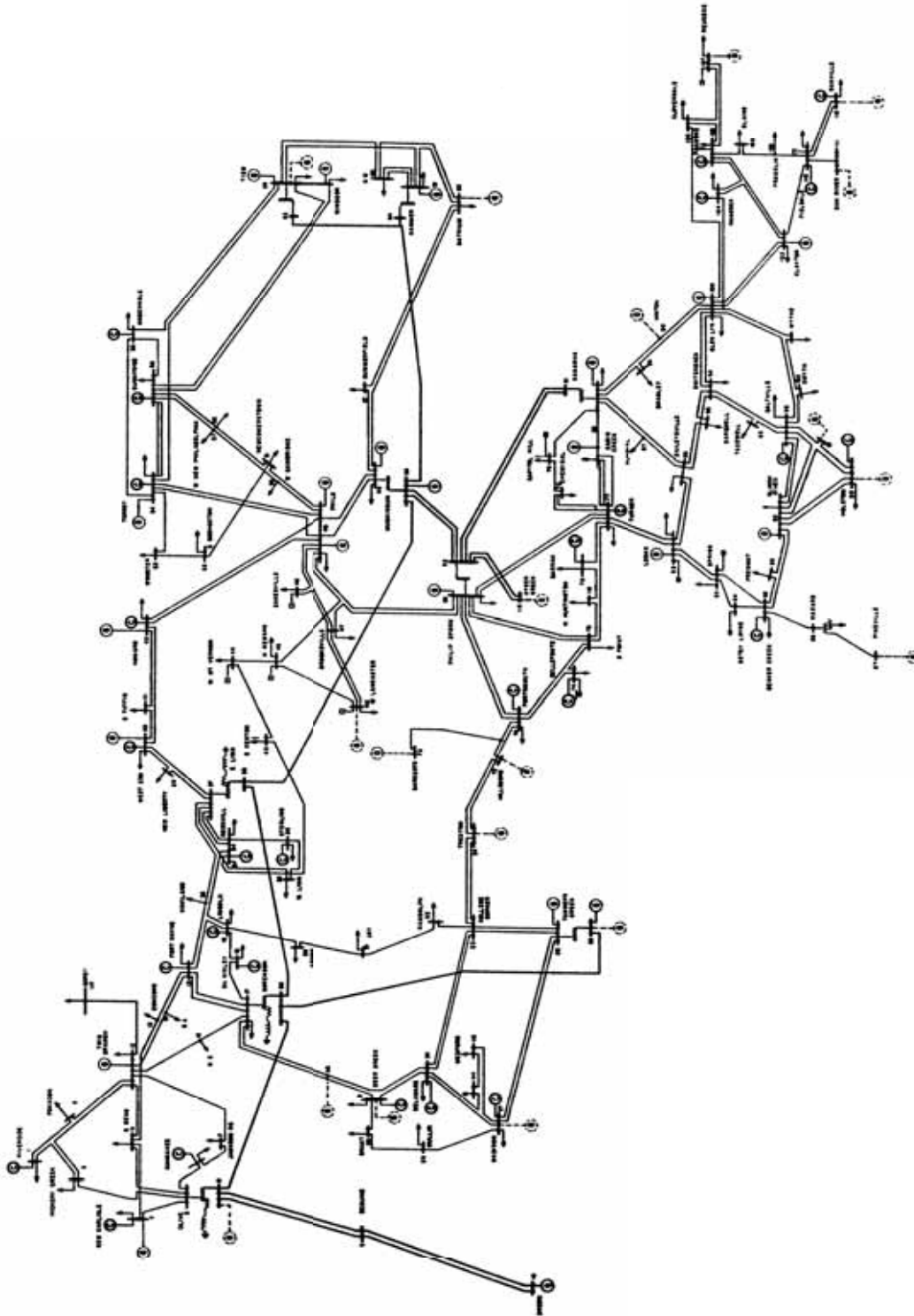
Para os testes no sistema elétrico IEEE 30 barras, o pacote *Knitro* escolheu para a estratégia *default* a estratégia (iv) para atualizar o parâmetro de barreira.

### **6.1.2.3 Sistema elétrico IEEE 118 barras**

O sistema elétrico IEEE 118 barras tem as seguintes características:

- 1 barra de geração (*slack*);
- 53 barras de controle de reativo;
- 64 barras de carga;
- 186 linhas de transmissão;
- 9 transformadores com *tap* fixo;

O diagrama do sistema elétrico IEEE 118 barras é apresentado na Figura 8.



**Figura 8** - Sistema elétrico IEEE 118 barras.

Fonte: <http://www.ee.washington.edu/research/pstca/pf118/118bus6.tif>

O modelo do problema de FPO para o sistema elétrico IEEE 118 barras possui 181 restrições de igualdade, 360 restrições de desigualdade, 235 variáveis contínuas.

A Tabela 7 apresenta os resultados das Perdas (MW) e a quantidade de iterações para as diferentes estratégias de atualizações do parâmetro de barreira, presentes no pacote de algoritmos de otimização *Knitro*.

**Tabela 7** – Valores das Perdas (MW) e número total de iterações para o sistema IEEE 118 barras.

| <b>Estratégias</b> | <b>Perdas (MW)</b> | <b>Total de Iterações</b> |
|--------------------|--------------------|---------------------------|
| <b>(i)</b>         | 107,868087314503   | 19                        |
| <b>(ii)</b>        | 107,867910016911   | 16                        |
| <b>(iii)</b>       | 107,867922218211   | 14                        |
| <b>(iv)</b>        | 107,867885035162   | 10                        |
| <b>(v)</b>         | 107,867917854151   | 9                         |
| <b>(vi)</b>        | 107,867918315493   | 13                        |
| <b>default</b>     | 107,867885035162   | 10                        |

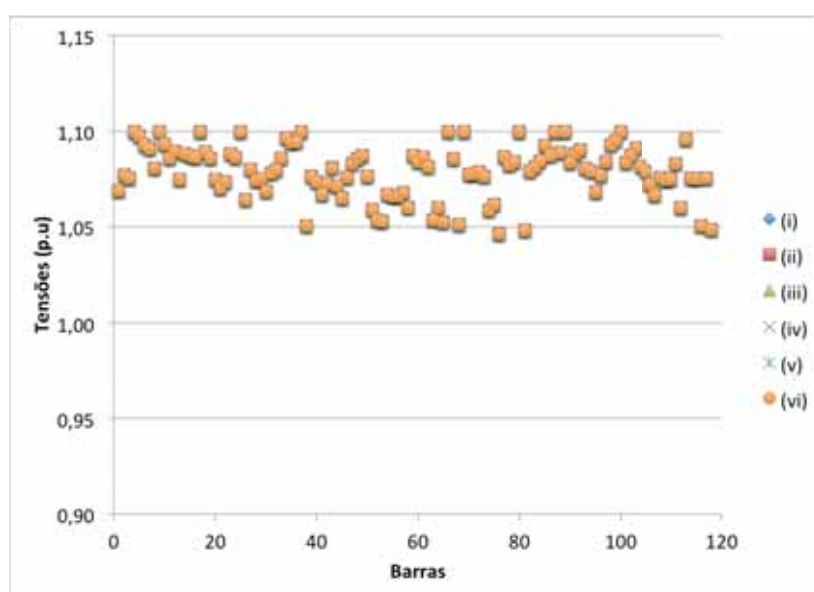
A Tabela 8 apresenta as estatísticas finais mostrando o número de iterações feitas utilizando o Método Região de Confiança com passo CG, número de avaliações da função objetivo, número de avaliações do gradiente, número de avaliações da Hessiana, tempo total de processamento em segundos e tempo total gasto nas avaliações também em segundos, para cada estratégia de atualização do parâmetro de barreira disponível no pacote de algoritmos de otimização *Knitro*.

**Tabela 8** - Estatísticas finais para o sistema elétrico 118 barras.

| <b>Estratégias</b>                   | <b>(i)</b> | <b>(ii)</b> | <b>(iii)</b> | <b>(iv)</b> | <b>(v)</b> | <b>(vi)</b> |
|--------------------------------------|------------|-------------|--------------|-------------|------------|-------------|
| <b>Número de iterações CG</b>        | 0          | 0           | 0            | 0           | 0          | 0           |
| <b>Avaliações da função objetivo</b> | 20         | 17          | 15           | 11          | 10         | 14          |
| <b>Avaliações do gradiente</b>       | 20         | 17          | 15           | 11          | 10         | 14          |
| <b>Avaliações da Hessiana</b>        | 19         | 16          | 14           | 10          | 9          | 13          |
| <b>Tempo total</b>                   | 0,189      | 0,150       | 0,141        | 0,209       | 0,139      | 0,176       |
| <b>Tempo gasto na avaliação</b>      | 0,143      | 0,117       | 0,106        | 0,083       | 0,066      | 0,107       |

Os valores das tensões, ângulos, restrições e multiplicadores de Lagrange obtidos nos testes utilizando a estratégia (i) e (iv) são apresentados no Apêndice A.

Na Figura 9 tem-se a magnitude das tensões na última iteração das estratégias de atualização do parâmetro de barreira. Observa-se que as magnitudes de tensão estão ativas dentro dos limites definidos no ponto de solução do sistema. Nota-se que todas as tensões para cada barra tem os mesmos valores em todas as estratégias de atualização.



**Figura 9** – Magnitude de tensões no sistema elétrico IEEE 118 barras para as estratégias de atualização do parâmetro de barreira

Observa-se na Tabela 7 que a estratégia (v) obteve o menor número de iterações e a estratégia (i) obteve o maior número de iterações; a estratégia (iv) obteve o menor valor de perdas de potência e a estratégia (i) o maior valor.

Na Tabela 8 observa-se que a estratégia (v) obteve o menor número de avaliações da função objetivo e avaliações do gradiente, o menor número de avaliação da Hessiana e também obteve o menor tempo total de processamento; a estratégia (i) obteve o maior número de avaliações da função, avaliações do gradiente e da Hessiana e também obteve o maior tempo de avaliação.

Destacamos que nenhuma das estratégias de atualização do parâmetro de barreira obteve iterações do CG, o que significa que nenhuma destas utilizaram o Método de Região de Confiança para o cálculo do passo.

Para os testes no sistema elétrico IEEE 118 barras, o pacote *Knitro* escolheu para a estratégia *default* a estratégia (iv) para atualizar o parâmetro de barreira.

Na próxima seção apresentamos uma análise geral dos resultados obtidos nos testes citados.

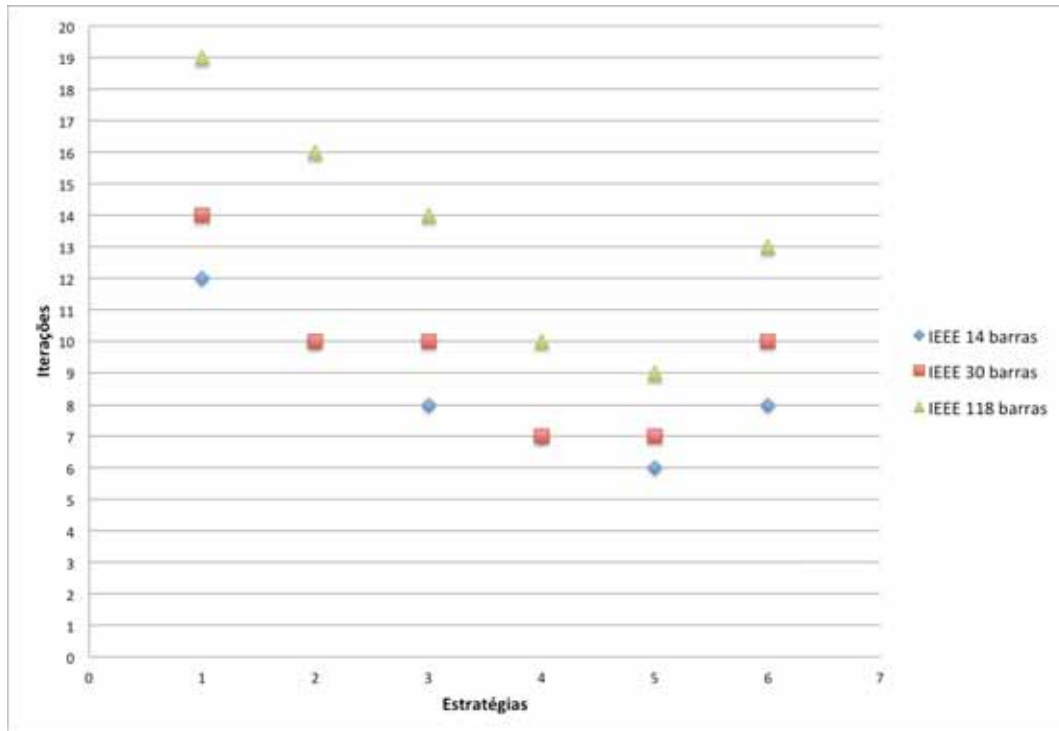
## 6.2 Análise dos resultados

Nesta seção, apresenta-se uma análise dos resultados obtidos para o problema matemático teste e para os sistemas elétricos IEEE 14, 30 e 118 barras.

Nota-se que as estratégias de atualização que obtiveram o melhor desempenho, em termos de número de iterações foram: as estratégias (ii), (iv), (v) e (vi), empatadas, para o problema matemático teste; a estratégia (iv) para o sistema elétrico IEEE 14 barras; as estratégias (iv) e (v) para o sistema elétrico 30 barras; e a estratégia (v) para o sistema elétrico 118 barras. Portanto, conclui-se que, em média, as estratégias que obtiveram o melhor desempenho foram as estratégias (iv) e (v), as quais utilizam o Método Previsor Corretor.

A estratégia que obteve o maior número de iterações, para o problema matemático teste e para os sistemas elétricos IEEE 14, 30 e 118 barras, foi a estratégia (i), a qual utiliza a atualização de Fiacco-McCormick.

A Figura 10 mostra a relação entre a quantidade de iterações realizadas em cada estratégia de atualização do parâmetro de barreira para cada sistema elétrico IEEE, na qual podemos observar qual estratégia obtém o maior número de iterações, sendo ela a (i), ou menor número de iterações, (iv) e (v), e podemos observar que os sistemas elétricos IEEE 14 e 30 barras empataram no números de iterações nas estratégias (ii) e (iv).



**Figura 10** – Relação entre quantidade de iterações com cada estratégia de atualização para os sistemas elétricos IEEE 14, 30 e 118 barras

Ao observarmos os valores finais da função objetivo, nota-se que a estratégia de atualização que obteve o menor valor, foi a estratégia (iv) em todos os testes realizados.

Nota-se que nos testes com as diferentes estratégias de atualização do parâmetro de barreira para o problema matemático teste, apenas a estratégia (v) realizou uma iteração CG, ou seja, para o cálculo do passo foi utilizado o Método Gradiente Conjugado apresentado no capítulo 3. Os testes realizados com os sistemas elétricos IEEE 14, 30 e 118 barras, não realizaram iteração CG em nenhuma das estratégias de atualização do parâmetro de barreira.

Nos resultados do problema matemático teste (115) – (119), observa-se que a estratégia (v) foi a que realizou um maior número de avaliações da função objetivo e as estratégias (ii), (iv) e (vi) o menor número; no sistema elétrico 14 barras, a estratégia (v) obteve o menor número de avaliações da função e a (i) o maior número. No sistema elétrico 30 barras, as estratégias (iv) e (v) obtiveram o menor número de avaliações da função e a (i) o maior número, e o sistema elétrico 118 barras as estratégias (iv) obteve o menor número de avaliações da função e a estratégia (i).

De um modo geral, excluindo-se as opções de tempo, observamos que a estratégia (i) é a que apresentou nos testes o pior desempenho quanto aos requisitos explorados e as estratégias (iv) e (v) o melhor desempenho.

Quanto ao tempo total de processamento, destacamos que não houve um padrão devido ao número de algoritmos acessados durante o processo.

## CAPÍTULO 7

### CONCLUSÕES

Neste trabalho apresentamos uma revisão matemática de alguns métodos implementados no pacote de otimização *Knitro*, tais como: Método de Pontos Interiores, Método de Busca Linear, Método de Região de Confiança, Método do Gradiente Conjugado e Método do Gradiente Projetado.

O pacote de algoritmos de otimização *Knitro* foi projetado para resolver problemas de grande porte e mostra-se eficaz para problemas restritos e irrestritos, sistemas de equações não-lineares, problemas de mínimos quadrados lineares e não-lineares, problemas de programação linear, problemas de programação quadrática e programas de programação não linear inteira mista e convexa de porte médio, inclusive para o problema de FPO.

Nossa proposta foi a de utilizar o Método de Ponto Interior proposto por Waltz et al. (2004) com o objetivo de investigar a influência das diferentes estratégias de atualização do parâmetro de barreira, no problema de Fluxo de Potência Ótimo, estudado na Engenharia Elétrica, área de Sistemas Elétricos de Potência, utilizando o pacote de algoritmos de otimização *Knitro*.

Um breve histórico a respeito do problema de FPO foi apresentado e sua formulação matemática. Testes foram realizados com um problema matemático teste e com os sistemas elétricos IEEE 14, 30 e 118 barras.

Através dos resultados obtidos e apresentados no capítulo 6, observamos em todos os testes que existe uma influência significativa quanto ao número de iterações do método, de acordo com a estratégia de atualização do parâmetro de barreira escolhida.

Conclui-se que as estratégias baseadas no Método Previsor Corretor foram as que apresentaram melhor desempenho quanto as características investigadas em relação ao problema de FPO.

Nota-se também que nenhum dos problemas investigados executaram estratégias do Gradiente Conjugado. Se alterarmos os outros parâmetros envolvidos,

como multiplicadores de Lagrange e ou parâmetro de penalidade, ou alterarmos os pontos iniciais das tensões dos sistemas elétricos de potência, pode-se haver iterações CG, pois o problema a ser resolvido estaria mais longe do ponto factível.

Como trabalhos futuros destacamos:

- atualização dos demais parâmetros envolvidos;
- utilização de Hessianas exatas e aproximadas;
- inicialização do parâmetro de barreira.

## CAPÍTULO 8

### Trabalhos publicados

Neste capítulo relacionamos os trabalhos publicados e apresentados.

- 1. Título:** A influência dos parâmetros nos métodos mistos de otimização restrita não linear.

Trabalho completo publicado e apresentado no congresso Ibero Latino Americano em métodos computacionais em engenharia - CILAMCE, 2011, Ouro Preto/MG.
- 2. Título:** Uma investigação da convergência do problema de fluxo de potência ótimo sob a influência dos parâmetros.

Resumo publicado e apresentado em pôster no I Seminário da Pós-graduação em Engenharia Elétrica, 2011, Unesp – Campus Bauru/SP.
- 3. Título:** An investigation about barrier parameters update strategy and the Optimal Power Flow Solution.

Trabalho completo publicado e apresentado em pôster no 3º Congresso Internacional de Otimização em Engenharia - EngOpt, 2012, Rio de Janeiro/RJ.
- 4. Título:** Um método de ponto interior com análise das atualizações do parâmetro de barreira através do *Knitro*.

Resumo estendido publicado e apresentado no Encontro Regional de Matemática Aplicada e Computacional – ERMAC, 2012, Unesp – Botucatu/SP.
- 5. Título:** Regras de Atualização do Parâmetro de Barreira e o Problema de Fluxo de Potência Ótimo.

Resumo publicado no XXXIV Congresso Nacional de Matemática Aplicada e Computacional – CNMAC, 2012, Águas de Lindóia/SP.

- 6. Título:** Um método misto de busca linear e região de confiança com diferentes estratégias de atualização de parâmetros aplicado ao FPO.  
Trabalho completo publicado no Congresso Ibero Latinoamericano de Investigação Operacional – CLAIO, Simpósio Brasileiro de Pesquisa Operacional – SBPO, 2012, Rio de Janeiro/RJ.
  
- 7. Título:** Uma investigação dos parâmetros nos métodos mistos de otimização não linear.  
Trabalho completo publicado e apresentado no II Seminário da Pós-Graduação em Engenharia Elétrica, 2012, Unesp – Campus Bauru/SP.

## Referências

Adler, I., Karmakar, N. Resende & M. Veiga, G. (1989). An implementation of Karmakar's algorithm for linear programming. *Mathematical Programming*, **44**, 3, 297-335.

Almeida, K. C. de, (2005). Um estudo sobre casos críticos de solução do fluxo de potência ótimo. *Revista Controle e Automação*, maio 2005.

Baptista E. C., Belati E. A., Sousa V. A. & Costa G. R. M. Da, (2006). Primal-Dual Logarithmic Barrier and Augmented Lagrangian Function to the Loss Minimization in Power Systems. *Electric Power Components and Systems*, 34, 7, 775-784

Barnes, E. R. (1986). A variation on Karmarkar algorithm for solving linear-programming problems. *Mathematical Programming*, **36**, 2, 174-182.

Bazaraa, M. S., Sherali, H. D. & Shetty, C. M. (1993). *Nonlinear programming: theory and algorithms*. John Wiley, 2 ed., New York.

Betts, J., Eldersveld, S. K., Frank, P. D., & Lewis, J. G. (2003). An interior-point nonlinear programming algorithm for large scale optimization. In O. Ghattas, M. Heinkenschloss, D. Keyes, L. T. Biegler, & B. van Bloemen Waanders, Editors, *Large-Scale PDE- Constrained Optimization*, Lecture Notes in Computational Science and Engineering, 184–198. Springer Verlag.

Byrd, R. H., Hribar, M. E. & Nocedal, J. (1999). An interior point algorithm for large scale nonlinear programming. *Siam Journal on Optimization*, **9**(4), 877–900.

Byrd, R. H., Nocedal, J. & Waltz, R. A. (2006). *Knitro*: An integrated package for nonlinear optimization, In: G. Di Pillo & M. Roma, Editors, *Large-Scale Nonlinear Optimization*, 35–59, Springer.

Carpentier, J. L. (1962). Contribution a L'étude du Dispatching Economique. *Bull-Soc. Fr Elec.*, Ser. B3, 431 – 447.

Carrol, C.W. (1961). The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems. *Operations Research*, 9, 169-184.

Carvalho, L. M. R., (2005). Métodos de pontos interiores aplicados ao pre despacho de um sistema hidroeletrico usando o principio de mínimo esforço – comparação com modelo de fluxo de redes, Tese (Doutorado), Universidade de São Paulo, São Carlos.

Celis, M. R., Dennis, J. E., Tapia, R. A. (1984). A trust region strategy for nonlinear equality constrained optimization. In Numerical Optimization, *SIAM Society on Industrial and Applied Mathematics*.

Coleman, T. F. (1994). Linearly constrained optimization and projected preconditioned conjugate gradients. In J. Lewis, Editor, Proceedings Of The Fifth SIAM Conference On Applied Linear Algebra, 118–122, Philadelphia. Siam.

Chung, C. Y., Yan W., Liu F., (2011). Decomposed Predictor-Corrector Interior Point Method for Dynamic Optimal Power Flow, *IEEE Transactions on Power Systems*, 26(3), 1030-1039.

Conn, A. R., Gould, N. I. M., Orban, D. & Toint, Ph. L. (1988). *A global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25, 433-460.

Conn, A. R., Gould, N. I. M. & Toint, Ph. L. (2000). *Trust-region methods*. MPS-SIAM Series on Optimizatón. SIAM publications, Philadelphia, Pennsylvania, USA.

Costa, G. R. M., Costa, C. E. U., Souza, A. M. (2000). Comparative studies of optimization methods for the optimal power flow problem. *Electric Power Systems Research*, **56** (3), 249-254;

Dommel, H. W. & Tinney, W. F. (1968). Optimal power flow solutions. *IEEE Transactions on Power Apparatus and Systems*, **PAS-87**(10), 1866–1876.

El-Bakry, S., Tapia, R. A., Tsuchiya, T., & Zhang, Y. (1996). On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of optimization theory and applications*, **89**(3): 507–541.

Fang, S. C., Puthenpura, S., (1993). *Linear Optimization and Extensions: Theory and Algorithms*, Prentice-Hall, Englewood Cliffs, New Jersey.

Ferreira, E. C., Baptista, E. C. & Soler, E. M. (2012). Um método de ponto interior com análise das atualizações do parâmetro de barreira através do *Knitro*. Encontro Regional de Matemática Aplicada e Computacional – ERMAC, Botucatu, São Paulo.

Fiacco, A. V. & Mc-Cormick, G. P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. New York, John Wiley & Sons.

Fletcher, R., (1970). A class of methods for nonlinear programming with termination and convergence properties. In J. Abadie, ed., *Integer and Nonlinear Programming*, 157-175, North-Holland, Amsterdam, the Netherlands.

Fletcher, R., (1980). Practical methods of optimization. *Unconstrained optimization*, v. I, John Wiley & Sons, New York.

Frisch, K. R. (1955). The logarithmic Potential Method of Convex Programming, Memorandum. University Institute of Economics, Oslo, Norway.

Gay, D. M., Overton, M. L., & Wright, M. H. (1998). A primal-dual interior method for nonconvex nonlinear programming. In: Y. Yuan, ed., *Advances in Nonlinear Programming*, Beijing, 1996. Kluwer Academic Publishers, Dordrecht, The Netherlands, 31–56.

Goldfeldt, S. M., Quandt, R. E., & Trotter, H. F. (1966). Maximization by quadratic hill-climbing. *Econometrica*, 34:541–551.

Gonzaga, C. (1989). An algorithm for solving linear programming problems in  $O(n^3L)$  operations. In: Progress in Mathematical Programming: Interior-Point and Related Methods, ed. N. Megiddo, Springer-Verlag, New York, 1-28.

Gonzaga, C. (1990). Polynomial affine algorithms for linear programming. *Mathematical Programming*, 49, 7-21.

Gould, N. I. M., Hribar, M. E., & Nocedal, J. (2001). On the solution of equality constrained quadratic problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1375–1394.

Granville, S. (1994). Optimal Reactive Dispatch Through Interior Point Methods. *IEEE Transactions on Power Systems*, 9, 136-146.

Griffith, R. E., Stewart R. A. (1961). A nonlinear programming technique for the optimization of continuous processing systems. *Management Science*, 7, 379–392.

Karmakar, N. (1984). A new polynomial-time algorithm for linear programming. *Combinatorics*, 4, 373-395.

Keller, C., Gould, N. I. M., & Wathen, A. J. (2000). Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1300–1317.

Khiat, M., Marano, A., (2010). A Primal-Dual Interior Point Method to Minimize Total Active Power Loss in Electric Power Systems, *International Review of Electrical Engineering-IREE*, 5(4), 1627-1632.

Kojima, M., Mizuno, S. & Yoshise, A. (1989). A primal dual - interior point method for linear programming. In: Progress in Mathematical Programming: Interior-Point and Related Methods, Ed. N. Megiddo, Springer-Verlag, New York, 29-48.

Lage, G. G., Sousa, V. A. de, Pereira, L. S., Costa, G. R. M. da, (2008). An Enhanced Model Applied to the Optimal Power Flow Problem, 40TH North American Power Symposium (NAPS 2008), North American Power Symposium, 122-127.

Lage, G. G., Sousa, V. A. de, Costa, G. R. M. da, (2009). Optimal Power Flow Solution Using the Penalty/Modified Barrier Method, IEEE Bucharest Powertech, 1-5, 1932-1937.

Levenberg, K. (1944). *A method for the solution of certain problems in least squares*. Quart. Appl. Math., Vol. 2, pp. 164–168.

Luenberger, D. G. (1984). *Linear and Nonlinear Programming*. Addison-Wesley Publishing Company, California, Second Edition.

Luenberger D. C., Ye Y. (2008). *Linear and Nonlinear Programming*. 3.ed., Springer, New York.

Luksan, L. & Vlcek, J. (1998). Indefinitely preconditioned inexact Newton Method for large sparse equality constrained nonlinear programming problems. *Numerical linear algebra with applications*, **5**(3):219–247.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of non-linear parameter. *Journal of the Society for Industrial and Applied Mathematics*, **11**(2), 431-441.

Megiddo, N. (1987). Pathways to the optimal set in linear programming. In *Progress in Mathematical Programming: Interior-Point and Related Methods*, N. Megiddo, Editor, Springer - Verlag, New York., **8**, 131–158.

Megiddo, N. & Shub, M. (1989). Boundary behavior of interior point algorithms in linear programming. *Mathematics of Operations Research*, **14**, 97-146.

Mehrotra, S. (1992). On implementation of a primal-dual interior point method. *Siam Journal on Optimization*, **2**, 575-601.

Momoh, J. A., EL-Hawary, M. & Adapa, R. (1999a). A review of selected optimal power flow literature to 1993 - part i: Nonlinear and quadratic programming approaches. *IEEE Transactions on Power Systems*, **14**(1), 96–104.

Momoh, J. A., EL-Hawary, M. & Adapa, R. (1999b). A review of selected optimal power flow literature to 1993 - part ii: Newton, linear programming and interior point methods. *IEEE Transactions on Power Systems*, **14**(1), 105–111.

Monteiro, R. D. C. & Adler, I. (1989). Interior path-following primal-dual algorithms. Part I: Linear Programming, Part II: Convex Quadratic Programming. *Mathematical Programming*, **44**, 27-66.

Monteiro, R. D. C., Adler, I. & Resende, M. C. (1990). A polynomial-time primal-dual affine scaling algorithm for linear and convex quadratic programming and its power series extension. *Mathematics of Operations Research*, **15**, 191-214.

Moré, J. & Sorensen, D. (1983). Computing a Trust Region Step. *SIAM Journal on Scientific and Statistical Computing*, v. 4, n. 3, pp. 553-571.

Morrison, D. D. (1960). Methods for nonlinear least squares problems and convergence proofs. In: Lorell J, Yagi F (eds) *Proceedings of the Seminar on Tracking Programs and Orbit Determination*. Pasadena, USA, Jet Propulsion Laboratory, 1–9.

Nejdawi, I. M., Clements, K. A., Davis, P. W. (2000). Na Efficient Interior Point Method for Sequential Quadratic Programming Based Optimal Power Flow. *IEEE Transactions on Power Systems*, **15** (4), 1179-1183.

Nocedal, J., Waltz, R. A. (2001). “Knitro 1.00 User’s Manual”, Technical Report, Ziena Optimization, Evanston, Usa.

Nocedal, J., Wright, S. J. (1999). *Numerical Optimization*. New York: Springer-Verlag.

Oliveira L. M. R. (2005). Métodos de pontos interiores aplicados ao pré-despacho de um sistema hidroelétrico usando o princípio de mínimo esforço – comparação com o modelo de fluxo em redes. Tese de doutorado, Instituto de Ciências Matemáticas e Computação – ICMC-USP, São Carlos.

Oliveira A. R. L. & Filho S. S. (2003). Métodos de pontos interiores para problema de fluxo de potência ótimo DC. *Revista Controle & Automação*, **14** (3), 278-285, jul. 2003.

Platenga T. D. (2006). Knitro for Nonlinear Optimal Power Flow Applications. Case studies in optimization. Ziena Optimization Inc. ([http://www.ziena.com/papers/case\\_OPF.pdf](http://www.ziena.com/papers/case_OPF.pdf)).

Platenga T. D., Waltz R. A. (2006). *Knitro Performance on Large Nonlinear Problems*. Case studies in optimization. Ziena Optimization Inc. ([http://www.ziena.com/papers/case\\_performance.pdf](http://www.ziena.com/papers/case_performance.pdf)).

Pinheiro R. B. N. (2012). Um método previsor-corretor primal-dual de pontos interiores barreira logarítmica modificada, com estratégias de convergência global e de ajuste cúbico, para problemas de programação não-linear e não-convexa. Dissertação de mestrado. Programa de Pós Graduação em Engenharia Elétrica, Universidade Estadual Paulista “Júlio de Mesquita Filho”, Bauru.

Polyak, B. T. (1969). The conjugate gradient method in extremal problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, **9**, 94–112.

Powell, M. J. D., (1970a). A new algorithm for unconstrained optimization. In Ritter Rosen, Mangasarian, editor, *Nonlinear Programming*, 31–65. Academic Press, New York.

Powell, M. J. D., (1970b). Trust methods that employ quadratic interpolation to the objective function. *SIAM Conference on Optimization*, Victoria, BC, Canada.

Probst, R. W., (2010). Um novo método previsor-corretor para fluxo de potência ótimo. Tese de doutorado. Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, UNICAMP: Programa de Pós-Graduação em Matemática Aplicada.

Rashed, A. M. H., Kelly, D. H. (1974). Optimal Load Flow Solution Using Lagrangian Multipliers and the Hessian Matrix. *IEEE Transactions on Power Apparatus and Systems*, **93**, 5, 1292-1297.

Renegar, J. (1988). *A polynomial-time algorithm based on Newton's method for linear programming*. *Mathematical Programming*, **40**, 59-93.

Sasson, A., Vilorio, F. & Aboites, F. (1973). Optimal load flow solution using the hessian matrix. *IEEE Transactions on Power Apparatus and Systems*, **PAS-92**(1), 31–41.

Soler E. M., Sousa V. A. de, Costa G. R.M. da, (2012). A modified Primal–Dual Logarithmic-Barrier Method for solving the Optimal Power Flow problem with discrete and continuous control variables. *European Journal of Operational Research*, **222**(3), 616–622.

Song, H., Dosano R. D., Kim, T., Lee, B., Shin J., Kim, T., (2009). A Penalty Approach for NIPM based Optimal Reactive Power Flow Including Discrete Variables, *IEEE, Transmission & Distribution Conference & Exposition Asia and Pacific*, 350-353.

Sousa V. A. de, Baptista E. C., Costa G. R. M. da, (2009). Loss minimization by the predictor–corrector modified barrier approach. *Electric Power Systems Research*, **79**(5), 803–808.

Sun, D. I., Ashley, B., Hugues, A., Tinney, W. F. (1984). Optimal Power Flow by Newton Approach. *IEEE Transactions on Power Apparatus and Systems*, **PAS-103**, 10, 2864-2875.

Torres, G. L. & Quintana, V. H. (1998). An interior point method for nonlinear optimal power flow using voltage rectangular coordinates. *IEEE Transactions on Power Systems*, **13**(4), 1211–1218.

Thomaz, A. (2007). *Método de pontos interiores aplicado ao fluxo de potência ótimo utilizando coordenadas cartesianas*. Tese de Doutorado - Universidade Estadual de Campinas - UNICAMP . Faculdade de Engenharia Elétrica e de Computação, Campinas.

Ulbrich, M., Ulbrich, S. & Vicente, L. N. (2004). *A globally convergent primal-dual interior point filter method for nonconvex nonlinear programming*. *Mathematical Programming*, **100** (2), 379–410.

Vaidya, P. M. (1990). An algorithm for linear programming which requires  $O(((m+n)n^2+(m+n)^{1.5}n)L)$  arithmetic operations. *Mathematical Programming*, **47**, 175-2001.

Vanderbei, R. J., Meketon, M. S. & Freedman, B. A. (1986). A modification of Karmarkar's linear programming algorithms. *Algorithmica*, **1**, 395-407.

Vanderbei, R. J. & Shanno, D. F. (1999). An interior point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, **13**, 231–252.

Wächter, A. & Biegler, L. T. (2005). Line search filter methods for nonlinear programming: Motivation and global convergence. *Siam Journal on Optimization*, **16**(1):1–31.

Waltz, R. A., Morales, J. L., Nocedal, J., Orban, D. (2004). An interior algorithm for nonlinear optimization that combines line search and trust region steps. *Mathematical Programming, Series A*, **107**, 391–408.

Waltz, R. A., Nocedal, J. (2002). “*Knitro 2.0 User's Manual*”, Technical Report, Ziena Optimization, Evanston, Il, Usa.

Waltz, R. A., Nocedal, J. (2003). “Knitro User’s Manual”, Technical Report, Ziena Optimization, Evanston, Il, Usa.

Waltz, R. A., Platenga, T. D. (2009). “Knitro User’s Manual. Version 6.0”, Technical Report, Ziena Optimization Inc., Evanston, Il, Usa.

Waltz, R. A., Platenga, T. D. (2010). “Knitro User’s Manual. Version 7.0”, Technical Report, Ziena Optimization, Evanston, Il, Usa.

Winfield, D. (1973). Function minimization by interpolation in a data table. *J. Inst. Maths Applics*, **12**, 339–347.

Wright, S. J. (1997). Primal-Dual Interior-Point Methods. *Society for Industrial and Applied Mathematics*, Philadelphia.

Wu, Y., Debs, A. S., Marsten, R. E. (1994). A Direct Nonlinear Predictor Corrector Primal-Dual Interior Point Algorithm for Optimal Power Flows. *IEEE Transactions on Power Systems*, **9**, 876-883.

Ye, Y. (1986). An  $O(n^3 L)$  potential reduction algorithm for linear programming. *Contemporary Mathematics*, **114**, 91-107.

# APÊNDICE A

## RESULTADOS

### A.1 Arquivo de saída para o sistema IEEE 14 barras – Estratégia (i)

A seguir são apresentados os valores finais obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange no teste realizado para o Sistema IEEE 14 barras utilizando a estratégia (i) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 32$ ;
- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 14$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 15, \dots, 32$  e
- $\text{lambda}[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 32$ .

| Constraint Vector              | Lagrange Multipliers                      |
|--------------------------------|---|
| $c[0] = -4.59481149934e-008,$  | $\text{lambda}[0] = -1.00000000000e+000$  |
| $c[1] = -1.83000016775e-001,$  | $\text{lambda}[1] = -5.07015017473e-002$  |
| $c[2] = 9.41999989816e-001,$   | $\text{lambda}[2] = -1.26364539056e-001$  |
| $c[3] = 4.78000030298e-001,$   | $\text{lambda}[3] = -1.02631482653e-001$  |
| $c[4] = 7.60000089011e-002,$   | $\text{lambda}[4] = -8.60717034858e-002$  |
| $c[5] = 1.11999987402e-001,$   | $\text{lambda}[5] = -8.80070136303e-002$  |
| $c[6] = 1.49426420126e-008,$   | $\text{lambda}[6] = -1.02102899136e-001$  |
| $c[7] = 0.00000000000e+000,$   | $\text{lambda}[7] = -1.02102899136e-001$  |
| $c[8] = 2.94999977894e-001,$   | $\text{lambda}[8] = -1.01797032720e-001$  |
| $c[9] = 8.99999916052e-002,$   | $\text{lambda}[9] = -1.05791411455e-001$  |
| $c[10] = 3.49999971586e-002,$  | $\text{lambda}[10] = -1.00725795975e-001$ |
| $c[11] = 6.09999962263e-002,$  | $\text{lambda}[11] = -1.06796072402e-001$ |
| $c[12] = 1.34999994629e-001,$  | $\text{lambda}[12] = -1.12522626126e-001$ |
| $c[13] = 1.48999986212e-001,$  | $\text{lambda}[13] = -1.30442351195e-001$ |
| $c[14] = -3.90000046440e-002,$ | $\text{lambda}[14] = -6.21579144776e-004$ |
| $c[15] = 1.60000008562e-002,$  | $\text{lambda}[15] = -2.15923098889e-003$ |
| $c[16] = 1.20033760709e-010,$  | $\text{lambda}[16] = -1.99247578751e-005$ |
| $c[17] = 1.65999985174e-001,$  | $\text{lambda}[17] = 2.68039366719e-004$  |
| $c[18] = 5.79999990846e-002,$  | $\text{lambda}[18] = -4.51525597081e-003$ |
| $c[19] = 1.79999998051e-002,$  | $\text{lambda}[19] = -6.95349506174e-003$ |
| $c[20] = 1.60000013725e-002,$  | $\text{lambda}[20] = -1.19443799759e-002$ |
| $c[21] = 5.79999993795e-002,$  | $\text{lambda}[21] = -1.43136955436e-002$ |
| $c[22] = 5.00000018789e-002,$  | $\text{lambda}[22] = -1.44661019512e-002$ |
| $c[23] = -2.41877377259e-001,$ | $\text{lambda}[23] = -4.01010053838e-009$ |

|                               |                                   |
|-------------------------------|-----------------------------------|
| c[ 24] = 2.23454061108e-001,  | lambda[ 24] = -5.33055990150e-007 |
| c[ 25] = 9.86384425302e-003,  | lambda[ 25] = -2.00139871665e-006 |
| c[ 26] = 1.64934435345e-001,  | lambda[ 26] = -1.33362523253e-006 |
| c[ 27] = 2.10983097476e-001,  | lambda[ 27] = -1.47629930044e-006 |
| c[ 28] = -2.41877377259e-001, | lambda[ 28] = 3.99074636427e-009  |
| c[ 29] = 2.23454061108e-001,  | lambda[ 29] = 2.67648962379e-006  |
| c[ 30] = 9.86384425302e-003,  | lambda[ 30] = 2.98798717409e-003  |
| c[ 31] = 1.64934435345e-001,  | lambda[ 31] = 6.10425854146e-003  |
| c[ 32] = 2.10983097476e-001,  | lambda[ 32] = 1.57060612252e-005  |

Solution Vector

|                               |                                   |
|-------------------------------|-----------------------------------|
| x[ 0] = 1.23615894864e-001,   | lambda[ 33] = 0.00000000000e+000  |
| x[ 1] = 1.09999839217e+000,   | lambda[ 34] = 2.48779449309e-001  |
| x[ 2] = 1.08657375661e+000,   | lambda[ 35] = 2.68638014242e-005  |
| x[ 3] = 1.05278774869e+000,   | lambda[ 36] = 4.58196283570e-006  |
| x[ 4] = 1.06509652644e+000,   | lambda[ 37] = 7.98887019039e-006  |
| x[ 5] = 1.07069928745e+000,   | lambda[ 38] = 1.03420508257e-005  |
| x[ 6] = 1.03338329712e+000,   | lambda[ 39] = 1.20860028227e-006  |
| x[ 7] = 1.05192035410e+000,   | lambda[ 40] = 4.40746462504e-006  |
| x[ 8] = 1.08613763149e+000,   | lambda[ 41] = 2.84076184547e-005  |
| x[ 9] = 1.03732440421e+000,   | lambda[ 42] = 1.80493254431e-006  |
| x[ 10] = 1.02908849802e+000,  | lambda[ 43] = 5.85226445794e-007  |
| x[ 11] = 1.02775464577e+000,  | lambda[ 44] = 3.93669653722e-007  |
| x[ 12] = 1.01938049924e+000,  | lambda[ 45] = -8.03035548663e-007 |
| x[ 13] = 1.01581172235e+000,  | lambda[ 46] = -1.32600783682e-006 |
| x[ 14] = 1.00950543668e+000,  | lambda[ 47] = -2.30077372899e-006 |
| x[ 15] = -8.03625711060e-002, | lambda[ 48] = 0.00000000000e+000  |
| x[ 16] = -2.04427209141e-001, | lambda[ 49] = 0.00000000000e+000  |
| x[ 17] = -1.67385131396e-001, | lambda[ 50] = 0.00000000000e+000  |
| x[ 18] = -1.42963672756e-001, | lambda[ 51] = 0.00000000000e+000  |
| x[ 19] = -2.42217237900e-001, | lambda[ 52] = 0.00000000000e+000  |
| x[ 20] = -2.24772019235e-001, | lambda[ 53] = 0.00000000000e+000  |
| x[ 21] = -2.24772019235e-001, | lambda[ 54] = 0.00000000000e+000  |
| x[ 22] = -2.55075362413e-001, | lambda[ 55] = 0.00000000000e+000  |
| x[ 23] = -2.58202983767e-001, | lambda[ 56] = 0.00000000000e+000  |
| x[ 24] = -2.52844790939e-001, | lambda[ 57] = 0.00000000000e+000  |
| x[ 25] = -2.57935010167e-001, | lambda[ 58] = 0.00000000000e+000  |
| x[ 26] = -2.59696473882e-001, | lambda[ 59] = 0.00000000000e+000  |
| x[ 27] = -2.75487382252e-001, | lambda[ 60] = 0.00000000000e+000  |

## A.2 Arquivo de saída para o sistema IEEE 30 barras – Estratégia (i)

A seguir são apresentados os valores obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange para o sistema IEEE 30 barras no teste realizado utilizando a estratégia (i) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 65$ ;

- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 30$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 31, \dots, 59$  e
- $\text{lambda}[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 125$ .

Constraint Vector

Lagrange Multipliers

| -----                         | -----                             |
|-------------------------------|-----------------------------------|
| c[ 0] = -1.12987021406e-008,  | lambda[ 0] = -1.00000000000e+000  |
| c[ 1] = -1.83000000987e-001,  | lambda[ 1] = -5.80794833095e-002  |
| c[ 2] = 2.40000000118e-002,   | lambda[ 2] = -7.77022726192e-002  |
| c[ 3] = 7.59999997782e-002,   | lambda[ 3] = -1.00721133100e-001  |
| c[ 4] = 9.41999997471e-001,   | lambda[ 4] = -1.44901781001e-001  |
| c[ 5] = -1.05903019687e-009,  | lambda[ 5] = -1.18433077438e-001  |
| c[ 6] = 2.27999996771e-001,   | lambda[ 6] = -1.36034335115e-001  |
| c[ 7] = 2.9999999473e-001,    | lambda[ 7] = -1.25917597054e-001  |
| c[ 8] = -1.40854605757e-010,  | lambda[ 8] = -1.18992778815e-001  |
| c[ 9] = 5.7999999133e-002,    | lambda[ 9] = -1.19243457385e-001  |
| c[ 10] = 0.0000000000e+000,   | lambda[ 10] = -1.18992778815e-001 |
| c[ 11] = 1.1199999633e-001,   | lambda[ 11] = -1.03610619897e-001 |
| c[ 12] = 0.0000000000e+000,   | lambda[ 12] = -1.03610619897e-001 |
| c[ 13] = 6.1999999762e-002,   | lambda[ 13] = -1.21144706154e-001 |
| c[ 14] = 8.1999999875e-002,   | lambda[ 14] = -1.26828655147e-001 |
| c[ 15] = 3.5000000038e-002,   | lambda[ 15] = -1.17542466525e-001 |
| c[ 16] = 8.9999999419e-002,   | lambda[ 16] = -1.22304257369e-001 |
| c[ 17] = 3.1999999878e-002,   | lambda[ 17] = -1.39090575272e-001 |
| c[ 18] = 9.4999999598e-002,   | lambda[ 18] = -1.42237016742e-001 |
| c[ 19] = 2.1999999878e-002,   | lambda[ 19] = -1.37165379961e-001 |
| c[ 20] = 1.7499999919e-001,   | lambda[ 20] = -1.30283662919e-001 |
| c[ 21] = -5.72519809339e-012, | lambda[ 21] = -1.29865638821e-001 |
| c[ 22] = 3.1999999846e-002,   | lambda[ 22] = -1.37418580666e-001 |
| c[ 23] = 8.6999999476e-002,   | lambda[ 23] = -1.42815063220e-001 |
| c[ 24] = -1.00026653627e-011, | lambda[ 24] = -1.36947894781e-001 |
| c[ 25] = 3.4999999758e-002,   | lambda[ 25] = -1.56623927772e-001 |
| c[ 26] = 2.97982749586e-011,  | lambda[ 26] = -1.25353645313e-001 |
| c[ 27] = -6.28732621522e-011, | lambda[ 27] = -1.25227976155e-001 |
| c[ 28] = 2.3999999767e-002,   | lambda[ 28] = -1.54758647550e-001 |
| c[ 29] = 1.0599999875e-001,   | lambda[ 29] = -1.75052373623e-001 |
| c[ 30] = 1.1999999886e-002,   | lambda[ 30] = -1.16749748491e-002 |
| c[ 31] = 1.6000000798e-002,   | lambda[ 31] = -1.15161521846e-002 |
| c[ 32] = 3.39120731496e-010,  | lambda[ 32] = -9.12860289070e-003 |
| c[ 33] = 1.0900000622e-001,   | lambda[ 33] = -1.01586596650e-002 |
| c[ 34] = 2.89723800506e-012,  | lambda[ 34] = -5.95613440900e-003 |
| c[ 35] = 1.9999999299e-002,   | lambda[ 35] = -7.49213465389e-003 |
| c[ 36] = 7.5000000031e-002,   | lambda[ 36] = -4.17526160157e-003 |
| c[ 37] = 1.6000000121e-002,   | lambda[ 37] = -9.47371745628e-003 |
| c[ 38] = 2.4999999926e-002,   | lambda[ 38] = -1.23889243198e-002 |
| c[ 39] = 1.79999999891e-002,  | lambda[ 39] = -9.65628817389e-003 |
| c[ 40] = 5.8000000060e-002,   | lambda[ 40] = -1.05927800409e-002 |
| c[ 41] = 9.0000000207e-003,   | lambda[ 41] = -1.58469555270e-002 |
| c[ 42] = 3.4000000056e-002,   | lambda[ 42] = -1.66516259646e-002 |
| c[ 43] = 7.0000000011e-003,   | lambda[ 43] = -1.46437275931e-002 |
| c[ 44] = 1.1199999995e-001,   | lambda[ 44] = -1.40796660817e-002 |
| c[ 45] = 2.76223488527e-013,  | lambda[ 45] = -1.36169995906e-002 |
| c[ 46] = 1.6000000016e-002,   | lambda[ 46] = -1.70835492004e-002 |

|                               |                                   |
|-------------------------------|-----------------------------------|
| c[ 47] = 6.70000000031e-002,  | lambda[ 47] = -1.88311246712e-002 |
| c[ 48] = 1.75948144943e-012,  | lambda[ 48] = -2.19513071264e-002 |
| c[ 49] = 2.29999999977e-002,  | lambda[ 49] = -3.50745019328e-002 |
| c[ 50] = -6.53108678250e-011, | lambda[ 50] = -1.78149070467e-002 |
| c[ 51] = 4.49666970326e-011,  | lambda[ 51] = -1.06804974993e-002 |
| c[ 52] = 9.00000000173e-003,  | lambda[ 52] = -2.59601613843e-002 |
| c[ 53] = 1.90000000281e-002,  | lambda[ 53] = -2.92942680166e-002 |
| c[ 54] = -4.63373889797e-002, | lambda[ 54] = -4.00225476575e-009 |
| c[ 55] = 3.72765689680e-001,  | lambda[ 55] = -4.44560183538e-007 |
| c[ 56] = 2.09316584162e-001,  | lambda[ 56] = -5.00427266975e-007 |
| c[ 57] = 9.99431867372e-002,  | lambda[ 57] = -8.00090911293e-007 |
| c[ 58] = 1.05144870448e-001,  | lambda[ 58] = -2.42211534358e-006 |
| c[ 59] = 8.17246529906e-002,  | lambda[ 59] = -2.82237342243e-006 |
| c[ 60] = -4.63373889797e-002, | lambda[ 60] = 3.99854703234e-009  |
| c[ 61] = 3.72765689680e-001,  | lambda[ 61] = 1.70755920386e-003  |
| c[ 62] = 2.09316584162e-001,  | lambda[ 62] = 6.21749104809e-004  |
| c[ 63] = 9.99431867372e-002,  | lambda[ 63] = 7.04062116750e-003  |
| c[ 64] = 1.05144870448e-001,  | lambda[ 64] = 2.96614502433e-006  |
| c[ 65] = 8.17246529906e-002,  | lambda[ 65] = 2.52724067782e-006  |

### Solution Vector

|                               |                                   |
|-------------------------------|-----------------------------------|
| x[ 0] = 1.62599352470e-001,   | lambda[ 66] = 0.00000000000e+000  |
| x[ 1] = 1.09999856850e+000,   | lambda[ 67] = 2.79424442031e-001  |
| x[ 2] = 1.08010496922e+000,   | lambda[ 68] = 1.70310789392e-005  |
| x[ 3] = 1.05651999768e+000,   | lambda[ 69] = 5.44446715023e-006  |
| x[ 4] = 1.04676188184e+000,   | lambda[ 70] = 3.37955364998e-006  |
| x[ 5] = 1.04795390537e+000,   | lambda[ 71] = 3.60193644421e-006  |
| x[ 6] = 1.04475519944e+000,   | lambda[ 72] = 3.01909523520e-006  |
| x[ 7] = 1.03818220092e+000,   | lambda[ 73] = 1.93456404161e-006  |
| x[ 8] = 1.04495504057e+000,   | lambda[ 74] = 3.05426625903e-006  |
| x[ 9] = 1.08010489420e+000,   | lambda[ 75] = 1.70310008413e-005  |
| x[ 10] = 1.07692254742e+000,  | lambda[ 76] = 1.41814035606e-005  |
| x[ 11] = 1.09998706710e+000,  | lambda[ 77] = 3.09261575589e-002  |
| x[ 12] = 1.08958625649e+000,  | lambda[ 78] = 3.55451714937e-005  |
| x[ 13] = 1.09998769234e+000,  | lambda[ 79] = 3.24973957092e-002  |
| x[ 14] = 1.07517105037e+000,  | lambda[ 80] = 1.29145971819e-005  |
| x[ 15] = 1.07068362724e+000,  | lambda[ 81] = 1.03297998999e-005  |
| x[ 16] = 1.07681320297e+000,  | lambda[ 82] = 1.40969482224e-005  |
| x[ 17] = 1.07203971700e+000,  | lambda[ 83] = 1.10283828785e-005  |
| x[ 18] = 1.06111467526e+000,  | lambda[ 84] = 6.68677161340e-006  |
| x[ 19] = 1.05848923708e+000,  | lambda[ 85] = 5.94905260512e-006  |
| x[ 20] = 1.06234125720e+000,  | lambda[ 86] = 7.06112263745e-006  |
| x[ 21] = 1.06514686980e+000,  | lambda[ 87] = 8.00290500968e-006  |
| x[ 22] = 1.06574003385e+000,  | lambda[ 88] = 8.21941193388e-006  |
| x[ 23] = 1.06066331152e+000,  | lambda[ 89] = 6.55405575984e-006  |
| x[ 24] = 1.05543732029e+000,  | lambda[ 90] = 5.18239685657e-006  |
| x[ 25] = 1.05225008757e+000,  | lambda[ 91] = 4.46500057682e-006  |
| x[ 26] = 1.03518112387e+000,  | lambda[ 92] = 1.47516556831e-006  |
| x[ 27] = 1.05851721331e+000,  | lambda[ 93] = 5.95650070560e-006  |
| x[ 28] = 1.04089005334e+000,  | lambda[ 94] = 2.36612832924e-006  |
| x[ 29] = 1.03940740165e+000,  | lambda[ 95] = 2.12756334024e-006  |
| x[ 30] = 1.02835175487e+000,  | lambda[ 96] = 4.77647917551e-007  |
| x[ 31] = -8.54857949237e-002, | lambda[ 97] = 0.00000000000e+000  |
| x[ 32] = -1.20337049914e-001, | lambda[ 98] = 0.00000000000e+000  |
| x[ 33] = -1.48448567839e-001, | lambda[ 99] = 0.00000000000e+000  |
| x[ 34] = -2.28653282194e-001, | lambda[ 100] = 0.00000000000e+000 |
| x[ 35] = -1.77259926318e-001, | lambda[ 101] = 0.00000000000e+000 |
| x[ 36] = -2.06793242211e-001, | lambda[ 102] = 0.00000000000e+000 |
| x[ 37] = -1.89611210941e-001, | lambda[ 103] = 0.00000000000e+000 |

|                               |                                   |
|-------------------------------|-----------------------------------|
| x[ 38] = -2.26969436351e-001, | lambda[ 104] = 0.00000000000e+000 |
| x[ 39] = -2.53038671643e-001, | lambda[ 105] = 0.00000000000e+000 |
| x[ 40] = -2.26969436351e-001, | lambda[ 106] = 0.00000000000e+000 |
| x[ 41] = -2.40993979532e-001, | lambda[ 107] = 0.00000000000e+000 |
| x[ 42] = -2.40993979532e-001, | lambda[ 108] = 0.00000000000e+000 |
| x[ 43] = -2.55622043421e-001, | lambda[ 109] = 0.00000000000e+000 |
| x[ 44] = -2.57099103290e-001, | lambda[ 110] = 0.00000000000e+000 |
| x[ 45] = -2.50374545994e-001, | lambda[ 111] = 0.00000000000e+000 |
| x[ 46] = -2.55764062992e-001, | lambda[ 112] = 0.00000000000e+000 |
| x[ 47] = -2.67057583466e-001, | lambda[ 113] = 0.00000000000e+000 |
| x[ 48] = -2.69838111978e-001, | lambda[ 114] = 0.00000000000e+000 |
| x[ 49] = -2.66580484075e-001, | lambda[ 115] = 0.00000000000e+000 |
| x[ 50] = -2.60348476215e-001, | lambda[ 116] = 0.00000000000e+000 |
| x[ 51] = -2.60129196783e-001, | lambda[ 117] = 0.00000000000e+000 |
| x[ 52] = -2.63518660230e-001, | lambda[ 118] = 0.00000000000e+000 |
| x[ 53] = -2.66433117451e-001, | lambda[ 119] = 0.00000000000e+000 |
| x[ 54] = -2.59337020729e-001, | lambda[ 120] = 0.00000000000e+000 |
| x[ 55] = -2.66175418886e-001, | lambda[ 121] = 0.00000000000e+000 |
| x[ 56] = -2.50697602250e-001, | lambda[ 122] = 0.00000000000e+000 |
| x[ 57] = -1.87351448175e-001, | lambda[ 123] = 0.00000000000e+000 |
| x[ 58] = -2.70730775210e-001, | lambda[ 124] = 0.00000000000e+000 |
| x[ 59] = -2.85079363509e-001, | lambda[ 125] = 0.00000000000e+000 |

=====

### A.3 Arquivo de saída para o sistema IEEE 118 barras – Estratégia (i)

A seguir são apresentados os valores obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange para o sistema IEEE 118 barras no teste realizado utilizando a estratégia (i) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 289$ ;
- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 118$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 119, \dots, 235$  e
- $lambda[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 525$ .

| Constraint Vector            | Lagrange Multipliers             |
|------------------------------|----------------------------------|
| -----                        | -----                            |
| c[ 0] = -1.72146841004e-007, | lambda[ 0] = -1.00000000000e+000 |
| c[ 1] = 5.09999999719e-001,  | lambda[ 1] = -9.90884414476e-002 |
| c[ 2] = 1.99999999796e-001,  | lambda[ 2] = -9.30619784500e-002 |
| c[ 3] = 3.89999999806e-001,  | lambda[ 3] = -8.88236065522e-002 |
| c[ 4] = 3.89999998831e-001,  | lambda[ 4] = -5.79218715947e-002 |
| c[ 5] = -1.35878419627e-009, | lambda[ 5] = -5.49999764442e-002 |

|                               |                                   |
|-------------------------------|-----------------------------------|
| c[ 6] = 5.1999999835e-001,    | lambda[ 6] = -7.53257078869e-002  |
| c[ 7] = 1.89999999758e-001,   | lambda[ 7] = -7.88729533288e-002  |
| c[ 8] = 2.79999998673e-001,   | lambda[ 8] = -5.55477130766e-002  |
| c[ 9] = -5.12850206746e-010,  | lambda[ 9] = -3.66208825028e-002  |
| c[ 10] = -4.4999999962e+000,  | lambda[ 10] = -1.72895792629e-002 |
| c[ 11] = 6.99999999116e-001,  | lambda[ 11] = -8.00102858817e-002 |
| c[ 12] = -3.79999999877e-001, | lambda[ 12] = -8.24339071294e-002 |
| c[ 13] = 3.39999998430e-001,  | lambda[ 13] = -9.39934658671e-002 |
| c[ 14] = 1.39999998610e-001,  | lambda[ 14] = -8.91962201962e-002 |
| c[ 15] = 8.99999984811e-001,  | lambda[ 15] = -9.26429539075e-002 |
| c[ 16] = 2.49999998385e-001,  | lambda[ 16] = -8.48645343326e-002 |
| c[ 17] = 1.09999981424e-001,  | lambda[ 17] = -6.93475458853e-002 |
| c[ 18] = 5.99999986784e-001,  | lambda[ 18] = -8.86856707195e-002 |
| c[ 19] = 4.50000016663e-001,  | lambda[ 19] = -9.42534801027e-002 |
| c[ 20] = 1.79999997085e-001,  | lambda[ 20] = -9.38435225203e-002 |
| c[ 21] = 1.39999998998e-001,  | lambda[ 21] = -8.70968500977e-002 |
| c[ 22] = 9.9999996648e-002,   | lambda[ 22] = -7.36611532190e-002 |
| c[ 23] = 6.9999999241e-002,   | lambda[ 23] = -4.59030492439e-002 |
| c[ 24] = 1.29999999242e-001,  | lambda[ 24] = -5.13374672312e-002 |
| c[ 25] = -2.20000000032e+000, | lambda[ 25] = -8.94574288275e-003 |
| c[ 26] = -3.14000000003e+000, | lambda[ 26] = -1.48821820520e-002 |
| c[ 27] = 7.09999998875e-001,  | lambda[ 27] = -8.12961815743e-002 |
| c[ 28] = 1.6999999202e-001,   | lambda[ 28] = -9.16316201786e-002 |
| c[ 29] = 2.39999997969e-001,  | lambda[ 29] = -9.70878998930e-002 |
| c[ 30] = 1.33251276679e-009,  | lambda[ 30] = -6.28007963929e-002 |
| c[ 31] = 3.5999999998e-001,   | lambda[ 31] = -9.48336646456e-002 |
| c[ 32] = 5.89999996270e-001,  | lambda[ 32] = -8.46740258428e-002 |
| c[ 33] = 2.29999996681e-001,  | lambda[ 33] = -9.37936184534e-002 |
| c[ 34] = 5.89999968413e-001,  | lambda[ 34] = -8.23626346108e-002 |
| c[ 35] = 3.29999990903e-001,  | lambda[ 35] = -8.57486934110e-002 |
| c[ 36] = 3.0999999064e-001,   | lambda[ 36] = -8.61479314722e-002 |
| c[ 37] = -1.31094599620e-008, | lambda[ 37] = -7.72619790929e-002 |
| c[ 38] = -2.93812263408e-009, | lambda[ 38] = -6.89714946709e-002 |
| c[ 39] = 2.6999999986e-001,   | lambda[ 39] = -1.16241827390e-001 |
| c[ 40] = 6.6000000046e-001,   | lambda[ 40] = -1.29463620860e-001 |
| c[ 41] = 3.6999999972e-001,   | lambda[ 41] = -1.35861200946e-001 |
| c[ 42] = 9.5999999940e-001,   | lambda[ 42] = -1.23682126657e-001 |
| c[ 43] = 1.7999999909e-001,   | lambda[ 43] = -9.86458594911e-002 |
| c[ 44] = 1.5999999945e-001,   | lambda[ 44] = -1.02310440259e-001 |
| c[ 45] = 5.2999999970e-001,   | lambda[ 45] = -9.67191436759e-002 |
| c[ 46] = 8.9999999906e-002,   | lambda[ 46] = -7.23732383073e-002 |
| c[ 47] = 3.3999999998e-001,   | lambda[ 47] = -5.36789938547e-002 |
| c[ 48] = 2.0000000000e-001,   | lambda[ 48] = -5.61233752288e-002 |
| c[ 49] = -1.17000000054e+000, | lambda[ 49] = -4.49537103298e-002 |
| c[ 50] = 1.69999999365e-001,  | lambda[ 50] = -6.58482141290e-002 |
| c[ 51] = 1.69999999443e-001,  | lambda[ 51] = -9.31321260547e-002 |
| c[ 52] = 1.79999999827e-001,  | lambda[ 52] = -1.02852177728e-001 |
| c[ 53] = 2.29999999892e-001,  | lambda[ 53] = -1.06439055290e-001 |
| c[ 54] = 6.49999957723e-001,  | lambda[ 54] = -9.62211435794e-002 |
| c[ 55] = 6.30000006809e-001,  | lambda[ 55] = -9.74108459725e-002 |
| c[ 57] = 1.19999999326e-001,  | lambda[ 57] = -8.86605847501e-002 |
| c[ 58] = 1.19999999216e-001,  | lambda[ 58] = -9.84416352898e-002 |
| c[ 59] = 1.21999999468e+000,  | lambda[ 59] = -5.04189951895e-002 |
| c[ 60] = 7.7999999999e-001,   | lambda[ 60] = -3.30163302972e-002 |
| c[ 61] = -1.5999999999e+000,  | lambda[ 61] = -2.80666009470e-002 |
| c[ 62] = 7.7000000000e-001,   | lambda[ 62] = -3.26355815987e-002 |
| c[ 63] = 1.73585590346e-011,  | lambda[ 63] = -4.02253741597e-002 |
| c[ 64] = -3.42748052162e-012, | lambda[ 64] = -2.95982789247e-002 |
| c[ 65] = -3.91000000003e+000, | lambda[ 65] = -1.34237230588e-002 |
| c[ 66] = -3.53000000001e+000, | lambda[ 66] = -7.95004727595e-003 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 67] = 2.80000000001e-001,   | lambda[ 67] = -2.48924169893e-002  |
| c[ 68] = -3.83693077310e-013,  | lambda[ 68] = -9.26949550322e-003  |
| c[ 69] = 6.60000000649e-001,   | lambda[ 69] = -6.04960209287e-002  |
| c[ 70] = -9.14677222852e-010,  | lambda[ 70] = -6.15054136003e-002  |
| c[ 71] = 1.19999999846e-001,   | lambda[ 71] = -6.16932236866e-002  |
| c[ 72] = 5.99999995743e-002,   | lambda[ 72] = -6.24562783306e-002  |
| c[ 73] = 6.79999999676e-001,   | lambda[ 73] = -7.89192041406e-002  |
| c[ 74] = 4.69999999735e-001,   | lambda[ 74] = -6.83007618059e-002  |
| c[ 75] = 6.7999999919e-001,    | lambda[ 75] = -8.26363803036e-002  |
| c[ 76] = 6.1000000012e-001,    | lambda[ 76] = -3.04121459658e-002  |
| c[ 77] = 7.0999999963e-001,    | lambda[ 77] = -3.26492190743e-002  |
| c[ 78] = 3.8999999970e-001,    | lambda[ 78] = -2.86785920464e-002  |
| c[ 79] = -3.4700000013e+000,   | lambda[ 79] = -6.92637427138e-003  |
| c[ 80] = -2.43799425093e-012,  | lambda[ 80] = -7.41318826391e-003  |
| c[ 81] = 5.3999999978e-001,    | lambda[ 81] = -2.41170701282e-002  |
| c[ 82] = 1.9999999999e-001,    | lambda[ 82] = -1.32437341151e-002  |
| c[ 83] = 1.1000000000e-001,    | lambda[ 83] = 1.15346255801e-002   |
| c[ 84] = 2.3999999996e-001,    | lambda[ 84] = 2.90125631120e-002   |
| c[ 85] = 2.0999999997e-001,    | lambda[ 85] = 1.92160473035e-002   |
| c[ 86] = -3.9999999975e-002,   | lambda[ 86] = 2.12672780103e-002   |
| c[ 87] = 4.7999999999e-001,    | lambda[ 87] = 4.41140208540e-002   |
| c[ 88] = -6.0700000000e+000,   | lambda[ 88] = 6.51679459274e-002   |
| c[ 89] = 1.6300000000e+000,    | lambda[ 89] = 2.17098733044e-002   |
| c[ 90] = 1.0000000000e-001,    | lambda[ 90] = 2.22186747457e-002   |
| c[ 91] = 6.5000000000e-001,    | lambda[ 91] = 2.94051243325e-002   |
| c[ 92] = 1.2000000000e-001,    | lambda[ 92] = 4.96367413056e-003   |
| c[ 93] = 2.9999999998e-001,    | lambda[ 93] = -1.17962363743e-002  |
| c[ 94] = 4.1999999999e-001,    | lambda[ 94] = -2.13876295875e-002  |
| c[ 95] = 3.7999999994e-001,    | lambda[ 95] = -2.07084325051e-002  |
| c[ 96] = 1.4999999999e-001,    | lambda[ 96] = -1.62411146029e-002  |
| c[ 97] = 3.4000000000e-001,    | lambda[ 97] = -1.75719687634e-002  |
| c[ 98] = 4.1999999999e-001,    | lambda[ 98] = -2.00315091316e-002  |
| c[ 99] = -2.1500000004e+000,   | lambda[ 99] = -1.24254459178e-002  |
| c[ 100] = 2.2000000000e-001,   | lambda[ 100] = -1.11457201008e-003 |
| c[ 101] = 4.9999999999e-002,   | lambda[ 101] = 1.86001614188e-002  |
| c[ 102] = -1.70000000486e-001, | lambda[ 102] = -4.38219456657e-002 |
| c[ 103] = 3.7999999991e-001,   | lambda[ 103] = -6.65230153794e-002 |
| c[ 104] = 3.09999999524e-001,  | lambda[ 104] = -7.61896407984e-002 |
| c[ 105] = 4.3000000001e-001,   | lambda[ 105] = -7.84031930734e-002 |
| c[ 106] = 5.0000000000e-001,   | lambda[ 106] = -1.02845212916e-001 |
| c[ 107] = 1.99999997515e-002,  | lambda[ 107] = -8.56439393203e-002 |
| c[ 108] = 7.9999997361e-002,   | lambda[ 108] = -8.90515793754e-002 |
| c[ 109] = 3.90000000423e-001,  | lambda[ 109] = -9.38846181355e-002 |
| c[ 110] = -3.60000001254e-001, | lambda[ 110] = -7.91859441651e-002 |
| c[ 111] = 6.7999998241e-001,   | lambda[ 111] = -1.27154449115e-001 |
| c[ 112] = 6.00000141326e-002,  | lambda[ 112] = -7.21447157716e-002 |
| c[ 113] = 7.9999996248e-002,   | lambda[ 113] = -8.72748899900e-002 |
| c[ 114] = 2.1999999144e-001,   | lambda[ 114] = -8.73729775617e-002 |
| c[ 115] = 1.8400000000e+000,   | lambda[ 115] = -1.04142105134e-002 |
| c[ 116] = 1.9999999810e-001,   | lambda[ 116] = -9.48371357214e-002 |
| c[ 117] = 3.2999999927e-001,   | lambda[ 117] = -8.00783594862e-002 |
| c[ 118] = 9.00000000150e-002,  | lambda[ 118] = -4.44385628866e-003 |
| c[ 119] = 1.00000000019e-001,  | lambda[ 119] = -6.24934960495e-003 |
| c[ 120] = -1.62323487984e-010, | lambda[ 120] = -1.77236313847e-003 |
| c[ 121] = 2.00000000436e-002,  | lambda[ 121] = -8.79617457224e-005 |
| c[ 122] = 6.56612542116e-011,  | lambda[ 122] = 3.75673204333e-003  |
| c[ 123] = 2.30000000118e-001,  | lambda[ 123] = -2.65338487467e-003 |
| c[ 124] = 1.60000000202e-001,  | lambda[ 124] = -5.98085001043e-003 |
| c[ 125] = 1.00000002809e-002,  | lambda[ 125] = 8.20985321570e-004  |
| c[ 126] = 1.00000000217e-001,  | lambda[ 126] = -1.51262232318e-003 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 127] = 3.00000044708e-002,  | lambda[ 127] = 6.62200794312e-004  |
| c[ 128] = 3.00000001566e-002,  | lambda[ 128] = -3.40143661007e-003 |
| c[ 129] = 7.99999997784e-002,  | lambda[ 129] = -5.55520913146e-003 |
| c[ 130] = 4.99999998489e-002,  | lambda[ 130] = -5.14093527486e-003 |
| c[ 131] = 2.99999998211e-002,  | lambda[ 131] = -7.45320626179e-004 |
| c[ 132] = 6.9999999895e-002,   | lambda[ 132] = -1.50261535204e-003 |
| c[ 133] = 4.00000003633e-002,  | lambda[ 133] = -8.73194482814e-004 |
| c[ 134] = -5.94717164404e-010, | lambda[ 134] = -6.95777840413e-005 |
| c[ 135] = 9.00000006448e-002,  | lambda[ 135] = -1.82407128715e-003 |
| c[ 136] = 9.00000006162e-002,  | lambda[ 136] = -2.62269218428e-004 |
| c[ 137] = 1.02187858175e-009,  | lambda[ 137] = 2.25534267349e-004  |
| c[ 138] = -2.25995222536e-010, | lambda[ 138] = -1.29955671327e-003 |
| c[ 139] = 1.10000000001e-001,  | lambda[ 139] = -2.67221059865e-003 |
| c[ 140] = 1.00000000001e-001,  | lambda[ 140] = -1.66841256151e-003 |
| c[ 141] = 6.9999999974e-002,   | lambda[ 141] = 3.33729778440e-004  |
| c[ 142] = 7.9999999968e-002,   | lambda[ 142] = 1.80163338491e-003  |
| c[ 143] = 2.1999999997e-001,   | lambda[ 143] = -1.14369190064e-003 |
| c[ 144] = -1.06936681732e-012, | lambda[ 144] = 2.09424174185e-004  |
| c[ 145] = 1.10000000000e-001,  | lambda[ 145] = 2.34972321649e-003  |
| c[ 146] = 4.00000000896e-002,  | lambda[ 146] = -1.48628674197e-003 |
| c[ 147] = 8.00000000895e-002,  | lambda[ 147] = -5.45716672528e-003 |
| c[ 148] = 5.00000000397e-002,  | lambda[ 148] = -5.94628893701e-003 |
| c[ 149] = 1.10000000015e-001,  | lambda[ 149] = -4.74674971520e-003 |
| c[ 150] = 3.00000000895e-002,  | lambda[ 150] = -1.09640160117e-003 |
| c[ 151] = 3.00000001279e-002,  | lambda[ 151] = -3.39909576336e-003 |
| c[ 152] = 2.9999999999e-002,   | lambda[ 152] = -9.51278465941e-005 |
| c[ 153] = 2.13162820728e-013,  | lambda[ 153] = 1.05583508640e-005  |
| c[ 154] = 1.84741111298e-013,  | lambda[ 154] = 7.52369343273e-004  |
| c[ 155] = 6.9999999998e-002,   | lambda[ 155] = 3.27342879651e-003  |
| c[ 156] = -1.70530256582e-013, | lambda[ 156] = 9.04239014106e-007  |
| c[ 157] = 1.46094691900e-010,  | lambda[ 157] = 2.70290206603e-004  |
| c[ 158] = 1.10000000028e-001,  | lambda[ 158] = -6.85854800834e-003 |
| c[ 159] = 2.60000000006e-001,  | lambda[ 159] = -2.31596568576e-003 |
| c[ 160] = 3.20000000001e-001,  | lambda[ 160] = -2.46003347274e-003 |
| c[ 161] = 7.10542735760e-014,  | lambda[ 161] = -4.19679231667e-004 |
| c[ 162] = 2.70000000004e-001,  | lambda[ 162] = -3.42294591694e-003 |
| c[ 163] = 9.9999999997e-002,   | lambda[ 163] = -3.51482712555e-003 |
| c[ 164] = 7.00000000000e-002,  | lambda[ 164] = -6.66077686140e-003 |
| c[ 165] = 9.9999999994e-002,   | lambda[ 165] = -4.16054633617e-003 |
| c[ 166] = 1.00000000000e-001,  | lambda[ 166] = -4.02663898849e-003 |
| c[ 167] = 7.00000000000e-002,  | lambda[ 167] = -7.63013259029e-003 |
| c[ 168] = 1.60000000000e-001,  | lambda[ 168] = -7.23346124086e-003 |
| c[ 169] = 3.10000000000e-001,  | lambda[ 169] = -1.08217029903e-002 |
| c[ 170] = 1.50000000001e-001,  | lambda[ 170] = -6.01622364025e-003 |
| c[ 171] = 9.00000000000e-002,  | lambda[ 171] = -4.12166680576e-003 |
| c[ 172] = 8.00000000000e-002,  | lambda[ 172] = -1.06055941295e-003 |
| c[ 173] = 1.50000000000e-001,  | lambda[ 173] = -5.90663856802e-003 |
| c[ 174] = 3.00000000000e-002,  | lambda[ 174] = -5.33434048313e-003 |
| c[ 175] = 1.59999999999e-001,  | lambda[ 175] = -2.98650575655e-003 |
| c[ 176] = 1.00000000512e-002,  | lambda[ 176] = -4.53060381739e-004 |
| c[ 177] = 3.00000000511e-002,  | lambda[ 177] = -6.66038920898e-004 |
| c[ 178] = 3.00000000054e-002,  | lambda[ 178] = -1.05758686867e-003 |
| c[ 179] = 7.00000000174e-002,  | lambda[ 179] = -1.15405839253e-003 |
| c[ 180] = 7.9999999989e-002,   | lambda[ 180] = -3.99728667280e-003 |
| c[ 181] = 1.50000000004e-001,  | lambda[ 181] = -1.15386191225e-002 |
| c[ 182] = -1.20010540448e-001, | lambda[ 182] = -4.00021082454e-007 |
| c[ 183] = 5.95302309865e-001,  | lambda[ 183] = -2.15326239336e-008 |
| c[ 184] = 1.22908779669e-001,  | lambda[ 184] = -1.69165974186e-007 |
| c[ 185] = -7.23226029713e-001, | lambda[ 185] = -3.51374336334e-008 |
| c[ 186] = -1.05255593209e+000, | lambda[ 186] = -1.91644912728e-007 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 187] = 4.58285043518e-001,  | lambda[ 187] = -8.80781108004e-008 |
| c[ 188] = -3.62178252535e-002, | lambda[ 188] = -2.19915654003e-007 |
| c[ 189] = -1.54620964995e-003, | lambda[ 189] = -1.60498847397e-007 |
| c[ 190] = -1.05109579481e-002, | lambda[ 190] = -2.50550731099e-007 |
| c[ 191] = -7.47066239853e-002, | lambda[ 191] = -2.73476858157e-008 |
| c[ 192] = -4.69980997829e-001, | lambda[ 192] = -4.21022177935e-003 |
| c[ 193] = -3.18274685243e-001, | lambda[ 193] = -8.26299291586e-009 |
| c[ 194] = 1.40501512175e-001,  | lambda[ 194] = -2.44610802543e-008 |
| c[ 195] = 8.86179736357e-004,  | lambda[ 195] = -2.44582051840e-008 |
| c[ 196] = -4.39527225658e-003, | lambda[ 196] = -2.18815517218e-007 |
| c[ 197] = -2.83371203745e-001, | lambda[ 197] = -1.71466860403e-006 |
| c[ 198] = 5.54613727225e-002,  | lambda[ 198] = -2.62157316175e-007 |
| c[ 199] = 1.04163317840e-001,  | lambda[ 199] = -2.39940256479e-008 |
| c[ 200] = 1.06057948818e-001,  | lambda[ 200] = -2.39803987786e-008 |
| c[ 201] = -1.62776642285e-001, | lambda[ 201] = -8.53585239562e-008 |
| c[ 202] = -3.62728142841e-002, | lambda[ 202] = -7.18309041851e-008 |
| c[ 203] = 5.58220281834e-002,  | lambda[ 203] = -2.36991340459e-008 |
| c[ 204] = -2.25697142883e-003, | lambda[ 204] = -2.68915680802e-007 |
| c[ 205] = -3.11488940940e-002, | lambda[ 205] = -3.49811564366e-007 |
| c[ 206] = -5.74094309501e-002, | lambda[ 206] = -4.78300742882e-008 |
| c[ 207] = 3.18908505774e-001,  | lambda[ 207] = -6.06562172559e-008 |
| c[ 208] = -2.07293118172e-001, | lambda[ 208] = -6.02833243401e-007 |
| c[ 209] = -6.69860020681e-001, | lambda[ 209] = -5.71511600030e-004 |
| c[ 210] = -8.49990374645e-001, | lambda[ 210] = -8.31138427043e-003 |
| c[ 211] = -1.10458501945e+000, | lambda[ 211] = -4.22072099369e-008 |
| c[ 212] = 9.10325253782e-002,  | lambda[ 212] = -2.04605732181e-007 |
| c[ 213] = -5.73164198791e-002, | lambda[ 213] = -8.48641140654e-008 |
| c[ 214] = -2.37458561909e-002, | lambda[ 214] = -8.19461333867e-008 |
| c[ 215] = -1.80014108483e-001, | lambda[ 215] = -5.33383628340e-007 |
| c[ 216] = -1.30006588897e-001, | lambda[ 216] = -2.58070002360e-007 |
| c[ 217] = 4.19850303157e-001,  | lambda[ 217] = -8.89041769433e-008 |
| c[ 218] = -4.14319073117e-001, | lambda[ 218] = -5.34874606638e-008 |
| c[ 219] = 7.99836849617e-002,  | lambda[ 219] = -2.58078096249e-007 |
| c[ 220] = 3.51612555428e-002,  | lambda[ 220] = -7.72826480341e-008 |
| c[ 221] = -4.37548204099e-001, | lambda[ 221] = -4.81217000776e-008 |
| c[ 222] = 5.45329830526e-002,  | lambda[ 222] = -2.30246770984e-008 |
| c[ 223] = 3.74105764549e-002,  | lambda[ 223] = -7.71150804090e-008 |
| c[ 224] = -1.00183599758e-002, | lambda[ 224] = -6.66768703575e-007 |
| c[ 225] = -3.62698436997e-002, | lambda[ 225] = -8.30107944753e-008 |
| c[ 226] = 3.55733735727e-001,  | lambda[ 226] = -7.72400252281e-008 |
| c[ 227] = 3.85467301083e-002,  | lambda[ 227] = -2.29527975039e-007 |
| c[ 228] = -2.01386123163e-002, | lambda[ 228] = -2.58194120762e-007 |
| c[ 229] = -1.19407442702e-001, | lambda[ 229] = -3.62700184060e-007 |
| c[ 230] = -7.92186125835e-002, | lambda[ 230] = -3.92006711012e-008 |
| c[ 231] = -7.02243201961e-002, | lambda[ 231] = -2.58302460479e-007 |
| c[ 232] = 6.18337411425e-003,  | lambda[ 232] = -7.95086668191e-008 |
| c[ 233] = -2.98843420687e-004, | lambda[ 233] = -7.08153529344e-008 |
| c[ 234] = -7.78660303519e-002, | lambda[ 234] = -8.67570093825e-008 |
| c[ 235] = -7.95514846595e-002, | lambda[ 235] = -8.06415149533e-009 |
| c[ 236] = -1.20010540448e-001, | lambda[ 236] = 7.58983138124e-003  |
| c[ 237] = 5.95302309865e-001,  | lambda[ 237] = 3.50158072508e-008  |
| c[ 238] = 1.22908779669e-001,  | lambda[ 238] = 5.09265697904e-007  |
| c[ 239] = -7.23226029713e-001, | lambda[ 239] = 2.14867422203e-008  |
| c[ 240] = -1.05255593209e+000, | lambda[ 240] = 2.62075517489e-008  |
| c[ 241] = 4.58285043518e-001,  | lambda[ 241] = 1.24666046545e-007  |
| c[ 242] = -3.62178252535e-002, | lambda[ 242] = 2.22142866938e-006  |
| c[ 243] = -1.54620964995e-003, | lambda[ 243] = 4.95279836965e-007  |
| c[ 244] = -1.05109579481e-002, | lambda[ 244] = 2.27391691747e-004  |
| c[ 245] = -7.47066239853e-002, | lambda[ 245] = 2.60187437863e-008  |
| c[ 246] = -4.69980997829e-001, | lambda[ 246] = 4.27811833610e-008  |

|                                |                                   |
|--------------------------------|-----------------------------------|
| c[ 247] = -3.18274685243e-001, | lambda[ 247] = 7.75323645958e-009 |
| c[ 248] = 1.40501512175e-001,  | lambda[ 248] = 2.93094135508e-008 |
| c[ 249] = 8.86179736357e-004,  | lambda[ 249] = 2.93135472919e-008 |
| c[ 250] = -4.39527225658e-003, | lambda[ 250] = 4.11532375122e-007 |
| c[ 251] = -2.83371203745e-001, | lambda[ 251] = 3.03575971750e-007 |
| c[ 252] = 5.54613727225e-002,  | lambda[ 252] = 1.11259956625e-005 |
| c[ 253] = 1.04163317840e-001,  | lambda[ 253] = 3.00093403183e-008 |
| c[ 254] = 1.06057948818e-001,  | lambda[ 254] = 3.00306832508e-008 |
| c[ 255] = -1.62776642285e-001, | lambda[ 255] = 7.52745203055e-008 |
| c[ 256] = -3.62728142841e-002, | lambda[ 256] = 4.35665172444e-008 |
| c[ 257] = 5.58220281834e-002,  | lambda[ 257] = 3.04879674436e-008 |
| c[ 258] = -2.25697142883e-003, | lambda[ 258] = 1.27510966331e-005 |
| c[ 259] = -3.11488940940e-002, | lambda[ 259] = 1.25527057700e-004 |
| c[ 260] = -5.74094309501e-002, | lambda[ 260] = 1.09980321362e-007 |
| c[ 261] = 3.18908505774e-001,  | lambda[ 261] = 2.98385938733e-008 |
| c[ 262] = -2.07293118172e-001, | lambda[ 262] = 2.99296968913e-007 |
| c[ 263] = -6.69860020681e-001, | lambda[ 263] = 2.99641175196e-008 |
| c[ 264] = -8.49990374645e-001, | lambda[ 264] = 2.99626548310e-008 |
| c[ 265] = -1.10458501945e+000, | lambda[ 265] = 1.94904077220e-008 |
| c[ 266] = 9.10325253782e-002,  | lambda[ 266] = 2.90470034943e-006 |
| c[ 267] = -5.73164198791e-002, | lambda[ 267] = 7.56632584114e-008 |
| c[ 268] = -2.37458561909e-002, | lambda[ 268] = 7.81446058836e-008 |
| c[ 269] = -1.80014108483e-001, | lambda[ 269] = 5.67129023278e-003 |
| c[ 270] = -1.30006588897e-001, | lambda[ 270] = 1.21418664457e-002 |
| c[ 271] = 4.19850303157e-001,  | lambda[ 271] = 5.36669843382e-004 |
| c[ 272] = -4.14319073117e-001, | lambda[ 272] = 2.70790119711e-008 |
| c[ 273] = 7.99836849617e-002,  | lambda[ 273] = 4.90344996137e-003 |
| c[ 274] = 3.51612555428e-002,  | lambda[ 274] = 8.02822826190e-009 |
| c[ 275] = -4.37548204099e-001, | lambda[ 275] = 2.32724013425e-008 |
| c[ 276] = 5.45329830526e-002,  | lambda[ 276] = 3.16773093722e-008 |
| c[ 277] = 3.74105764549e-002,  | lambda[ 277] = 8.31091616751e-008 |
| c[ 278] = -1.00183599758e-002, | lambda[ 278] = 4.35730430267e-003 |
| c[ 279] = -3.62698436997e-002, | lambda[ 279] = 7.71999745516e-008 |
| c[ 280] = 3.55733735727e-001,  | lambda[ 280] = 7.88748358366e-008 |
| c[ 281] = 3.85467301083e-002,  | lambda[ 281] = 3.97130038475e-007 |
| c[ 282] = -2.01386123163e-002, | lambda[ 282] = 5.82446175539e-004 |
| c[ 283] = -1.19407442702e-001, | lambda[ 283] = 8.95194418347e-007 |
| c[ 284] = -7.92186125835e-002, | lambda[ 284] = 4.08326051213e-008 |
| c[ 285] = -7.02243201961e-002, | lambda[ 285] = 3.93090116006e-004 |
| c[ 286] = 6.18337411425e-003,  | lambda[ 286] = 8.00495005309e-009 |
| c[ 287] = -2.98843420687e-004, | lambda[ 287] = 8.10512464180e-009 |
| c[ 288] = -7.78660303519e-002, | lambda[ 288] = 3.85011954469e-008 |
| c[ 289] = -7.95514846595e-002, | lambda[ 289] = 7.93686106583e-009 |

#### Solution Vector

|                              |                                   |
|------------------------------|-----------------------------------|
| x[ 0] = 1.07868087315e+000,  | lambda[ 290] = 0.00000000000e+000 |
| x[ 1] = 1.06899478841e+000,  | lambda[ 291] = 2.10686202878e-006 |
| x[ 2] = 1.07720176593e+000,  | lambda[ 292] = 3.05771992118e-006 |
| x[ 3] = 1.07572200366e+000,  | lambda[ 293] = 2.83996279275e-006 |
| x[ 4] = 1.09999769129e+000,  | lambda[ 294] = 3.46376594691e-002 |
| x[ 5] = 1.09756064292e+000,  | lambda[ 295] = 3.24028526319e-005 |
| x[ 6] = 1.09290222428e+000,  | lambda[ 296] = 1.08614420056e-005 |
| x[ 7] = 1.09108929395e+000,  | lambda[ 297] = 8.56211588295e-006 |
| x[ 8] = 1.08093844834e+000,  | lambda[ 298] = 3.75482097785e-006 |
| x[ 9] = 1.09999959157e+000,  | lambda[ 299] = 1.95872326111e-001 |
| x[ 10] = 1.09315711543e+000, | lambda[ 300] = 1.12771191704e-005 |
| x[ 11] = 1.08606139565e+000, | lambda[ 301] = 5.30995352998e-006 |
| x[ 12] = 1.08966974759e+000, | lambda[ 302] = 7.32462571829e-006 |
| x[ 13] = 1.07486338052e+000, | lambda[ 303] = 2.72531708931e-006 |

|                              |                                   |
|------------------------------|-----------------------------------|
| x[ 14] = 1.08843914677e+000, | lambda[ 304] = 6.49913172809e-006 |
| x[ 15] = 1.08742703197e+000, | lambda[ 305] = 5.94786667907e-006 |
| x[ 16] = 1.08635481636e+000, | lambda[ 306] = 5.43552454983e-006 |
| x[ 17] = 1.09996370928e+000, | lambda[ 307] = 4.41528847033e-003 |
| x[ 18] = 1.08945601510e+000, | lambda[ 308] = 7.18271075917e-006 |
| x[ 19] = 1.08602843998e+000, | lambda[ 309] = 5.30972818561e-006 |
| x[ 20] = 1.07486068423e+000, | lambda[ 310] = 2.72605984479e-006 |
| x[ 21] = 1.07020984622e+000, | lambda[ 311] = 2.21594989906e-006 |
| x[ 22] = 1.07356769151e+000, | lambda[ 312] = 2.56609565735e-006 |
| x[ 23] = 1.08827846617e+000, | lambda[ 313] = 6.40124501820e-006 |
| x[ 24] = 1.08637423326e+000, | lambda[ 314] = 5.44242000484e-006 |
| x[ 25] = 1.0999977213e+000,  | lambda[ 315] = 3.51095943208e-001 |
| x[ 26] = 1.06435503811e+000, | lambda[ 316] = 1.75762055388e-006 |
| x[ 27] = 1.08026356711e+000, | lambda[ 317] = 3.61032414600e-006 |
| x[ 28] = 1.07440069054e+000, | lambda[ 318] = 2.66679650206e-006 |
| x[ 29] = 1.07518507420e+000, | lambda[ 319] = 2.76783824851e-006 |
| x[ 30] = 1.06874089689e+000, | lambda[ 320] = 2.08520078375e-006 |
| x[ 31] = 1.07820989064e+000, | lambda[ 321] = 3.22352933824e-006 |
| x[ 32] = 1.07978908632e+000, | lambda[ 322] = 3.51418687946e-006 |
| x[ 33] = 1.08601565150e+000, | lambda[ 323] = 5.29285181316e-006 |
| x[ 34] = 1.09651144894e+000, | lambda[ 324] = 2.25299501381e-005 |
| x[ 35] = 1.09441742423e+000, | lambda[ 325] = 1.39454104510e-005 |
| x[ 36] = 1.09472443526e+000, | lambda[ 326] = 1.47999433711e-005 |
| x[ 37] = 1.09999859159e+000, | lambda[ 327] = 5.69795910653e-002 |
| x[ 38] = 1.05063808072e+000, | lambda[ 328] = 1.08960969612e-006 |
| x[ 39] = 1.07670635586e+000, | lambda[ 329] = 2.98168052323e-006 |
| x[ 40] = 1.07364232391e+000, | lambda[ 330] = 2.57444708769e-006 |
| x[ 41] = 1.06703789441e+000, | lambda[ 331] = 1.94809372793e-006 |
| x[ 42] = 1.07247713085e+000, | lambda[ 332] = 2.44283982984e-006 |
| x[ 43] = 1.08099644002e+000, | lambda[ 333] = 3.76774053110e-006 |
| x[ 44] = 1.07140453301e+000, | lambda[ 334] = 2.33091124021e-006 |
| x[ 45] = 1.06497815717e+000, | lambda[ 335] = 1.79937494107e-006 |
| x[ 46] = 1.07593035670e+000, | lambda[ 336] = 2.86896215057e-006 |
| x[ 47] = 1.08306676161e+000, | lambda[ 337] = 4.28743495356e-006 |
| x[ 48] = 1.08576610026e+000, | lambda[ 338] = 5.18973181650e-006 |
| x[ 49] = 1.08741600195e+000, | lambda[ 339] = 5.93041655080e-006 |
| x[ 50] = 1.07651269628e+000, | lambda[ 340] = 2.95289522281e-006 |
| x[ 51] = 1.05923065582e+000, | lambda[ 341] = 1.45985190167e-006 |
| x[ 52] = 1.05372325950e+000, | lambda[ 342] = 1.20831900298e-006 |
| x[ 53] = 1.05304607731e+000, | lambda[ 343] = 1.18108036646e-006 |
| x[ 54] = 1.06696769246e+000, | lambda[ 344] = 1.94273373992e-006 |
| x[ 55] = 1.06595326032e+000, | lambda[ 345] = 1.86784288589e-006 |
| x[ 56] = 1.06606037180e+000, | lambda[ 346] = 1.87546460263e-006 |
| x[ 57] = 1.06786530896e+000, | lambda[ 347] = 2.01300237927e-006 |
| x[ 58] = 1.06018623322e+000, | lambda[ 348] = 1.50996122226e-006 |
| x[ 59] = 1.08750790110e+000, | lambda[ 349] = 5.97733704374e-006 |
| x[ 60] = 1.08407416744e+000, | lambda[ 350] = 4.58866976051e-006 |
| x[ 61] = 1.08635098216e+000, | lambda[ 351] = 5.43191695679e-006 |
| x[ 62] = 1.08138073739e+000, | lambda[ 352] = 3.85556276677e-006 |
| x[ 63] = 1.05351596320e+000, | lambda[ 353] = 1.19990216950e-006 |
| x[ 64] = 1.06027291918e+000, | lambda[ 354] = 1.51459095461e-006 |
| x[ 65] = 1.05258440163e+000, | lambda[ 355] = 1.16290842981e-006 |
| x[ 66] = 1.09999986230e+000, | lambda[ 356] = 5.80954855325e-001 |
| x[ 67] = 1.08574672047e+000, | lambda[ 357] = 5.18204837160e-006 |
| x[ 68] = 1.05120490788e+000, | lambda[ 358] = 1.11042574576e-006 |
| x[ 69] = 1.09999964311e+000, | lambda[ 359] = 2.24159366068e-001 |
| x[ 70] = 1.07744670550e+000, | lambda[ 360] = 3.09656162271e-006 |
| x[ 71] = 1.07781792069e+000, | lambda[ 361] = 3.15671793662e-006 |
| x[ 72] = 1.07898062549e+000, | lambda[ 362] = 3.35910500590e-006 |
| x[ 73] = 1.07661919438e+000, | lambda[ 363] = 2.96866387701e-006 |

|                                |                                   |
|--------------------------------|-----------------------------------|
| x[ 74] = 1.05885713754e+000,   | lambda[ 364] = 1.44086554891e-006 |
| x[ 75] = 1.06163235209e+000,   | lambda[ 365] = 1.59015448473e-006 |
| x[ 76] = 1.04638309376e+000,   | lambda[ 366] = 9.45559392339e-007 |
| x[ 77] = 1.08677560686e+000,   | lambda[ 367] = 5.62122196589e-006 |
| x[ 78] = 1.08253797782e+000,   | lambda[ 368] = 4.14314480875e-006 |
| x[ 79] = 1.08421181388e+000,   | lambda[ 369] = 4.63282852619e-006 |
| x[ 80] = 1.09999881349e+000,   | lambda[ 370] = 6.74250354860e-002 |
| x[ 81] = 1.04816692023e+000,   | lambda[ 371] = 1.00348432615e-006 |
| x[ 82] = 1.07931163861e+000,   | lambda[ 372] = 3.42076697313e-006 |
| x[ 83] = 1.08133870002e+000,   | lambda[ 373] = 3.84579531025e-006 |
| x[ 84] = 1.08451969358e+000,   | lambda[ 374] = 4.73431681972e-006 |
| x[ 85] = 1.09249277723e+000,   | lambda[ 375] = 1.02409740137e-005 |
| x[ 86] = 1.08727050081e+000,   | lambda[ 376] = 5.85755831726e-006 |
| x[ 87] = 1.09998824446e+000,   | lambda[ 377] = 6.80566972773e-003 |
| x[ 88] = 1.08888429230e+000,   | lambda[ 378] = 6.77349035921e-006 |
| x[ 89] = 1.09999983992e+000,   | lambda[ 379] = 4.99750633812e-001 |
| x[ 90] = 1.08377289817e+000,   | lambda[ 380] = 4.49470406561e-006 |
| x[ 91] = 1.08823191863e+000,   | lambda[ 381] = 6.37304224435e-006 |
| x[ 92] = 1.09051128165e+000,   | lambda[ 382] = 8.01114248141e-006 |
| x[ 93] = 1.08068331873e+000,   | lambda[ 383] = 3.69873483625e-006 |
| x[ 94] = 1.07915121221e+000,   | lambda[ 384] = 3.39060385016e-006 |
| x[ 95] = 1.06846581431e+000,   | lambda[ 385] = 2.06205583917e-006 |
| x[ 96] = 1.07679292043e+000,   | lambda[ 386] = 2.99471870632e-006 |
| x[ 97] = 1.08419368308e+000,   | lambda[ 387] = 4.62694386854e-006 |
| x[ 98] = 1.09302461830e+000,   | lambda[ 388] = 1.10544426552e-005 |
| x[ 99] = 1.09530169568e+000,   | lambda[ 389] = 1.66179766888e-005 |
| x[ 100] = 1.09999848882e+000,  | lambda[ 390] = 5.29388172735e-002 |
| x[ 101] = 1.08391336442e+000,  | lambda[ 391] = 4.53808410286e-006 |
| x[ 102] = 1.08737245213e+000,  | lambda[ 392] = 5.90839759207e-006 |
| x[ 103] = 1.09146384793e+000,  | lambda[ 393] = 8.95399924812e-006 |
| x[ 104] = 1.08189531443e+000,  | lambda[ 394] = 3.97910378822e-006 |
| x[ 105] = 1.07920854979e+000,  | lambda[ 395] = 3.40133355223e-006 |
| x[ 106] = 1.07216969479e+000,  | lambda[ 396] = 2.40990790634e-006 |
| x[ 107] = 1.06663244417e+000,  | lambda[ 397] = 1.91744134410e-006 |
| x[ 108] = 1.07590416905e+000,  | lambda[ 398] = 2.86536016379e-006 |
| x[ 109] = 1.07472429417e+000,  | lambda[ 399] = 2.70733491543e-006 |
| x[ 110] = 1.07476417854e+000,  | lambda[ 400] = 2.71258184290e-006 |
| x[ 111] = 1.08304106500e+000,  | lambda[ 401] = 4.28022005287e-006 |
| x[ 112] = 1.06019188521e+000,  | lambda[ 402] = 1.51024924781e-006 |
| x[ 113] = 1.09590577131e+000,  | lambda[ 403] = 1.96195046784e-005 |
| x[ 114] = 1.07567696864e+000,  | lambda[ 404] = 2.83412182560e-006 |
| x[ 115] = 1.07533390900e+000,  | lambda[ 405] = 2.78745784721e-006 |
| x[ 116] = 1.05062775619e+000,  | lambda[ 406] = 1.08923292659e-006 |
| x[ 117] = 1.07556508606e+000,  | lambda[ 407] = 2.81846680080e-006 |
| x[ 118] = 1.04868213665e+000,  | lambda[ 408] = 1.02085606792e-006 |
| x[ 119] = -2.82119161922e-001, | lambda[ 409] = 0.00000000000e+000 |
| x[ 120] = -2.72881836398e-001, | lambda[ 410] = 0.00000000000e+000 |
| x[ 121] = -2.68352537013e-001, | lambda[ 411] = 0.00000000000e+000 |
| x[ 122] = -2.14341453857e-001, | lambda[ 412] = 0.00000000000e+000 |
| x[ 123] = -2.06682952555e-001, | lambda[ 413] = 0.00000000000e+000 |
| x[ 124] = -2.47150154936e-001, | lambda[ 414] = 0.00000000000e+000 |
| x[ 125] = -2.53237447167e-001, | lambda[ 415] = 0.00000000000e+000 |
| x[ 126] = -1.31874543861e-001, | lambda[ 416] = 0.00000000000e+000 |
| x[ 127] = -1.85215418825e-002, | lambda[ 417] = 0.00000000000e+000 |
| x[ 128] = 1.02960279655e-001,  | lambda[ 418] = 0.00000000000e+000 |
| x[ 129] = -2.50718288296e-001, | lambda[ 419] = 0.00000000000e+000 |
| x[ 130] = -2.57852125021e-001, | lambda[ 420] = 0.00000000000e+000 |
| x[ 131] = -2.71372385423e-001, | lambda[ 421] = 0.00000000000e+000 |
| x[ 132] = -2.69322332994e-001, | lambda[ 422] = 0.00000000000e+000 |
| x[ 133] = -2.76431439713e-001, | lambda[ 423] = 0.00000000000e+000 |

|                                |                                  |
|--------------------------------|----------------------------------|
| x[ 134] = -2.62817986123e-001, | lambda[ 424] = 0.0000000000e+000 |
| x[ 135] = -2.38041644908e-001, | lambda[ 425] = 0.0000000000e+000 |
| x[ 136] = -2.71797518064e-001, | lambda[ 426] = 0.0000000000e+000 |
| x[ 137] = -2.80056938614e-001, | lambda[ 427] = 0.0000000000e+000 |
| x[ 138] = -2.66480788051e-001, | lambda[ 428] = 0.0000000000e+000 |
| x[ 139] = -2.43200441385e-001, | lambda[ 429] = 0.0000000000e+000 |
| x[ 140] = -2.05719000616e-001, | lambda[ 430] = 0.0000000000e+000 |
| x[ 141] = -1.32008332262e-001, | lambda[ 431] = 0.0000000000e+000 |
| x[ 142] = -1.35567058837e-001, | lambda[ 432] = 0.0000000000e+000 |
| x[ 143] = -2.01303316860e-002, | lambda[ 433] = 0.0000000000e+000 |
| x[ 144] = 7.53378941307e-003,  | lambda[ 434] = 0.0000000000e+000 |
| x[ 145] = -2.17090271961e-001, | lambda[ 435] = 0.0000000000e+000 |
| x[ 146] = -2.41152083446e-001, | lambda[ 436] = 0.0000000000e+000 |
| x[ 147] = -2.54841825781e-001, | lambda[ 437] = 0.0000000000e+000 |
| x[ 148] = -1.64425250949e-001, | lambda[ 438] = 0.0000000000e+000 |
| x[ 149] = -2.53082293341e-001, | lambda[ 439] = 0.0000000000e+000 |
| x[ 150] = -2.25813525748e-001, | lambda[ 440] = 0.0000000000e+000 |
| x[ 151] = -2.85323207323e-001, | lambda[ 441] = 0.0000000000e+000 |
| x[ 152] = -2.76774560956e-001, | lambda[ 442] = 0.0000000000e+000 |
| x[ 153] = -2.83434891023e-001, | lambda[ 443] = 0.0000000000e+000 |
| x[ 154] = -2.83604899008e-001, | lambda[ 444] = 0.0000000000e+000 |
| x[ 155] = -2.69719397265e-001, | lambda[ 445] = 0.0000000000e+000 |
| x[ 156] = -1.94939228677e-001, | lambda[ 446] = 0.0000000000e+000 |
| x[ 157] = -3.18026741919e-001, | lambda[ 447] = 0.0000000000e+000 |
| x[ 158] = -3.33489018394e-001, | lambda[ 448] = 0.0000000000e+000 |
| x[ 159] = -3.39629602430e-001, | lambda[ 449] = 0.0000000000e+000 |
| x[ 160] = -3.15585960177e-001, | lambda[ 450] = 0.0000000000e+000 |
| x[ 161] = -2.77629048064e-001, | lambda[ 451] = 0.0000000000e+000 |
| x[ 162] = -2.41235299885e-001, | lambda[ 452] = 0.0000000000e+000 |
| x[ 163] = -2.13413000045e-001, | lambda[ 453] = 0.0000000000e+000 |
| x[ 164] = -1.70201006280e-001, | lambda[ 454] = 0.0000000000e+000 |
| x[ 165] = -1.35769685760e-001, | lambda[ 455] = 0.0000000000e+000 |
| x[ 166] = -1.46892052153e-001, | lambda[ 456] = 0.0000000000e+000 |
| x[ 167] = -1.30645293825e-001, | lambda[ 457] = 0.0000000000e+000 |
| x[ 168] = -1.63998251815e-001, | lambda[ 458] = 0.0000000000e+000 |
| x[ 169] = -2.05550136145e-001, | lambda[ 459] = 0.0000000000e+000 |
| x[ 170] = -2.20228592826e-001, | lambda[ 460] = 0.0000000000e+000 |
| x[ 171] = -2.35485635585e-001, | lambda[ 461] = 0.0000000000e+000 |
| x[ 172] = -2.23327260652e-001, | lambda[ 462] = 0.0000000000e+000 |
| x[ 173] = -2.27657275707e-001, | lambda[ 463] = 0.0000000000e+000 |
| x[ 174] = -2.24765448544e-001, | lambda[ 464] = 0.0000000000e+000 |
| x[ 175] = -2.05106059925e-001, | lambda[ 465] = 0.0000000000e+000 |
| x[ 176] = -2.18088731701e-001, | lambda[ 466] = 0.0000000000e+000 |
| x[ 177] = -1.60926959337e-001, | lambda[ 467] = 0.0000000000e+000 |
| x[ 178] = -1.03408661617e-001, | lambda[ 468] = 0.0000000000e+000 |
| x[ 179] = -9.03870205883e-002, | lambda[ 469] = 0.0000000000e+000 |
| x[ 180] = -9.80242444976e-002, | lambda[ 470] = 0.0000000000e+000 |
| x[ 181] = -1.10597194083e-001, | lambda[ 471] = 0.0000000000e+000 |
| x[ 182] = -8.30529215439e-002, | lambda[ 472] = 0.0000000000e+000 |
| x[ 183] = -3.10009304759e-002, | lambda[ 473] = 0.0000000000e+000 |
| x[ 184] = -3.00906026116e-002, | lambda[ 474] = 0.0000000000e+000 |
| x[ 185] = -7.38454938477e-002, | lambda[ 475] = 0.0000000000e+000 |
| x[ 186] = -3.41116704396e-002, | lambda[ 476] = 0.0000000000e+000 |
| x[ 187] = -1.14922644598e-001, | lambda[ 477] = 0.0000000000e+000 |
| x[ 188] = -1.20260863903e-001, | lambda[ 478] = 0.0000000000e+000 |
| x[ 189] = -1.37200169379e-001, | lambda[ 479] = 0.0000000000e+000 |
| x[ 190] = -1.22482069791e-001, | lambda[ 480] = 0.0000000000e+000 |
| x[ 191] = -1.30109280262e-001, | lambda[ 481] = 0.0000000000e+000 |
| x[ 192] = -1.10374113578e-001, | lambda[ 482] = 0.0000000000e+000 |
| x[ 193] = -1.27728450666e-001, | lambda[ 483] = 0.0000000000e+000 |

|                                |                                  |
|--------------------------------|----------------------------------|
| x[ 194] = -4.99825768910e-002, | lambda[ 484] = 0.0000000000e+000 |
| x[ 195] = -5.38845421019e-002, | lambda[ 485] = 0.0000000000e+000 |
| x[ 196] = -4.83666719356e-002, | lambda[ 486] = 0.0000000000e+000 |
| x[ 197] = -1.05836744482e-002, | lambda[ 487] = 0.0000000000e+000 |
| x[ 198] = -2.49941510249e-002, | lambda[ 488] = 0.0000000000e+000 |
| x[ 199] = -4.17382401979e-002, | lambda[ 489] = 0.0000000000e+000 |
| x[ 200] = -2.49860000870e-002, | lambda[ 490] = 0.0000000000e+000 |
| x[ 201] = 1.05461714243e-002,  | lambda[ 491] = 0.0000000000e+000 |
| x[ 202] = 3.23491290266e-002,  | lambda[ 492] = 0.0000000000e+000 |
| x[ 203] = 1.46206093451e-002,  | lambda[ 493] = 0.0000000000e+000 |
| x[ 204] = 2.00891741876e-002,  | lambda[ 494] = 0.0000000000e+000 |
| x[ 205] = 7.90503910210e-002,  | lambda[ 495] = 0.0000000000e+000 |
| x[ 206] = 1.39140548086e-001,  | lambda[ 496] = 0.0000000000e+000 |
| x[ 207] = 4.61894062914e-002,  | lambda[ 497] = 0.0000000000e+000 |
| x[ 208] = 4.40016401066e-002,  | lambda[ 498] = 0.0000000000e+000 |
| x[ 209] = 5.36736243397e-002,  | lambda[ 499] = 0.0000000000e+000 |
| x[ 210] = 1.11117845301e-002,  | lambda[ 500] = 0.0000000000e+000 |
| x[ 211] = -1.93633262742e-002, | lambda[ 501] = 0.0000000000e+000 |
| x[ 212] = -3.34502967096e-002, | lambda[ 502] = 0.0000000000e+000 |
| x[ 213] = -3.56092023263e-002, | lambda[ 503] = 0.0000000000e+000 |
| x[ 214] = -2.86173017621e-002, | lambda[ 504] = 0.0000000000e+000 |
| x[ 215] = -3.54528695559e-002, | lambda[ 505] = 0.0000000000e+000 |
| x[ 216] = -4.29109156180e-002, | lambda[ 506] = 0.0000000000e+000 |
| x[ 217] = -2.71856846256e-002, | lambda[ 507] = 0.0000000000e+000 |
| x[ 218] = -5.50714666170e-003, | lambda[ 508] = 0.0000000000e+000 |
| x[ 219] = 3.25032992602e-002,  | lambda[ 509] = 0.0000000000e+000 |
| x[ 220] = -8.10300232964e-002, | lambda[ 510] = 0.0000000000e+000 |
| x[ 221] = -1.24843567869e-001, | lambda[ 511] = 0.0000000000e+000 |
| x[ 222] = -1.41216764083e-001, | lambda[ 512] = 0.0000000000e+000 |
| x[ 223] = -1.43835737111e-001, | lambda[ 513] = 0.0000000000e+000 |
| x[ 224] = -1.83557703838e-001, | lambda[ 514] = 0.0000000000e+000 |
| x[ 225] = -1.56715629232e-001, | lambda[ 515] = 0.0000000000e+000 |
| x[ 226] = -1.62540375518e-001, | lambda[ 516] = 0.0000000000e+000 |
| x[ 227] = -1.73026782188e-001, | lambda[ 517] = 0.0000000000e+000 |
| x[ 228] = -1.50013071637e-001, | lambda[ 518] = 0.0000000000e+000 |
| x[ 229] = -2.11979127408e-001, | lambda[ 519] = 0.0000000000e+000 |
| x[ 230] = -2.37516324758e-001, | lambda[ 520] = 0.0000000000e+000 |
| x[ 231] = -2.30023553859e-001, | lambda[ 521] = 0.0000000000e+000 |
| x[ 232] = -2.30060149856e-001, | lambda[ 522] = 0.0000000000e+000 |
| x[ 233] = -4.08624984532e-002, | lambda[ 523] = 0.0000000000e+000 |
| x[ 234] = -2.80080095444e-001, | lambda[ 524] = 0.0000000000e+000 |
| x[ 235] = -1.25127805003e-001, | lambda[ 525] = 0.0000000000e+000 |

=====

## A.4 Arquivo de saída para o sistema IEEE 14 barras – Estratégia (iv)

A seguir são apresentados os valores finais obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange no teste realizado para o Sistema IEEE 14 barras utilizando a estratégia (iv) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 32$ ;
- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 14$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 15, \dots, 32$  e
- $\text{lambda}[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 32$ .

| Constraint Vector             | Lagrange Multipliers              |
|-------------------------------|-----------------------------------|
| -----                         | -----                             |
| c[ 0] = -4.56087229095e-009,  | lambda[ 0] = -1.00000000000e+000  |
| c[ 1] = -1.83000001601e-001,  | lambda[ 1] = -5.07009721814e-002  |
| c[ 2] = 9.41999999012e-001,   | lambda[ 2] = -1.26357070076e-001  |
| c[ 3] = 4.78000002963e-001,   | lambda[ 3] = -1.02626650689e-001  |
| c[ 4] = 7.60000008591e-002,   | lambda[ 4] = -8.60685087251e-002  |
| c[ 5] = 1.11999998713e-001,   | lambda[ 5] = -8.80009248868e-002  |
| c[ 6] = 1.48725137761e-009,   | lambda[ 6] = -1.02095782419e-001  |
| c[ 7] = 0.00000000000e+000,   | lambda[ 7] = -1.02095782419e-001  |
| c[ 8] = 2.94999997843e-001,   | lambda[ 8] = -1.01788654714e-001  |
| c[ 9] = 8.99999991577e-002,   | lambda[ 9] = -1.05782127914e-001  |
| c[ 10] = 3.49999997063e-002,  | lambda[ 10] = -1.00717540748e-001 |
| c[ 11] = 6.09999996290e-002,  | lambda[ 11] = -1.06786707712e-001 |
| c[ 12] = 1.34999999472e-001,  | lambda[ 12] = -1.12512177860e-001 |
| c[ 13] = 1.48999998622e-001,  | lambda[ 13] = -1.30427821370e-001 |
| c[ 14] = -3.90000004787e-002, | lambda[ 15] = -2.14342860004e-003 |
| c[ 16] = 1.36886058044e-011,  | lambda[ 16] = -4.41274195552e-007 |
| c[ 17] = 1.65999998505e-001,  | lambda[ 17] = 2.90957169403e-004  |
| c[ 18] = 5.79999999051e-002,  | lambda[ 18] = -4.49316103567e-003 |
| c[ 19] = 1.79999999817e-002,  | lambda[ 19] = -6.93608646090e-003 |
| c[ 20] = 1.60000001378e-002,  | lambda[ 20] = -1.19315850670e-002 |
| c[ 21] = 5.79999999349e-002,  | lambda[ 21] = -1.42986486751e-002 |
| c[ 22] = 5.00000001861e-002,  | lambda[ 22] = -1.44441256184e-002 |
| c[ 23] = -2.42179914693e-001, | lambda[ 23] = -1.00252692471e-010 |
| c[ 24] = 2.23080314668e-001,  | lambda[ 24] = -1.33251453411e-008 |
| c[ 25] = 9.99664600217e-003,  | lambda[ 25] = -5.00092249252e-008 |
| c[ 26] = 1.64998359954e-001,  | lambda[ 26] = -3.33337508260e-008 |
| c[ 27] = 2.11446569446e-001,  | lambda[ 27] = -3.69038414635e-008 |
| c[ 28] = -2.42179914693e-001, | lambda[ 28] = 9.97684819423e-011  |
| c[ 29] = 2.23080314668e-001,  | lambda[ 29] = 6.68720907474e-008  |
| c[ 30] = 9.99664600217e-003,  | lambda[ 30] = 2.96868623596e-003  |
| c[ 31] = 1.64998359954e-001,  | lambda[ 31] = 6.09186403762e-003  |
| c[ 32] = 2.11446569446e-001,  | lambda[ 32] = 3.47590995339e-007  |

### Solution Vector

```
-----
x[ 0] = 1.23614758450e-001,      lambda[ 33] = 0.00000000000e+000
x[ 1] = 1.09999995981e+000,      lambda[ 34] = 2.48823044655e-001
x[ 2] = 1.08658316833e+000,      lambda[ 35] = 6.72051684671e-007
x[ 3] = 1.05282726560e+000,      lambda[ 36] = 1.14631210305e-007
x[ 4] = 1.06513869780e+000,      lambda[ 37] = 1.99741444152e-007
x[ 5] = 1.07073449422e+000,      lambda[ 38] = 2.58588163180e-007
x[ 6] = 1.03345010142e+000,      lambda[ 39] = 3.02675251730e-008
x[ 7] = 1.05202107390e+000,      lambda[ 40] = 1.09950938255e-007
x[ 8] = 1.08630813483e+000,      lambda[ 41] = 6.49886917893e-007
x[ 9] = 1.03741259374e+000,      lambda[ 42] = 4.51024653906e-008
x[10] = 1.02917352651e+000,      lambda[ 43] = 1.46460084944e-008
x[11] = 1.02783099266e+000,      lambda[ 44] = 9.88066132632e-009
x[12] = 1.01944991166e+000,      lambda[ 45] = -2.00116936735e-008
x[13] = 1.01588292130e+000,      lambda[ 46] = -3.30843302367e-008
x[14] = 1.00958774947e+000,      lambda[ 47] = -5.74674009009e-008
x[15] = -8.03634043784e-002,      lambda[ 48] = 0.00000000000e+000
x[16] = -2.04428819082e-001,      lambda[ 49] = 0.00000000000e+000
x[17] = -1.67390861957e-001,      lambda[ 50] = 0.00000000000e+000
x[18] = -1.42968464379e-001,      lambda[ 51] = 0.00000000000e+000
x[19] = -2.42209056761e-001,      lambda[ 52] = 0.00000000000e+000
x[20] = -2.24772129329e-001,      lambda[ 53] = 0.00000000000e+000
x[21] = -2.24772129329e-001,      lambda[ 54] = 0.00000000000e+000
x[22] = -2.55071131686e-001,      lambda[ 55] = 0.00000000000e+000
x[23] = -2.58197434728e-001,      lambda[ 56] = 0.00000000000e+000
x[24] = -2.52837851154e-001,      lambda[ 57] = 0.00000000000e+000
x[25] = -2.57924869264e-001,      lambda[ 58] = 0.00000000000e+000
x[26] = -2.59686854014e-001,      lambda[ 59] = 0.00000000000e+000
x[27] = -2.75477945398e-001,      lambda[ 60] = 0.00000000000e+000
=====
```

## A.5 Arquivo de saída para o sistema IEEE 30 barras – Estratégia (iv)

A seguir são apresentados os valores obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange para o sistema IEEE 30 barras no teste realizado utilizando a estratégia (iv) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 65$ ;
- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 30$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 31, \dots, 59$  e
- $\text{lambda}[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 125$ .

Constraint Vector

Lagrange Multipliers

-----  
c[ 0] = -1.64262756730e-007,  
c[ 1] = -1.83000010604e-001,  
c[ 2] = 2.40000001743e-002,  
c[ 3] = 7.59999963470e-002,  
c[ 4] = 9.41999962979e-001,  
c[ 5] = -1.55250834410e-008,  
c[ 6] = 2.27999953897e-001,  
c[ 7] = 2.99999992076e-001,  
c[ 8] = -2.11766876257e-009,  
c[ 9] = 5.79999986892e-002,  
c[ 10] = 0.00000000000e+000,  
c[ 11] = 1.11999994429e-001,  
c[ 12] = 0.00000000000e+000,  
c[ 13] = 6.19999996388e-002,  
c[ 14] = 8.1999998112e-002,  
c[ 15] = 3.5000000588e-002,  
c[ 16] = 8.9999991234e-002,  
c[ 17] = 3.19999998152e-002,  
c[ 18] = 9.49999993918e-002,  
c[ 19] = 2.19999998159e-002,  
c[ 20] = 1.74999998782e-001,  
c[ 21] = -8.66151594892e-011,  
c[ 22] = 3.19999997673e-002,  
c[ 23] = 8.69999992090e-002,  
c[ 24] = -1.49984469289e-010,  
c[ 25] = 3.49999996350e-002,  
c[ 26] = 4.47515691171e-010,  
c[ 27] = -9.43657596508e-010,  
c[ 28] = 2.39999996485e-002,  
c[ 29] = 1.05999998110e-001,  
c[ 30] = 1.19999998271e-002,  
c[ 31] = 1.60000012906e-002,  
c[ 32] = 5.04236652432e-009,  
c[ 33] = 1.09000008784e-001,  
c[ 34] = 4.39808189867e-011,  
c[ 35] = 1.99999989435e-002,  
c[ 36] = 7.5000000441e-002,  
c[ 37] = 1.60000001832e-002,  
c[ 38] = 2.49999998880e-002,  
c[ 39] = 1.79999998362e-002,  
c[ 40] = 5.80000000895e-002,  
c[ 41] = 9.00000003112e-003,  
c[ 42] = 3.40000000851e-002,  
c[ 43] = 7.00000000159e-003,  
c[ 44] = 1.11999999928e-001,  
c[ 45] = 4.63007410190e-012,  
c[ 46] = 1.60000000239e-002,  
c[ 47] = 6.70000000449e-002,  
c[ 48] = 2.62030397380e-011,  
c[ 49] = 2.29999999652e-002,  
c[ 50] = -9.83293446666e-010,  
c[ 51] = 6.76838141089e-010,  
c[ 52] = 9.00000002598e-003,  
c[ 53] = 1.90000004226e-002,  
c[ 54] = -4.70255983732e-002,  
c[ 55] = 3.72993620021e-001,  
c[ 56] = 2.09833348389e-001,

-----  
lambda[ 0] = -1.00000000000e+000  
lambda[ 1] = -5.80789748208e-002  
lambda[ 2] = -7.77003733059e-002  
lambda[ 3] = -1.00717911084e-001  
lambda[ 4] = -1.44887409630e-001  
lambda[ 5] = -1.18428265864e-001  
lambda[ 6] = -1.36025213017e-001  
lambda[ 7] = -1.25912073915e-001  
lambda[ 8] = -1.18987966486e-001  
lambda[ 9] = -1.19238960730e-001  
lambda[ 10] = -1.18987966486e-001  
lambda[ 11] = -1.03606580851e-001  
lambda[ 12] = -1.03606580851e-001  
lambda[ 13] = -1.21141379990e-001  
lambda[ 14] = -1.26824837254e-001  
lambda[ 15] = -1.17538768410e-001  
lambda[ 16] = -1.22300081438e-001  
lambda[ 17] = -1.39086763516e-001  
lambda[ 18] = -1.42232972417e-001  
lambda[ 19] = -1.37161320896e-001  
lambda[ 20] = -1.30279008621e-001  
lambda[ 21] = -1.29861056770e-001  
lambda[ 22] = -1.37414464671e-001  
lambda[ 23] = -1.42810073881e-001  
lambda[ 24] = -1.36942021224e-001  
lambda[ 25] = -1.56616979835e-001  
lambda[ 26] = -1.25347341902e-001  
lambda[ 27] = -1.25222755071e-001  
lambda[ 28] = -1.54750247546e-001  
lambda[ 29] = -1.75042132475e-001  
lambda[ 30] = -1.16708842720e-002  
lambda[ 31] = -1.15101889191e-002  
lambda[ 32] = -9.12008356506e-003  
lambda[ 33] = -1.01376944595e-002  
lambda[ 34] = -5.95634061579e-003  
lambda[ 35] = -7.49554288205e-003  
lambda[ 36] = -4.18083168109e-003  
lambda[ 37] = -9.48167291978e-003  
lambda[ 38] = -1.23956169855e-002  
lambda[ 39] = -9.66215479484e-003  
lambda[ 40] = -1.05975747771e-002  
lambda[ 41] = -1.58541057427e-002  
lambda[ 42] = -1.66582927042e-002  
lambda[ 43] = -1.46498708848e-002  
lambda[ 44] = -1.40834929638e-002  
lambda[ 45] = -1.36209154346e-002  
lambda[ 46] = -1.70895290254e-002  
lambda[ 47] = -1.88347161131e-002  
lambda[ 48] = -2.19508010435e-002  
lambda[ 49] = -3.50735587748e-002  
lambda[ 50] = -1.78113219458e-002  
lambda[ 51] = -1.06725938267e-002  
lambda[ 52] = -2.59564807331e-002  
lambda[ 53] = -2.92900978787e-002  
lambda[ 54] = -1.00055471576e-010  
lambda[ 55] = -1.11133082953e-008  
lambda[ 56] = -1.25316018196e-008

|                               |                                   |
|-------------------------------|-----------------------------------|
| c[ 57] = 9.99985820025e-002,  | lambda[ 57] = -2.00001356866e-008 |
| c[ 58] = 1.05069853527e-001,  | lambda[ 58] = -6.04878303210e-008 |
| c[ 59] = 8.16445739636e-002,  | lambda[ 59] = -7.04674982665e-008 |
| c[ 60] = -4.70255983732e-002, | lambda[ 60] = 9.99645723774e-011  |
| c[ 61] = 3.72993620021e-001,  | lambda[ 61] = 1.69411872503e-003  |
| c[ 62] = 2.09833348389e-001,  | lambda[ 62] = 5.80784306877e-004  |
| c[ 63] = 9.99985820025e-002,  | lambda[ 63] = 7.03056509379e-003  |
| c[ 64] = 1.05069853527e-001,  | lambda[ 64] = 7.42522602842e-008  |
| c[ 65] = 8.16445739636e-002,  | lambda[ 65] = 6.32551383699e-008  |

#### Solution Vector

|                               |                                   |
|-------------------------------|-----------------------------------|
| x[ 0] = 1.62596921785e-001,   | lambda[ 66] = 0.00000000000e+000  |
| x[ 1] = 1.09999996422e+000,   | lambda[ 67] = 2.79568949406e-001  |
| x[ 2] = 1.08013511079e+000,   | lambda[ 68] = 4.25029958691e-007  |
| x[ 3] = 1.05654476106e+000,   | lambda[ 69] = 1.35933382556e-007  |
| x[ 4] = 1.04679210931e+000,   | lambda[ 70] = 8.43230173351e-008  |
| x[ 5] = 1.04803729831e+000,   | lambda[ 71] = 8.92992636232e-008  |
| x[ 6] = 1.04479266074e+000,   | lambda[ 72] = 7.52674688858e-008  |
| x[ 7] = 1.03823898848e+000,   | lambda[ 73] = 4.79864540219e-008  |
| x[ 8] = 1.04499437225e+000,   | lambda[ 74] = 7.61547299785e-008  |
| x[ 9] = 1.08013191787e+000,   | lambda[ 75] = 4.25278809556e-007  |
| x[ 10] = 1.07695113881e+000,  | lambda[ 76] = 3.54104094641e-007  |
| x[ 11] = 1.09999967780e+000,  | lambda[ 77] = 3.09308631064e-002  |
| x[ 12] = 1.08960856288e+000,  | lambda[ 78] = 8.87827404311e-007  |
| x[ 13] = 1.09999969337e+000,  | lambda[ 79] = 3.25391317252e-002  |
| x[ 14] = 1.07519462018e+000,  | lambda[ 80] = 3.22643224179e-007  |
| x[ 15] = 1.07070818849e+000,  | lambda[ 81] = 2.58058212848e-007  |
| x[ 16] = 1.07683833248e+000,  | lambda[ 82] = 3.52116323664e-007  |
| x[ 17] = 1.07206737045e+000,  | lambda[ 83] = 2.75427275787e-007  |
| x[ 18] = 1.06114095208e+000,  | lambda[ 84] = 1.67032046129e-007  |
| x[ 19] = 1.05851645508e+000,  | lambda[ 85] = 1.48591200230e-007  |
| x[ 20] = 1.06236883594e+000,  | lambda[ 86] = 1.76363041125e-007  |
| x[ 21] = 1.06517592898e+000,  | lambda[ 87] = 1.99857707496e-007  |
| x[ 22] = 1.06576912308e+000,  | lambda[ 88] = 2.05263209984e-007  |
| x[ 23] = 1.06069028607e+000,  | lambda[ 89] = 1.63706390738e-007  |
| x[ 24] = 1.05546731269e+000,  | lambda[ 90] = 1.29406221582e-007  |
| x[ 25] = 1.05228395738e+000,  | lambda[ 91] = 1.11439024802e-007  |
| x[ 26] = 1.03521556323e+000,  | lambda[ 92] = 3.67448401608e-008  |
| x[ 27] = 1.05855321580e+000,  | lambda[ 93] = 1.48648323059e-007  |
| x[ 28] = 1.04092767484e+000,  | lambda[ 94] = 5.89636146793e-008  |
| x[ 29] = 1.03944412000e+000,  | lambda[ 95] = 5.30255274804e-008  |
| x[ 30] = 1.0283888462e+000,   | lambda[ 96] = 1.17917537308e-008  |
| x[ 31] = -8.54923912794e-002, | lambda[ 97] = 0.00000000000e+000  |
| x[ 32] = -1.20338507937e-001, | lambda[ 98] = 0.00000000000e+000  |
| x[ 33] = -1.48449978173e-001, | lambda[ 99] = 0.00000000000e+000  |
| x[ 34] = -2.28660457725e-001, | lambda[ 100] = 0.00000000000e+000 |
| x[ 35] = -1.77261052115e-001, | lambda[ 101] = 0.00000000000e+000 |
| x[ 36] = -2.06795813634e-001, | lambda[ 102] = 0.00000000000e+000 |
| x[ 37] = -1.89611985620e-001, | lambda[ 103] = 0.00000000000e+000 |
| x[ 38] = -2.26967509194e-001, | lambda[ 104] = 0.00000000000e+000 |
| x[ 39] = -2.53035387235e-001, | lambda[ 105] = 0.00000000000e+000 |
| x[ 40] = -2.26967509194e-001, | lambda[ 106] = 0.00000000000e+000 |
| x[ 41] = -2.40989640763e-001, | lambda[ 107] = 0.00000000000e+000 |
| x[ 42] = -2.40989640763e-001, | lambda[ 108] = 0.00000000000e+000 |
| x[ 43] = -2.55617139044e-001, | lambda[ 109] = 0.00000000000e+000 |
| x[ 44] = -2.57094535969e-001, | lambda[ 110] = 0.00000000000e+000 |
| x[ 45] = -2.50370675499e-001, | lambda[ 111] = 0.00000000000e+000 |
| x[ 46] = -2.55760345777e-001, | lambda[ 112] = 0.00000000000e+000 |
| x[ 47] = -2.67052912872e-001, | lambda[ 113] = 0.00000000000e+000 |

|                               |                                   |
|-------------------------------|-----------------------------------|
| x[ 48] = -2.69833526055e-001, | lambda[ 114] = 0.00000000000e+000 |
| x[ 49] = -2.66576186954e-001, | lambda[ 115] = 0.00000000000e+000 |
| x[ 50] = -2.60344778642e-001, | lambda[ 116] = 0.00000000000e+000 |
| x[ 51] = -2.60125505817e-001, | lambda[ 117] = 0.00000000000e+000 |
| x[ 52] = -2.63514206822e-001, | lambda[ 118] = 0.00000000000e+000 |
| x[ 53] = -2.66429067709e-001, | lambda[ 119] = 0.00000000000e+000 |
| x[ 54] = -2.59333803780e-001, | lambda[ 120] = 0.00000000000e+000 |
| x[ 55] = -2.66171754220e-001, | lambda[ 121] = 0.00000000000e+000 |
| x[ 56] = -2.50695124532e-001, | lambda[ 122] = 0.00000000000e+000 |
| x[ 57] = -1.87352029466e-001, | lambda[ 123] = 0.00000000000e+000 |
| x[ 58] = -2.70726908617e-001, | lambda[ 124] = 0.00000000000e+000 |
| x[ 59] = -2.85074472436e-001, | lambda[ 125] = 0.00000000000e+000 |

---

## A.6 Arquivo de saída para o sistema IEEE 118 barras – Estratégia (iv)

A seguir são apresentados os valores obtidos para a função objetivo, tensões, ângulos, restrições e multiplicadores de Lagrange para o sistema IEEE 118 barras no teste realizado utilizando a estratégia (iv) de atualização do parâmetro de barreira, em que:

- $c[j]$  refere-se às restrições, com  $j = 0, \dots, 289$ ;
- $x[0]$  refere-se à função objetivo;
- $x[i]$  refere-se às tensões, com  $i = 1, \dots, 118$ ;
- $x[i]$  refere-se aos ângulos, com  $i = 119, \dots, 235$  e
- $\text{lambda}[k]$  refere-se aos multiplicadores de Lagrange relacionados às restrições, com  $k = 0, \dots, 525$ .

| Constraint Vector             | Lagrange Multipliers              |
|-------------------------------|-----------------------------------|
| -----                         | -----                             |
| c[ 0] = -4.57777672214e-007,  | lambda[ 0] = -1.00000000000e+000  |
| c[ 1] = 5.09999997442e-001,   | lambda[ 1] = -9.90883493366e-002  |
| c[ 2] = 1.9999998064e-001,    | lambda[ 2] = -9.30619339855e-002  |
| c[ 3] = 3.8999997843e-001,    | lambda[ 3] = -8.88235465851e-002  |
| c[ 4] = 3.89999988298e-001,   | lambda[ 4] = -5.79217711972e-002  |
| c[ 5] = -1.35790418909e-008,  | lambda[ 5] = -5.49999374705e-002  |
| c[ 6] = 5.1999999505e-001,    | lambda[ 6] = -7.53257253165e-002  |
| c[ 7] = 1.89999998180e-001,   | lambda[ 7] = -7.88729679319e-002  |
| c[ 8] = 2.79999992102e-001,   | lambda[ 8] = -5.55476324017e-002  |
| c[ 9] = -3.76424402759e-009,  | lambda[ 9] = -3.66208134047e-002  |
| c[ 10] = -4.4999999721e+000,  | lambda[ 10] = -1.72895477093e-002 |
| c[ 11] = 6.9999999111e-001,   | lambda[ 11] = -8.00102313531e-002 |
| c[ 12] = -3.79999992827e-001, | lambda[ 12] = -8.24338851515e-002 |
| c[ 13] = 3.39999990934e-001,  | lambda[ 13] = -9.39933070915e-002 |
| c[ 14] = 1.39999992932e-001,  | lambda[ 14] = -8.91961578128e-002 |

|                               |                                   |
|-------------------------------|-----------------------------------|
| c[ 15] = 8.99999966237e-001,  | lambda[ 15] = -9.26425506169e-002 |
| c[ 16] = 2.49999990112e-001,  | lambda[ 16] = -8.48644063578e-002 |
| c[ 17] = 1.09999877246e-001,  | lambda[ 17] = -6.93471030448e-002 |
| c[ 18] = 5.9999974918e-001,   | lambda[ 18] = -8.86852972405e-002 |
| c[ 19] = 4.50000025892e-001,  | lambda[ 19] = -9.42530622481e-002 |
| c[ 20] = 1.79999987305e-001,  | lambda[ 20] = -9.38432252321e-002 |
| c[ 21] = 1.39999995721e-001,  | lambda[ 21] = -8.70967290500e-002 |
| c[ 22] = 1.00000000095e-001,  | lambda[ 22] = -7.36612215682e-002 |
| c[ 23] = 7.00000030176e-002,  | lambda[ 23] = -4.59031841214e-002 |
| c[ 24] = 1.2999997078e-001,   | lambda[ 24] = -5.13375510848e-002 |
| c[ 25] = -2.2000000432e+000,  | lambda[ 25] = -8.94578786523e-003 |
| c[ 26] = -3.1399999945e+000,  | lambda[ 26] = -1.48822061099e-002 |
| c[ 27] = 7.0999991587e-001,   | lambda[ 27] = -8.12962560779e-002 |
| c[ 28] = 1.6999993863e-001,   | lambda[ 28] = -9.16316728202e-002 |
| c[ 29] = 2.39999985196e-001,  | lambda[ 29] = -9.70878924507e-002 |
| c[ 30] = 1.00611026177e-008,  | lambda[ 30] = -6.28005794657e-002 |
| c[ 31] = 3.5999999150e-001,   | lambda[ 31] = -9.48336109373e-002 |
| c[ 32] = 5.89999977190e-001,  | lambda[ 32] = -8.46740599598e-002 |
| c[ 33] = 2.29999979634e-001,  | lambda[ 33] = -9.37932913600e-002 |
| c[ 34] = 5.89999953480e-001,  | lambda[ 34] = -8.23624038818e-002 |
| c[ 35] = 3.29999994713e-001,  | lambda[ 35] = -8.57483779155e-002 |
| c[ 36] = 3.09999995128e-001,  | lambda[ 36] = -8.61476251157e-002 |
| c[ 37] = -2.45271079002e-008, | lambda[ 37] = -7.72616807183e-002 |
| c[ 38] = -1.95583496065e-008, | lambda[ 38] = -6.89712906601e-002 |
| c[ 39] = 2.6999999914e-001,   | lambda[ 39] = -1.16241517636e-001 |
| c[ 40] = 6.60000000209e-001,  | lambda[ 40] = -1.29463306674e-001 |
| c[ 41] = 3.6999999835e-001,   | lambda[ 41] = -1.35860896233e-001 |
| c[ 42] = 9.5999999680e-001,   | lambda[ 42] = -1.23681863458e-001 |
| c[ 43] = 1.7999999756e-001,   | lambda[ 43] = -9.86458467902e-002 |
| c[ 44] = 1.59999999838e-001,  | lambda[ 44] = -1.02310436200e-001 |
| c[ 45] = 5.2999999902e-001,   | lambda[ 45] = -9.67191075381e-002 |
| c[ 46] = 8.9999999948e-002,   | lambda[ 46] = -7.23732539667e-002 |
| c[ 47] = 3.3999999989e-001,   | lambda[ 47] = -5.36790038407e-002 |
| c[ 48] = 1.9999999990e-001,   | lambda[ 48] = -5.61233903643e-002 |
| c[ 49] = -1.17000000088e+000, | lambda[ 49] = -4.49536643397e-002 |
| c[ 50] = 1.6999999154e-001,   | lambda[ 50] = -6.58481815593e-002 |
| c[ 51] = 1.6999999276e-001,   | lambda[ 51] = -9.31317190710e-002 |
| c[ 52] = 1.7999999811e-001,   | lambda[ 52] = -1.02851643272e-001 |
| c[ 53] = 2.2999999892e-001,   | lambda[ 53] = -1.06438472834e-001 |
| c[ 54] = 6.49999940701e-001,  | lambda[ 54] = -9.62206069041e-002 |
| c[ 55] = 6.30000021397e-001,  | lambda[ 55] = -9.82562398042e-002 |
| c[ 56] = 8.39999953783e-001,  | lambda[ 56] = -9.74102478135e-002 |
| c[ 57] = 1.1999999110e-001,   | lambda[ 57] = -8.86602584344e-002 |
| c[ 58] = 1.1999999027e-001,   | lambda[ 58] = -9.84411020996e-002 |
| c[ 59] = 1.2199999160e+000,   | lambda[ 59] = -5.04191118928e-002 |
| c[ 60] = 7.7999999968e-001,   | lambda[ 60] = -3.30164415780e-002 |
| c[ 61] = -1.5999999996e+000,  | lambda[ 61] = -2.80667125176e-002 |
| c[ 62] = 7.6999999921e-001,   | lambda[ 62] = -3.26356746880e-002 |
| c[ 63] = 1.33137945113e-011,  | lambda[ 63] = -4.02254795070e-002 |
| c[ 64] = -1.72217795580e-011, | lambda[ 64] = -2.95983704000e-002 |
| c[ 65] = -3.91000000041e+000, | lambda[ 65] = -1.34237211413e-002 |
| c[ 66] = -3.53000000002e+000, | lambda[ 66] = -7.94998216751e-003 |
| c[ 67] = 2.7999999996e-001,   | lambda[ 67] = -2.48924598075e-002 |
| c[ 68] = -1.26153310021e-010, | lambda[ 68] = -9.26950048894e-003 |
| c[ 69] = 6.60000002000e-001,  | lambda[ 69] = -6.04957797593e-002 |
| c[ 70] = -1.78770775960e-009, | lambda[ 70] = -6.15052234320e-002 |
| c[ 71] = 1.1999998940e-001,   | lambda[ 71] = -6.16931989107e-002 |
| c[ 72] = 5.9999981744e-002,   | lambda[ 72] = -6.24561005740e-002 |
| c[ 73] = 6.7999998421e-001,   | lambda[ 73] = -7.89186276809e-002 |
| c[ 74] = 4.6999998735e-001,   | lambda[ 74] = -6.83003517299e-002 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 75] = 6.79999999319e-001,   | lambda[ 75] = -8.26357603484e-002  |
| c[ 76] = 6.10000001404e-001,   | lambda[ 76] = -3.04121293316e-002  |
| c[ 77] = 7.09999999150e-001,   | lambda[ 77] = -3.26491895255e-002  |
| c[ 78] = 3.89999999338e-001,   | lambda[ 78] = -2.86785872680e-002  |
| c[ 79] = -3.47000000328e+000,  | lambda[ 79] = -6.92639920267e-003  |
| c[ 80] = -2.89562818168e-011,  | lambda[ 80] = -7.41320507399e-003  |
| c[ 81] = 5.39999999558e-001,   | lambda[ 81] = -2.41172442905e-002  |
| c[ 82] = 1.9999999945e-001,    | lambda[ 82] = -1.32440069457e-002  |
| c[ 83] = 1.0999999975e-001,    | lambda[ 83] = 1.15341710172e-002   |
| c[ 84] = 2.3999999221e-001,    | lambda[ 84] = 2.90121197754e-002   |
| c[ 85] = 2.0999999130e-001,    | lambda[ 85] = 1.92153458182e-002   |
| c[ 86] = -3.9999993796e-002,   | lambda[ 86] = 2.12664347409e-002   |
| c[ 87] = 4.7999999909e-001,    | lambda[ 87] = 4.41137141539e-002   |
| c[ 88] = -6.0700000027e+000,   | lambda[ 88] = 6.51678203307e-002   |
| c[ 89] = 1.6299999985e+000,    | lambda[ 89] = 2.17096873717e-002   |
| c[ 90] = 1.00000000089e-001,   | lambda[ 90] = 2.22184079837e-002   |
| c[ 91] = 6.4999999897e-001,    | lambda[ 91] = 2.94049201530e-002   |
| c[ 92] = 1.20000000002e-001,   | lambda[ 92] = 4.96346333458e-003   |
| c[ 93] = 2.9999999980e-001,    | lambda[ 93] = -1.17963855881e-002  |
| c[ 94] = 4.1999999986e-001,    | lambda[ 94] = -2.13877761199e-002  |
| c[ 95] = 3.7999999800e-001,    | lambda[ 95] = -2.07085726802e-002  |
| c[ 96] = 1.4999999968e-001,    | lambda[ 96] = -1.62412291203e-002  |
| c[ 97] = 3.3999999999e-001,    | lambda[ 97] = -1.75721286343e-002  |
| c[ 98] = 4.1999999948e-001,    | lambda[ 98] = -2.00316844213e-002  |
| c[ 99] = -2.15000000211e+000,  | lambda[ 99] = -1.24254908494e-002  |
| c[ 100] = 2.1999999994e-001,   | lambda[ 100] = -1.11477305933e-003 |
| c[ 101] = 4.9999999951e-002,   | lambda[ 101] = 1.85999213565e-002  |
| c[ 102] = -1.70000002339e-001, | lambda[ 102] = -4.38220730608e-002 |
| c[ 103] = 3.80000001889e-001,  | lambda[ 103] = -6.65228920365e-002 |
| c[ 104] = 3.09999995503e-001,  | lambda[ 104] = -7.61894425185e-002 |
| c[ 105] = 4.2999999800e-001,   | lambda[ 105] = -7.84029756780e-002 |
| c[ 106] = 4.9999999653e-001,   | lambda[ 106] = -1.02844806628e-001 |
| c[ 107] = 1.99999994620e-002,  | lambda[ 107] = -8.56436806786e-002 |
| c[ 108] = 7.9999992433e-002,   | lambda[ 108] = -8.90512607298e-002 |
| c[ 109] = 3.90000006748e-001,  | lambda[ 109] = -9.38841255697e-002 |
| c[ 110] = -3.60000003778e-001, | lambda[ 110] = -7.91858072761e-002 |
| c[ 111] = 6.7999991801e-001,   | lambda[ 111] = -1.27153317515e-001 |
| c[ 112] = 6.00000951578e-002,  | lambda[ 112] = -7.21444521749e-002 |
| c[ 113] = 7.99999969760e-002,  | lambda[ 113] = -8.72749642075e-002 |
| c[ 114] = 2.1999992970e-001,   | lambda[ 114] = -8.73730543227e-002 |
| c[ 115] = 1.84000000002e+000,  | lambda[ 115] = -1.04142170861e-002 |
| c[ 116] = 1.9999998231e-001,   | lambda[ 116] = -9.48371228246e-002 |
| c[ 117] = 3.2999999552e-001,   | lambda[ 117] = -8.00777836518e-002 |
| c[ 118] = 9.00000001296e-002,  | lambda[ 118] = -4.44391732236e-003 |
| c[ 119] = 1.00000000274e-001,  | lambda[ 119] = -6.24949746280e-003 |
| c[ 120] = -7.09189151848e-010, | lambda[ 120] = -1.77263071071e-003 |
| c[ 121] = 2.00000002876e-002,  | lambda[ 121] = -8.78346722553e-005 |
| c[ 122] = 4.79083439586e-010,  | lambda[ 122] = 3.75681472954e-003  |
| c[ 123] = 2.30000001210e-001,  | lambda[ 123] = -2.65340529364e-003 |
| c[ 124] = 1.60000000953e-001,  | lambda[ 124] = -5.98055477914e-003 |
| c[ 125] = 1.00000013933e-002,  | lambda[ 125] = 8.21201741674e-004  |
| c[ 126] = 1.00000001144e-001,  | lambda[ 126] = -1.51239206475e-003 |
| c[ 127] = 3.00000295960e-002,  | lambda[ 127] = 6.63609322795e-004  |
| c[ 128] = 3.00000007050e-002,  | lambda[ 128] = -3.39810155079e-003 |
| c[ 129] = 7.9999988970e-002,   | lambda[ 129] = -5.55284231356e-003 |
| c[ 130] = 4.9999990949e-002,   | lambda[ 130] = -5.13957784885e-003 |
| c[ 131] = 2.9999986906e-002,   | lambda[ 131] = -7.45296380653e-004 |
| c[ 132] = 6.9999998754e-002,   | lambda[ 132] = -1.50272761493e-003 |
| c[ 133] = 4.00000025568e-002,  | lambda[ 133] = -8.73259957745e-004 |
| c[ 134] = -3.56327234385e-009, | lambda[ 134] = -6.89379461533e-005 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 135] = 9.00000039536e-002,  | lambda[ 135] = -1.82309898332e-00  |
| c[ 136] = 9.00000003619e-002,  | lambda[ 136] = -2.54468203127e-004 |
| c[ 137] = 1.99161398484e-009,  | lambda[ 137] = 2.26219803739e-004  |
| c[ 138] = -1.47404755069e-009, | lambda[ 138] = -1.29892823977e-003 |
| c[ 139] = 1.10000000011e-001,  | lambda[ 139] = -2.67203647113e-003 |
| c[ 140] = 1.00000000008e-001,  | lambda[ 140] = -1.66845933384e-003 |
| c[ 141] = 6.99999999901e-002,  | lambda[ 141] = 3.32300495383e-004  |
| c[ 142] = 7.99999999895e-002,  | lambda[ 142] = 1.80076657143e-003  |
| c[ 143] = 2.1999999998e-001,   | lambda[ 143] = -1.14417135519e-003 |
| c[ 144] = -4.88942220045e-013, | lambda[ 144] = 2.09280732516e-004  |
| c[ 145] = 1.10000000002e-001,  | lambda[ 145] = 2.34954968591e-003  |
| c[ 146] = 4.00000001299e-002,  | lambda[ 146] = -1.48504101944e-003 |
| c[ 147] = 8.00000001176e-002,  | lambda[ 147] = -5.45523676164e-003 |
| c[ 148] = 5.00000000440e-002,  | lambda[ 148] = -5.94484398951e-003 |
| c[ 149] = 1.10000000011e-001,  | lambda[ 149] = -4.74622786808e-003 |
| c[ 150] = 3.00000001252e-002,  | lambda[ 150] = -1.09253668981e-003 |
| c[ 151] = 3.00000001636e-002,  | lambda[ 151] = -3.39549441390e-003 |
| c[ 152] = 3.00000000017e-002,  | lambda[ 152] = -9.52251022974e-005 |
| c[ 153] = -3.12638803734e-012, | lambda[ 153] = 1.06703795710e-005  |
| c[ 154] = -2.53663756666e-012, | lambda[ 154] = 7.52463154331e-004  |
| c[ 155] = 6.9999999995e-002,   | lambda[ 155] = 3.27335477633e-003  |
| c[ 156] = -4.88853402203e-012, | lambda[ 156] = 9.31931671113e-007  |
| c[ 157] = 3.66444652400e-010,  | lambda[ 157] = 2.71331967008e-004  |
| c[ 158] = 1.10000000092e-001,  | lambda[ 158] = -6.85685695976e-003 |
| c[ 159] = 2.60000000143e-001,  | lambda[ 159] = -2.31511332029e-003 |
| c[ 160] = 3.20000000018e-001,  | lambda[ 160] = -2.45946366518e-003 |
| c[ 161] = 1.56319401867e-012,  | lambda[ 161] = -4.19618823575e-004 |
| c[ 162] = 2.70000000065e-001,  | lambda[ 162] = -3.42304237928e-003 |
| c[ 163] = 1.00000000007e-001,  | lambda[ 163] = -3.51519566318e-003 |
| c[ 164] = 6.9999999995e-002,   | lambda[ 164] = -6.66148330969e-003 |
| c[ 165] = 9.99999998782e-002,  | lambda[ 165] = -4.16085732450e-003 |
| c[ 166] = 1.00000000003e-001,  | lambda[ 166] = -4.02715929756e-003 |
| c[ 167] = 6.9999999981e-002,   | lambda[ 167] = -7.63050404500e-003 |
| c[ 168] = 1.60000000001e-001,  | lambda[ 168] = -7.23372998519e-003 |
| c[ 169] = 3.0999999998e-001,   | lambda[ 169] = -1.08220095733e-002 |
| c[ 170] = 1.50000000037e-001,  | lambda[ 170] = -6.01650405682e-003 |
| c[ 171] = 9.00000000016e-002,  | lambda[ 171] = -4.12199047178e-003 |
| c[ 172] = 7.9999999997e-002,   | lambda[ 172] = -1.06116938534e-003 |
| c[ 173] = 1.50000000000e-001,  | lambda[ 173] = -5.90711120421e-003 |
| c[ 174] = 2.9999999998e-002,   | lambda[ 174] = -5.33483799430e-003 |
| c[ 175] = 1.59999999984e-001,  | lambda[ 175] = -2.98622572537e-003 |
| c[ 176] = 1.00000000837e-002,  | lambda[ 176] = -4.51732962315e-004 |
| c[ 177] = 3.00000001119e-002,  | lambda[ 177] = -6.64302323762e-004 |
| c[ 178] = 3.00000000323e-002,  | lambda[ 178] = -1.05763590285e-003 |
| c[ 179] = 7.00000001283e-002,  | lambda[ 179] = -1.15412087298e-003 |
| c[ 180] = 7.9999999761e-002,   | lambda[ 180] = -3.99754629368e-003 |
| c[ 181] = 1.50000000001e-001,  | lambda[ 181] = -1.15369091118e-002 |
| c[ 182] = -1.20001317331e-001, | lambda[ 182] = -5.00031215168e-008 |
| c[ 183] = 5.95305909715e-001,  | lambda[ 183] = -2.68992725553e-009 |
| c[ 184] = 1.22978975805e-001,  | lambda[ 184] = -2.11598942542e-008 |
| c[ 185] = -7.23329111537e-001, | lambda[ 185] = -4.39323475358e-009 |
| c[ 186] = -1.05254282074e+000, | lambda[ 186] = -2.39184864913e-008 |
| c[ 187] = 4.58295350307e-001,  | lambda[ 187] = -1.10148981584e-008 |
| c[ 188] = -3.63006476141e-002, | lambda[ 188] = -2.75635508642e-008 |
| c[ 189] = -1.80442789574e-003, | lambda[ 189] = -2.00894837017e-008 |
| c[ 190] = -1.00701671298e-002, | lambda[ 190] = -3.14485326487e-008 |
| c[ 191] = -7.47247727471e-002, | lambda[ 191] = -3.41840982212e-009 |
| c[ 192] = -4.69997638452e-001, | lambda[ 192] = -4.20997274681e-003 |
| c[ 193] = -3.18287787821e-001, | lambda[ 193] = -1.03263219371e-009 |
| c[ 194] = 1.40513093195e-001,  | lambda[ 194] = -3.05770009280e-009 |

|                                |                                    |
|--------------------------------|------------------------------------|
| c[ 195] = 8.75745889650e-004,  | lambda[ 195] = -3.05763852557e-009 |
| c[ 196] = -4.37351050687e-003, | lambda[ 196] = -2.73588559271e-008 |
| c[ 197] = -2.84460772828e-001, | lambda[ 197] = -1.86391361710e-007 |
| c[ 198] = 5.65155606174e-002,  | lambda[ 198] = -3.28769840776e-008 |
| c[ 199] = 1.04157149791e-001,  | lambda[ 199] = -2.99912414739e-009 |
| c[ 200] = 1.06061509085e-001,  | lambda[ 200] = -2.99755920090e-009 |
| c[ 201] = -1.62763664548e-001, | lambda[ 201] = -1.06704275106e-008 |
| c[ 202] = -3.63073256230e-002, | lambda[ 202] = -8.97692130623e-009 |
| c[ 203] = 5.44609143618e-002,  | lambda[ 203] = -2.95972693175e-009 |
| c[ 204] = -1.69382616951e-003, | lambda[ 204] = -3.39193599973e-008 |
| c[ 205] = -3.01339524861e-002, | lambda[ 205] = -4.38335139592e-008 |
| c[ 206] = -5.75406535015e-002, | lambda[ 206] = -5.97758826323e-009 |
| c[ 207] = 3.18973184664e-001,  | lambda[ 207] = -7.58232234290e-009 |
| c[ 208] = -2.07322819204e-001, | lambda[ 208] = -7.53220738026e-008 |
| c[ 209] = -6.69982506410e-001, | lambda[ 209] = -5.71567069154e-004 |
| c[ 210] = -8.49998796846e-001, | lambda[ 210] = -8.31187405909e-003 |
| c[ 211] = -1.10469540896e+000, | lambda[ 211] = -5.27452476627e-009 |
| c[ 212] = 9.12027380427e-002,  | lambda[ 212] = -2.56145664291e-008 |
| c[ 213] = -5.73236497740e-002, | lambda[ 213] = -1.06077341056e-008 |
| c[ 214] = -2.38050190271e-002, | lambda[ 214] = -1.02407940309e-008 |
| c[ 215] = -1.80001762064e-001, | lambda[ 215] = -6.66755752134e-008 |
| c[ 216] = -1.30000823212e-001, | lambda[ 216] = -3.22595829860e-008 |
| c[ 217] = 4.19981229055e-001,  | lambda[ 217] = -1.11232104537e-008 |
| c[ 218] = -4.14415077364e-001, | lambda[ 218] = -6.68281380952e-009 |
| c[ 219] = 7.99979607033e-002,  | lambda[ 219] = -3.22586634270e-008 |
| c[ 220] = 3.51852785910e-002,  | lambda[ 220] = -9.66144055493e-009 |
| c[ 221] = -4.37632796408e-001, | lambda[ 221] = -6.01434415783e-009 |
| c[ 222] = 5.45447311937e-002,  | lambda[ 222] = -2.87812707937e-009 |
| c[ 223] = 3.74247798148e-002,  | lambda[ 223] = -9.64017016415e-009 |
| c[ 224] = -1.00022949201e-002, | lambda[ 224] = -8.33477817515e-008 |
| c[ 225] = -3.62344686560e-002, | lambda[ 225] = -1.03789679113e-008 |
| c[ 226] = 3.55625746764e-001,  | lambda[ 226] = -9.64961306623e-009 |
| c[ 227] = 3.85020147430e-002,  | lambda[ 227] = -2.86742586332e-008 |
| c[ 228] = -2.00170751018e-002, | lambda[ 228] = -3.23324296748e-008 |
| c[ 229] = -1.19476256101e-001, | lambda[ 229] = -4.52560469064e-008 |
| c[ 230] = -7.92241028522e-002, | lambda[ 230] = -4.90008112227e-009 |
| c[ 231] = -7.00274069109e-002, | lambda[ 231] = -3.23750263920e-008 |
| c[ 232] = 6.11605143121e-003,  | lambda[ 232] = -9.93561909882e-009 |
| c[ 233] = -3.60551276200e-004, | lambda[ 233] = -8.84975967263e-009 |
| c[ 234] = -7.79304255953e-002, | lambda[ 234] = -1.08689153727e-008 |
| c[ 235] = -7.95901485197e-002, | lambda[ 235] = -1.00797108571e-009 |
| c[ 236] = -1.20001317331e-001, | lambda[ 236] = 7.58947786756e-003  |
| c[ 237] = 5.95305909715e-001,  | lambda[ 237] = 4.38126934907e-009  |
| c[ 238] = 1.22978975805e-001,  | lambda[ 238] = 6.35297649284e-008  |
| c[ 239] = -7.23329111537e-001, | lambda[ 239] = 2.68543007361e-009  |
| c[ 240] = -1.05254282074e+000, | lambda[ 240] = 3.27669764261e-009  |
| c[ 241] = 4.58295350307e-001,  | lambda[ 241] = 1.55729422120e-008  |
| c[ 242] = -3.63006476141e-002, | lambda[ 242] = 2.94055772452e-007  |
| c[ 243] = -1.80442789574e-003, | lambda[ 243] = 6.16686986281e-008  |
| c[ 244] = -1.00701671298e-002, | lambda[ 244] = 2.22319001770e-004  |
| c[ 245] = -7.47247727471e-002, | lambda[ 245] = 3.25238686388e-009  |
| c[ 246] = -4.69997638452e-001, | lambda[ 246] = 5.34769309016e-009  |
| c[ 247] = -3.18287787821e-001, | lambda[ 247] = 9.69367404650e-010  |
| c[ 248] = 1.40513093195e-001,  | lambda[ 248] = 3.66357836126e-009  |
| c[ 249] = 8.75745889650e-004,  | lambda[ 249] = 3.66366883909e-009  |
| c[ 250] = -4.37351050687e-003, | lambda[ 250] = 5.14162717599e-008  |
| c[ 251] = -2.84460772828e-001, | lambda[ 251] = 3.84173305612e-008  |
| c[ 252] = 5.65155606174e-002,  | lambda[ 252] = 1.53984092749e-006  |
| c[ 253] = 1.04157149791e-001,  | lambda[ 253] = 3.75136643127e-009  |
| c[ 254] = 1.06061509085e-001,  | lambda[ 254] = 3.75381886318e-009  |

|                                |                                   |
|--------------------------------|-----------------------------------|
| c[ 255] = -1.62763664548e-001, | lambda[ 255] = 9.40882560369e-009 |
| c[ 256] = -3.63073256230e-002, | lambda[ 256] = 5.44652662954e-009 |
| c[ 257] = 5.44609143618e-002,  | lambda[ 257] = 3.81497719778e-009 |
| c[ 258] = -1.69382616951e-003, | lambda[ 258] = 4.39382165486e-006 |
| c[ 259] = -3.01339524861e-002, | lambda[ 259] = 1.19333578427e-004 |
| c[ 260] = -5.75406535015e-002, | lambda[ 260] = 1.37532814810e-008 |
| c[ 261] = 3.18973184664e-001,  | lambda[ 261] = 3.72974466791e-009 |
| c[ 262] = -2.07322819204e-001, | lambda[ 262] = 3.74199424309e-008 |
| c[ 263] = -6.69982506410e-001, | lambda[ 263] = 3.74536219920e-009 |
| c[ 264] = -8.49998796846e-001, | lambda[ 264] = 3.74532687708e-009 |
| c[ 265] = -1.10469540896e+000, | lambda[ 265] = 2.43659064029e-009 |
| c[ 266] = 9.12027380427e-002,  | lambda[ 266] = 3.47241418846e-007 |
| c[ 267] = -5.73236497740e-002, | lambda[ 267] = 9.45812156412e-009 |
| c[ 268] = -2.38050190271e-002, | lambda[ 268] = 9.77026577833e-009 |
| c[ 269] = -1.80001762064e-001, | lambda[ 269] = 5.66871544419e-003 |
| c[ 270] = -1.30000823212e-001, | lambda[ 270] = 1.21399402554e-002 |
| c[ 271] = 4.19981229055e-001,  | lambda[ 271] = 5.35529478577e-004 |
| c[ 272] = -4.14415077364e-001, | lambda[ 272] = 3.38567183113e-009 |
| c[ 273] = 7.99979607033e-002,  | lambda[ 273] = 4.90386925291e-003 |
| c[ 274] = 3.51852785910e-002,  | lambda[ 274] = 1.00351586948e-009 |
| c[ 275] = -4.37632796408e-001, | lambda[ 275] = 2.90925255154e-009 |
| c[ 276] = 5.45447311937e-002,  | lambda[ 276] = 3.95958021737e-009 |
| c[ 277] = 3.74247798148e-002,  | lambda[ 277] = 1.03877065793e-008 |
| c[ 278] = -1.00022949201e-002, | lambda[ 278] = 4.35696720317e-003 |
| c[ 279] = -3.62344686560e-002, | lambda[ 279] = 9.64769237061e-009 |
| c[ 280] = 3.55625746764e-001,  | lambda[ 280] = 9.86497446505e-009 |
| c[ 281] = 3.85020147430e-002,  | lambda[ 281] = 4.96883111149e-008 |
| c[ 282] = -2.00170751018e-002, | lambda[ 282] = 5.80418212073e-004 |
| c[ 283] = -1.19476256101e-001, | lambda[ 283] = 1.12322388109e-007 |
| c[ 284] = -7.92241028522e-002, | lambda[ 284] = 5.10407488293e-009 |
| c[ 285] = -7.00274069109e-002, | lambda[ 285] = 3.89904775478e-004 |
| c[ 286] = 6.11605143121e-003,  | lambda[ 286] = 1.00064815348e-009 |
| c[ 287] = -3.60551276200e-004, | lambda[ 287] = 1.01316809995e-009 |
| c[ 288] = -7.79304255953e-002, | lambda[ 288] = 4.80789197219e-009 |
| c[ 289] = -7.95901485197e-002, | lambda[ 289] = 9.92153164295e-010 |

#### Solution Vector

|                              |                                   |
|------------------------------|-----------------------------------|
| x[ 0] = 1.07867885035e+000,  | lambda[ 290] = 0.00000000000e+000 |
| x[ 1] = 1.06899809738e+000,  | lambda[ 291] = 2.63032396824e-007 |
| x[ 2] = 1.07720495677e+000,  | lambda[ 292] = 3.81660168886e-007 |
| x[ 3] = 1.07572496105e+000,  | lambda[ 293] = 3.54577391217e-007 |
| x[ 4] = 1.09999970505e+000,  | lambda[ 294] = 3.46778187540e-002 |
| x[ 5] = 1.09756255776e+000,  | lambda[ 295] = 4.10682297982e-006 |
| x[ 6] = 1.09290646604e+000,  | lambda[ 296] = 1.37082042005e-006 |
| x[ 7] = 1.09109310159e+000,  | lambda[ 297] = 1.07591985007e-006 |
| x[ 8] = 1.08093835827e+000,  | lambda[ 298] = 4.68793974542e-007 |
| x[ 9] = 1.09999994894e+000,  | lambda[ 299] = 1.95883063482e-001 |
| x[ 10] = 1.09315790803e+000, | lambda[ 300] = 1.41447984267e-006 |
| x[ 11] = 1.08606410242e+000, | lambda[ 301] = 6.63352104118e-007 |
| x[ 12] = 1.08967283487e+000, | lambda[ 302] = 9.19244189914e-007 |
| x[ 13] = 1.07486618322e+000, | lambda[ 303] = 3.39977411944e-007 |
| x[ 14] = 1.08844220696e+000, | lambda[ 304] = 8.16453710322e-007 |
| x[ 15] = 1.08742997144e+000, | lambda[ 305] = 7.51273768262e-007 |
| x[ 16] = 1.08635698897e+000, | lambda[ 306] = 6.80267822248e-007 |
| x[ 17] = 1.09996384363e+000, | lambda[ 307] = 4.52473337339e-003 |
| x[ 18] = 1.08945455676e+000, | lambda[ 308] = 9.25020416371e-007 |
| x[ 19] = 1.08603713347e+000, | lambda[ 309] = 6.68508719753e-007 |
| x[ 20] = 1.07486761873e+000, | lambda[ 310] = 3.39326658015e-007 |
| x[ 21] = 1.07021546524e+000, | lambda[ 311] = 2.75910639628e-007 |

|                              |                                   |
|------------------------------|-----------------------------------|
| x[ 22] = 1.07357176035e+000, | lambda[ 312] = 3.19672946971e-007 |
| x[ 23] = 1.08827999412e+000, | lambda[ 313] = 7.98189002766e-007 |
| x[ 24] = 1.08637584693e+000, | lambda[ 314] = 6.78976478281e-007 |
| x[ 25] = 1.09999997155e+000, | lambda[ 315] = 3.51095345020e-001 |
| x[ 26] = 1.06435490647e+000, | lambda[ 316] = 2.20039716077e-007 |
| x[ 27] = 1.08026498896e+000, | lambda[ 317] = 4.49890510185e-007 |
| x[ 28] = 1.07440179120e+000, | lambda[ 318] = 3.32146063691e-007 |
| x[ 29] = 1.07518580410e+000, | lambda[ 319] = 3.44627821543e-007 |
| x[ 30] = 1.06874103767e+000, | lambda[ 320] = 2.60167303710e-007 |
| x[ 31] = 1.07821048728e+000, | lambda[ 321] = 4.01495158093e-007 |
| x[ 32] = 1.07979051721e+000, | lambda[ 322] = 4.37850556114e-007 |
| x[ 33] = 1.08601781681e+000, | lambda[ 323] = 6.60878332159e-007 |
| x[ 34] = 1.09651027427e+000, | lambda[ 324] = 2.82301090305e-006 |
| x[ 35] = 1.09443211194e+000, | lambda[ 325] = 1.74176481443e-006 |
| x[ 36] = 1.09474187428e+000, | lambda[ 326] = 1.84821430344e-006 |
| x[ 37] = 1.09999982203e+000, | lambda[ 327] = 5.70402730441e-002 |
| x[ 38] = 1.05063858090e+000, | lambda[ 328] = 1.36132378516e-007 |
| x[ 39] = 1.07670756124e+000, | lambda[ 329] = 3.72815167641e-007 |
| x[ 40] = 1.07364349570e+000, | lambda[ 330] = 3.21928903589e-007 |
| x[ 41] = 1.06703918073e+000, | lambda[ 331] = 2.43587664694e-007 |
| x[ 42] = 1.07247869715e+000, | lambda[ 332] = 3.05427616887e-007 |
| x[ 43] = 1.08099623310e+000, | lambda[ 333] = 4.71171023627e-007 |
| x[ 44] = 1.07140573401e+000, | lambda[ 334] = 2.91398996871e-007 |
| x[ 45] = 1.06497983859e+000, | lambda[ 335] = 2.24930280423e-007 |
| x[ 46] = 1.07593267107e+000, | lambda[ 336] = 3.58586195587e-007 |
| x[ 47] = 1.08306840102e+000, | lambda[ 337] = 5.35936261707e-007 |
| x[ 48] = 1.08576786836e+000, | lambda[ 338] = 6.48745104853e-007 |
| x[ 49] = 1.08741759500e+000, | lambda[ 339] = 7.41381134642e-007 |
| x[ 50] = 1.07651683517e+000, | lambda[ 340] = 3.69023039645e-007 |
| x[ 51] = 1.05923619799e+000, | lambda[ 341] = 1.82424906542e-007 |
| x[ 52] = 1.05372821201e+000, | lambda[ 342] = 1.50999265204e-007 |
| x[ 53] = 1.05304926182e+000, | lambda[ 343] = 1.47607865814e-007 |
| x[ 54] = 1.06696948466e+000, | lambda[ 344] = 2.42807780156e-007 |
| x[ 55] = 1.06596922769e+000, | lambda[ 345] = 2.33034537813e-007 |
| x[ 56] = 1.06607233560e+000, | lambda[ 346] = 2.34218381735e-007 |
| x[ 57] = 1.06787401168e+000, | lambda[ 347] = 2.51466411234e-007 |
| x[ 58] = 1.06019453656e+000, | lambda[ 348] = 1.88643303811e-007 |
| x[ 59] = 1.08750977146e+000, | lambda[ 349] = 7.46857728377e-007 |
| x[ 60] = 1.08407558194e+000, | lambda[ 350] = 5.73492536098e-007 |
| x[ 61] = 1.08635251197e+000, | lambda[ 351] = 6.78858190509e-007 |
| x[ 62] = 1.08138148037e+000, | lambda[ 352] = 4.81917261484e-007 |
| x[ 63] = 1.05351703907e+000, | lambda[ 353] = 1.49973846299e-007 |
| x[ 64] = 1.06027358818e+000, | lambda[ 354] = 1.89310256940e-007 |
| x[ 65] = 1.05258349177e+000, | lambda[ 355] = 1.45360798386e-007 |
| x[ 66] = 1.09999998279e+000, | lambda[ 356] = 5.81005117219e-001 |
| x[ 67] = 1.08574713143e+000, | lambda[ 357] = 6.47734991748e-007 |
| x[ 68] = 1.05120431378e+000, | lambda[ 358] = 1.38820896714e-007 |
| x[ 69] = 1.09999995539e+000, | lambda[ 359] = 2.24184473580e-001 |
| x[ 70] = 1.07745376260e+000, | lambda[ 360] = 3.86422133818e-007 |
| x[ 71] = 1.07782263134e+000, | lambda[ 361] = 3.94072484431e-007 |
| x[ 72] = 1.07898325738e+000, | lambda[ 362] = 4.19263581881e-007 |
| x[ 73] = 1.07662141494e+000, | lambda[ 363] = 3.70796132194e-007 |
| x[ 74] = 1.05886286064e+000, | lambda[ 364] = 1.79963215181e-007 |
| x[ 75] = 1.06163707674e+000, | lambda[ 365] = 1.98630902624e-007 |
| x[ 76] = 1.04638776683e+000, | lambda[ 366] = 1.18122607715e-007 |
| x[ 77] = 1.08677906489e+000, | lambda[ 367] = 7.01825616567e-007 |
| x[ 78] = 1.08254118164e+000, | lambda[ 368] = 5.17466429595e-007 |
| x[ 79] = 1.08421448353e+000, | lambda[ 369] = 5.78716808417e-007 |
| x[ 80] = 1.09999985171e+000, | lambda[ 370] = 6.74556593839e-002 |
| x[ 81] = 1.04816692760e+000, | lambda[ 371] = 1.25445924174e-007 |

|                                |                                   |
|--------------------------------|-----------------------------------|
| x[ 82] = 1.07931427952e+000,   | lambda[ 372] = 4.27400453122e-007 |
| x[ 83] = 1.08134147534e+000,   | lambda[ 373] = 4.80480328283e-007 |
| x[ 84] = 1.08452263723e+000,   | lambda[ 374] = 5.91430816687e-007 |
| x[ 85] = 1.09249577602e+000,   | lambda[ 375] = 1.27862382728e-006 |
| x[ 86] = 1.08727628125e+000,   | lambda[ 376] = 7.30901180504e-007 |
| x[ 87] = 1.09999853402e+000,   | lambda[ 377] = 6.80126043905e-003 |
| x[ 88] = 1.08888560748e+000,   | lambda[ 378] = 8.46401372334e-007 |
| x[ 89] = 1.09999997999e+000,   | lambda[ 379] = 4.99798311047e-001 |
| x[ 90] = 1.08377422713e+000,   | lambda[ 380] = 5.61643893375e-007 |
| x[ 91] = 1.08823384205e+000,   | lambda[ 381] = 7.96086081506e-007 |
| x[ 92] = 1.09051244714e+000,   | lambda[ 382] = 1.00120377032e-006 |
| x[ 93] = 1.08068468554e+000,   | lambda[ 383] = 4.62290043522e-007 |
| x[ 94] = 1.07915273679e+000,   | lambda[ 384] = 4.23776973053e-007 |
| x[ 95] = 1.06846749759e+000,   | lambda[ 385] = 2.57727003629e-007 |
| x[ 96] = 1.07679476721e+000,   | lambda[ 386] = 3.74267061463e-007 |
| x[ 97] = 1.08419514420e+000,   | lambda[ 387] = 5.78294469785e-007 |
| x[ 98] = 1.09302577686e+000,   | lambda[ 388] = 1.38183462669e-006 |
| x[ 99] = 1.09530483977e+000,   | lambda[ 389] = 2.07288259858e-006 |
| x[ 100] = 1.09999981099e+000,  | lambda[ 390] = 5.29909683059e-002 |
| x[ 101] = 1.08391464387e+000,  | lambda[ 391] = 5.67229682804e-007 |
| x[ 102] = 1.08737366000e+000,  | lambda[ 392] = 7.38461305061e-007 |
| x[ 103] = 1.09146695273e+000,  | lambda[ 393] = 1.11818719063e-006 |
| x[ 104] = 1.08190250743e+000,  | lambda[ 394] = 4.96615057423e-007 |
| x[ 105] = 1.07921352698e+000,  | lambda[ 395] = 4.24822280579e-007 |
| x[ 106] = 1.07217402606e+000,  | lambda[ 396] = 3.01073325343e-007 |
| x[ 107] = 1.06663676604e+000,  | lambda[ 397] = 2.39549255123e-007 |
| x[ 108] = 1.07591122588e+000,  | lambda[ 398] = 3.57742590959e-007 |
| x[ 109] = 1.07473220015e+000,  | lambda[ 399] = 3.37963227162e-007 |
| x[ 110] = 1.07477429934e+000,  | lambda[ 400] = 3.38451155775e-007 |
| x[ 111] = 1.08304643248e+000,  | lambda[ 401] = 5.34387713084e-007 |
| x[ 112] = 1.06019846605e+000,  | lambda[ 402] = 1.88620964184e-007 |
| x[ 113] = 1.09590452961e+000,  | lambda[ 403] = 3.62865493818e-006 |
| x[ 114] = 1.07567840361e+000,  | lambda[ 404] = 3.53055192847e-007 |
| x[ 115] = 1.07533534389e+000,  | lambda[ 405] = 3.47247794572e-007 |
| x[ 116] = 1.05062701219e+000,  | lambda[ 406] = 1.36179712862e-007 |
| x[ 117] = 1.07556823189e+000,  | lambda[ 407] = 3.51745457536e-007 |
| x[ 118] = 1.04868686707e+000,  | lambda[ 408] = 1.27526499244e-007 |
| x[ 119] = -2.82119097483e-001, | lambda[ 409] = 0.00000000000e+000 |
| x[ 120] = -2.72881817439e-001, | lambda[ 410] = 0.00000000000e+000 |
| x[ 121] = -2.68352452696e-001, | lambda[ 411] = 0.00000000000e+000 |
| x[ 122] = -2.14341402271e-001, | lambda[ 412] = 0.00000000000e+000 |
| x[ 123] = -2.06682905855e-001, | lambda[ 413] = 0.00000000000e+000 |
| x[ 124] = -2.47150445913e-001, | lambda[ 414] = 0.00000000000e+000 |
| x[ 125] = -2.53237641900e-001, | lambda[ 415] = 0.00000000000e+000 |
| x[ 126] = -1.31874555066e-001, | lambda[ 416] = 0.00000000000e+000 |
| x[ 127] = -1.85216117834e-002, | lambda[ 417] = 0.00000000000e+000 |
| x[ 128] = 1.02960050866e-001,  | lambda[ 418] = 0.00000000000e+000 |
| x[ 129] = -2.50718281004e-001, | lambda[ 419] = 0.00000000000e+000 |
| x[ 130] = -2.57852184661e-001, | lambda[ 420] = 0.00000000000e+000 |
| x[ 131] = -2.71372344147e-001, | lambda[ 421] = 0.00000000000e+000 |
| x[ 132] = -2.69322383921e-001, | lambda[ 422] = 0.00000000000e+000 |
| x[ 133] = -2.76431595820e-001, | lambda[ 423] = 0.00000000000e+000 |
| x[ 134] = -2.62817900373e-001, | lambda[ 424] = 0.00000000000e+000 |
| x[ 135] = -2.38041350894e-001, | lambda[ 425] = 0.00000000000e+000 |
| x[ 136] = -2.71796981918e-001, | lambda[ 426] = 0.00000000000e+000 |
| x[ 137] = -2.80058533999e-001, | lambda[ 427] = 0.00000000000e+000 |
| x[ 138] = -2.66481965288e-001, | lambda[ 428] = 0.00000000000e+000 |
| x[ 139] = -2.43201433000e-001, | lambda[ 429] = 0.00000000000e+000 |
| x[ 140] = -2.05719802321e-001, | lambda[ 430] = 0.00000000000e+000 |
| x[ 141] = -1.32008655665e-001, | lambda[ 431] = 0.00000000000e+000 |

|                                |                                  |
|--------------------------------|----------------------------------|
| x[ 142] = -1.35567368048e-001, | lambda[ 432] = 0.0000000000e+000 |
| x[ 143] = -2.01304507143e-002, | lambda[ 433] = 0.0000000000e+000 |
| x[ 144] = 7.53373951171e-003,  | lambda[ 434] = 0.0000000000e+000 |
| x[ 145] = -2.17090330067e-001, | lambda[ 435] = 0.0000000000e+000 |
| x[ 146] = -2.41152014616e-001, | lambda[ 436] = 0.0000000000e+000 |
| x[ 147] = -2.54841645392e-001, | lambda[ 437] = 0.0000000000e+000 |
| x[ 148] = -1.64425151116e-001, | lambda[ 438] = 0.0000000000e+000 |
| x[ 149] = -2.53082075332e-001, | lambda[ 439] = 0.0000000000e+000 |
| x[ 150] = -2.25813570816e-001, | lambda[ 440] = 0.0000000000e+000 |
| x[ 151] = -2.85323125501e-001, | lambda[ 441] = 0.0000000000e+000 |
| x[ 152] = -2.76773539606e-001, | lambda[ 442] = 0.0000000000e+000 |
| x[ 153] = -2.83438044903e-001, | lambda[ 443] = 0.0000000000e+000 |
| x[ 154] = -2.83608785465e-001, | lambda[ 444] = 0.0000000000e+000 |
| x[ 155] = -2.69719149766e-001, | lambda[ 445] = 0.0000000000e+000 |
| x[ 156] = -1.94939159791e-001, | lambda[ 446] = 0.0000000000e+000 |
| x[ 157] = -3.18026315522e-001, | lambda[ 447] = 0.0000000000e+000 |
| x[ 158] = -3.33488515585e-001, | lambda[ 448] = 0.0000000000e+000 |
| x[ 159] = -3.39629090279e-001, | lambda[ 449] = 0.0000000000e+000 |
| x[ 160] = -3.15585524612e-001, | lambda[ 450] = 0.0000000000e+000 |
| x[ 161] = -2.77628234603e-001, | lambda[ 451] = 0.0000000000e+000 |
| x[ 162] = -2.41234833434e-001, | lambda[ 452] = 0.0000000000e+000 |
| x[ 163] = -2.13412718668e-001, | lambda[ 453] = 0.0000000000e+000 |
| x[ 164] = -1.70201059958e-001, | lambda[ 454] = 0.0000000000e+000 |
| x[ 165] = -1.35769682245e-001, | lambda[ 455] = 0.0000000000e+000 |
| x[ 166] = -1.46892032103e-001, | lambda[ 456] = 0.0000000000e+000 |
| x[ 167] = -1.30645266044e-001, | lambda[ 457] = 0.0000000000e+000 |
| x[ 168] = -1.63998843292e-001, | lambda[ 458] = 0.0000000000e+000 |
| x[ 169] = -2.05550703467e-001, | lambda[ 459] = 0.0000000000e+000 |
| x[ 170] = -2.20228824238e-001, | lambda[ 460] = 0.0000000000e+000 |
| x[ 171] = -2.35485365617e-001, | lambda[ 461] = 0.0000000000e+000 |
| x[ 172] = -2.23326799393e-001, | lambda[ 462] = 0.0000000000e+000 |
| x[ 173] = -2.27660390276e-001, | lambda[ 463] = 0.0000000000e+000 |
| x[ 174] = -2.24767624927e-001, | lambda[ 464] = 0.0000000000e+000 |
| x[ 175] = -2.05107588517e-001, | lambda[ 465] = 0.0000000000e+000 |
| x[ 176] = -2.18089953207e-001, | lambda[ 466] = 0.0000000000e+000 |
| x[ 177] = -1.60927084360e-001, | lambda[ 467] = 0.0000000000e+000 |
| x[ 178] = -1.03408804166e-001, | lambda[ 468] = 0.0000000000e+000 |
| x[ 179] = -9.03872143235e-002, | lambda[ 469] = 0.0000000000e+000 |
| x[ 180] = -9.80242588406e-002, | lambda[ 470] = 0.0000000000e+000 |
| x[ 181] = -1.10597399146e-001, | lambda[ 471] = 0.0000000000e+000 |
| x[ 182] = -8.30531072355e-002, | lambda[ 472] = 0.0000000000e+000 |
| x[ 183] = -3.10008927837e-002, | lambda[ 473] = 0.0000000000e+000 |
| x[ 184] = -3.00905266553e-002, | lambda[ 474] = 0.0000000000e+000 |
| x[ 185] = -7.38454600923e-002, | lambda[ 475] = 0.0000000000e+000 |
| x[ 186] = -3.41116741649e-002, | lambda[ 476] = 0.0000000000e+000 |
| x[ 187] = -1.14923642646e-001, | lambda[ 477] = 0.0000000000e+000 |
| x[ 188] = -1.20261325173e-001, | lambda[ 478] = 0.0000000000e+000 |
| x[ 189] = -1.37200356251e-001, | lambda[ 479] = 0.0000000000e+000 |
| x[ 190] = -1.22482075400e-001, | lambda[ 480] = 0.0000000000e+000 |
| x[ 191] = -1.30109832076e-001, | lambda[ 481] = 0.0000000000e+000 |
| x[ 192] = -1.10374622198e-001, | lambda[ 482] = 0.0000000000e+000 |
| x[ 193] = -1.27728761018e-001, | lambda[ 483] = 0.0000000000e+000 |
| x[ 194] = -4.99831855427e-002, | lambda[ 484] = 0.0000000000e+000 |
| x[ 195] = -5.38850346563e-002, | lambda[ 485] = 0.0000000000e+000 |
| x[ 196] = -4.83670516867e-002, | lambda[ 486] = 0.0000000000e+000 |
| x[ 197] = -1.05837644230e-002, | lambda[ 487] = 0.0000000000e+000 |
| x[ 198] = -2.49942098961e-002, | lambda[ 488] = 0.0000000000e+000 |
| x[ 199] = -4.17385662941e-002, | lambda[ 489] = 0.0000000000e+000 |
| x[ 200] = -2.49863863488e-002, | lambda[ 490] = 0.0000000000e+000 |
| x[ 201] = 1.05456395013e-002,  | lambda[ 491] = 0.0000000000e+000 |

|                                |                                   |
|--------------------------------|-----------------------------------|
| x[ 202] = 3.23485014990e-002,  | lambda[ 492] = 0.00000000000e+000 |
| x[ 203] = 1.46193963644e-002,  | lambda[ 493] = 0.00000000000e+000 |
| x[ 204] = 2.00873009183e-002,  | lambda[ 494] = 0.00000000000e+000 |
| x[ 205] = 7.90499097855e-002,  | lambda[ 495] = 0.00000000000e+000 |
| x[ 206] = 1.39140216043e-001,  | lambda[ 496] = 0.00000000000e+000 |
| x[ 207] = 4.61889474000e-002,  | lambda[ 497] = 0.00000000000e+000 |
| x[ 208] = 4.40010464395e-002,  | lambda[ 498] = 0.00000000000e+000 |
| x[ 209] = 5.36732560839e-002,  | lambda[ 499] = 0.00000000000e+000 |
| x[ 210] = 1.11115128077e-002,  | lambda[ 500] = 0.00000000000e+000 |
| x[ 211] = -1.93635177635e-002, | lambda[ 501] = 0.00000000000e+000 |
| x[ 212] = -3.34504618773e-002, | lambda[ 502] = 0.00000000000e+000 |
| x[ 213] = -3.56093809577e-002, | lambda[ 503] = 0.00000000000e+000 |
| x[ 214] = -2.86174280158e-002, | lambda[ 504] = 0.00000000000e+000 |
| x[ 215] = -3.54529358135e-002, | lambda[ 505] = 0.00000000000e+000 |
| x[ 216] = -4.29113313943e-002, | lambda[ 506] = 0.00000000000e+000 |
| x[ 217] = -2.71858099293e-002, | lambda[ 507] = 0.00000000000e+000 |
| x[ 218] = -5.50734978962e-003, | lambda[ 508] = 0.00000000000e+000 |
| x[ 219] = 3.25029873017e-002,  | lambda[ 509] = 0.00000000000e+000 |
| x[ 220] = -8.10303759334e-002, | lambda[ 510] = 0.00000000000e+000 |
| x[ 221] = -1.24844414065e-001, | lambda[ 511] = 0.00000000000e+000 |
| x[ 222] = -1.41216898647e-001, | lambda[ 512] = 0.00000000000e+000 |
| x[ 223] = -1.43835729213e-001, | lambda[ 513] = 0.00000000000e+000 |
| x[ 224] = -1.83557320241e-001, | lambda[ 514] = 0.00000000000e+000 |
| x[ 225] = -1.56716119004e-001, | lambda[ 515] = 0.00000000000e+000 |
| x[ 226] = -1.62540994685e-001, | lambda[ 516] = 0.00000000000e+000 |
| x[ 227] = -1.73027777725e-001, | lambda[ 517] = 0.00000000000e+000 |
| x[ 228] = -1.50013127306e-001, | lambda[ 518] = 0.00000000000e+000 |
| x[ 229] = -2.11978184439e-001, | lambda[ 519] = 0.00000000000e+000 |
| x[ 230] = -2.37515648668e-001, | lambda[ 520] = 0.00000000000e+000 |
| x[ 231] = -2.30023583671e-001, | lambda[ 521] = 0.00000000000e+000 |
| x[ 232] = -2.30060178842e-001, | lambda[ 522] = 0.00000000000e+000 |
| x[ 233] = -4.08624988321e-002, | lambda[ 523] = 0.00000000000e+000 |
| x[ 234] = -2.80080030287e-001, | lambda[ 524] = 0.00000000000e+000 |
| x[ 235] = -1.25128162391e-001, | lambda[ 525] = 0.00000000000e+000 |

---