

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SETOR TELEMÁTICA

UM METODO PRIMAL PARA PROBLEMAS DE CONTROLE ÓTIMO DE
SISTEMAS DINÂMICOS DE GRANDE PORTE

PAULO AUGUSTO VALENTE FERREIRA

Orientador: PROF. DR. JOSÉ CLÁUDIO GEROMEL

62/83

Tese apresentada à Faculdade de Engenharia de Campinas, da Universidade Estadual de Campinas - UNICAMP, como parte dos requisitos exigidos para a obtenção do título de MESTRE EM CIÊNCIAS

NOVEMBRO 1983

Ao Zito

À minha família

AGRADECIMENTOS

Gostaríamos de expressar aqui nossos agradecimentos à CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - e à TELEBRÁS S/A - Telecomunicações Brasileiras S/A - que através do indispensável apoio financeiro tornaram possível a realização deste trabalho.

Ao Prof. Dr. José Cláudio Geromel pelo tema, estímulo e constante trabalho de orientação e à Elza Aoki pelo eficiente trabalho de datilografia.

A todos aqueles que nos acompanharam durante o transcorrer do curso e em especial aos amigos Martins, Naka e Oscar.

INDICE

INTRODUÇÃO GERAL	1
SIMBOLOGIA	4
CAPÍTULO I :	
I.1. Introdução	6
I.2. Conceitos Básicos	6
I.3. Condições de Kuhn-Tucker	14
I.4. Dualidade	17
I.5. O Problema de Otimização	19
I.6. Gradiente Projetado	22
I.7. Gradiente Reduzido	28
I.8. Gradiente Projetado Via Decomposição	34
I.9. Conclusão	42
CAPÍTULO II :	
II.1. Introdução	45
II.2. Desenvolvimento Teórico do Método: Caso Estático	45
II.3. Método Dual-Clássico	52
II.4. Condições de Otimalidade	57
II.5. Exemplos	63
II.6. Conclusão	72

CAPÍTULO III :

III.1. Introdução	75
III.2. Desenvolvimento Teórico do Método: Caso Dinâmico ...	76
III.3. A Matriz de Projeção G	91
III.4. Conclusão	92

CAPÍTULO IV :

IV.1. Introdução	95
IV.2. Formulação do Problema da Geração Hidroelétrica de Energia	97
IV.3. Aplicação a um Sistema Real Brasileiro	100
IV.4. Conclusão	106

CONCLUSÃO GERAL	113
-----------------------	-----

REFERÊNCIAS BIBLIOGRÁFICAS	116
----------------------------------	-----

APÊNDICE A : Análise Numérica dos Polinômios Cota-Volume - Usinas de São Simão, Marimbondo, Água Vermelha e Ilha Solteira	121
-----------------------------------------------------------------------------------------------------------------------------------	-----

APÊNDICE B : Análise Pós-Otimização	131
-------------------------------------------	-----

APÊNDICE C : Listagens dos Programas Computacionais	134
-----------------------------------------------------------	-----

RESUMO

Neste trabalho é proposto um novo método primal para a resolução de problemas de otimização que apresentem função objetivo com primeiras derivadas parciais contínuas e restrições lineares. O método baseia-se na técnica de projeção de gradiente proposta por Rosén [16] obtendo-se porém uma direção de pesquisa em que a contribuição das restrições de igualdade, permanentemente ativas, é desacoplada das restrições de desigualdade, cuja contribuição é variável a cada nova iteração do algoritmo.

Inicialmente desenvolvido para problemas de otimização estática, o método proposto é em seguida estendido para problemas de otimização dinâmica e aplicado numericamente na resolução de um problema de geração hidroelétrica de energia. O trabalho inclui resultados computacionais que ilustram o desempenho do método.

INTRODUÇÃO GERAL

O advento de novos computadores digitais tem permitido nos últimos anos a resolução de uma quantidade crescente de problemas práticos de programação matemática de grande porte que até então, exigiam elevado tempo de processamento e/ou disponibilidade de memória.

Embora a quantidade de problemas hoje passíveis de eficiente resolução tenha crescido, persistem ainda limitações físicas de tempo e memória que proporcionam um desenvolvimento contínuo de novos métodos de programação matemática em geral e de métodos primais, em particular.

A grande maioria dos métodos desta classe baseia-se na técnica originalmente proposta por J.B. Rosen (1960) e que consiste em gerar direções de pesquisa, a partir da projeção do vetor gradiente da função objetivo sobre determinado subconjunto das restrições do problema. A forma na qual esta projeção é obtida distingue os diferentes métodos e a escolha de um deles tendo em vista, a resolução de um problema específico é ditada por um compromisso entre a natureza das hipóteses que o método assume válidas e o esforço computacional necessário para que a projeção seja obtida.

Relaxar as hipóteses de aplicabilidade e reduzir o esforço computacional necessário à projeção é uma tarefa a que se dedicam muitos pesquisadores e que motiva, por conseguinte, o surgimento de novos métodos.

Este trabalho propõe uma nova maneira de obter a projeção do vetor gradiente que acreditamos levar a um razoável compromisso, entre os dois fatores, anteriormente citados.

O método primal que será proposto destina-se à resolução de problemas de programação matemática que apresentem função objetivo com primeiras derivadas parciais contínuas e restrições lineares.

O conteúdo do trabalho está assim dividido:

CAPÍTULO 1 : Introduz os principais conceitos e resultados teóricos utilizados no decorrer do trabalho. São analisados ainda alguns métodos de projeção de gradiente encontrados na literatura. Esta análise deverá fornecer subsídios para futuras comparações.

CAPÍTULO 2 : Trata da proposição e análise do método a problemas de otimização estática. Suas relações com os métodos analisados são apontadas e no fim do capítulo, alguns exemplos ilustram seu desempenho.

CAPÍTULO 3 : Estende a metodologia proposta a problemas de controle ótimo de sistemas dinâmicos discretos com horizonte de tempo finito. Novas e importantes propriedades do método são demonstradas.

CAPÍTULO 4 : O método proposto é então aplicado numericamente na resolução de um problema de otimização dinâmica real: o problema de controle ótimo da geração hidroelétrica de energia.

APÊNDICE A : Produz uma análise numérica dos polinômios que relacionam a cota ou altura de queda d'água com o volume de água armazenado nos reservatórios das usinas hidráulicas estudadas no Capítulo IV.

APÊNDICE B : Apresenta uma figura de mérito para análise de resultados computacionais obtidos neste trabalho.

APÊNDICE C : Consiste de listagens dos programas computacionais desenvolvidos.

SIMBOLOGIA

$x \in \mathbb{R}^n$	-	indica vetor no \mathbb{R}^n
$A \in \mathbb{R}^{n \times n}$	-	indica matriz de dimensão $n \times n$
a, b	-	indicam escalares
x^t	-	indica transposição de x
$\ x\ $	-	norma euclidiana de x
$x \geq 0$	-	indica que todos os elementos de x são não-negativos
$A \geq 0$	-	indica que A é semidefinida positiva
$\ A\ $	-	norma espectral de A
$\lambda_M(A)$	-	autovalor máximo de A
$\min\{a, b\}$	-	o mínimo entre a e b
I_n	-	matriz identidade de ordem n

CAPÍTULO I

I.1. Introdução

Os conceitos e resultados contidos neste capítulo constituem uma amostragem pouco significativa de um campo de conhecimento tão vasto quanto o da Programação Matemática.

Com a inclusão deste capítulo no corpo do trabalho, objetiva-se unicamente oferecer uma rápida referência para muitos dos importantes resultados que se farão necessários nos capítulos seguintes e ao mesmo tempo introduzir boa parte da notação utilizada. De qualquer modo, a leitura deste capítulo não é requisito essencial para a compreensão dos demais e o que será aqui exposto pode ser obtido de forma mais completa através da extensa literatura dedicada ao assunto. A exceção a esta regra é talvez o recente método de Geromel-Baptistella (1981), abordado no fim do capítulo.

Este capítulo está dividido em duas partes distintas. Na primeira parte são apresentadas as definições, conceitos, proposições e teoremas diretamente relacionados com o trabalho. Vários destes resultados serão utilizados em seguida na parte destinada à análise qualitativa de alguns métodos de projeção de gradiente.

I.2. Conceitos Básicos

As primeiras seções deste capítulo têm por objetivo, apresentar alguns conceitos e condições de otimalidade que caracterizam a solução de problemas de otimização com restrições. Considere inicialmente o seguinte problema de programação matemática, comumente conhecido como problema primal:

$$\begin{aligned} \text{Min } f(x) \\ x \in X \end{aligned} \tag{I.1}$$

onde $x \in \mathbb{R}^n$, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ e $X \subseteq \mathbb{R}^n$, um subconjunto compacto do \mathbb{R}^n .

Definição I-1 : Um ponto $x^* \in X$ é dito um mínimo local de $f(x)$ sobre X se existe um escalar positivo ϵ tal que

$$f(x^*) \leq f(x), \quad \forall x \in X \mid \|x^* - x\| < \epsilon \tag{I.2}$$

Diz-se ainda que $x^* \in X$ é um mínimo local estrito de $f(x)$ sobre X se

$$f(x^*) < f(x), \quad \forall x \in X, x \neq x^* \mid \|x^* - x\| < \epsilon \tag{I.3}$$

Definição I-2 : Um ponto $x^* \in X$ é dito um mínimo global de $f(x)$ sobre X se

$$f(x^*) \leq f(x), \quad \forall x \in X \tag{I.4}$$

ou seja, se é possível arbitrar completamente o valor do escalar ϵ . Do mesmo modo, diz-se que $x^* \in X$ é um mínimo global estrito de $f(x)$ sobre X se

$$f(x^*) < f(x), \quad \forall x \in X, x \neq x^* \tag{I.5}$$

Embora o problema primal (I.1) esteja formulado no sentido de que sua solução deva, necessariamente, ser um mínimo global de $f(x)$ sobre X , limitações tanto de nível teórico quanto de implementação de algoritmos restringem a análise do comportamento da função nas vizinhanças de um ponto, resultando desta maneira, condições para existência de mínimos locais. Através

desta análise local, mínimos globais poderão ser seguramente obtidos apenas nos casos em que ao problema inicial forem adicionadas hipóteses de convexidade que essencialmente permitem a generalização dos resultados anteriormente desenvolvidos (Luenberger [14], Hadley [12], Avriel [3]).

Definição I-3 : Um conjunto de pontos \bar{X} é dito convexo se, dados quaisquer dois de seus pontos, o segmento de reta que os une estiver inteiramente contido no conjunto. Matematicamente, dados $\forall x_1, x_2 \in X$ então X é convexo se e somente se

$$x = \alpha x_1 + (1 - \alpha)x_2 \in X, \quad \forall 0 \leq \alpha \leq 1 \quad (I.6)$$

Definição I-4 : Uma função $f(x)$ é dita convexa ($-f(x)$ é dita côncava) sobre um domínio convexo X se dados $\forall x_1, x_2 \in X$,

$$f[\alpha x_1 + (1 - \alpha)x_2] \leq \alpha f(x_1) + (1 - \alpha)f(x_2), \quad \forall 0 \leq \alpha \leq 1 \quad (I.7)$$

ou em palavras, se o segmento de reta que une dois pontos quaisquer sobre sua curva, não for nunca inferior a mesma. Diz-se ainda que $f(x)$ é estritamente convexa sobre X convexo se dados $\forall x_1, x_2 \in X, x_1 \neq x_2$ a desigualdade em (I.7) sempre se verificar para $0 < \alpha < 1$.

Teorema I-1 : Seja $f(x)$ uma função convexa definida sobre um domínio convexo X . Então qualquer mínimo local é também um mínimo global de $f(x)$ sobre X .

O conceito de convexidade em programação matemática é de fundamental importância. Além do natural interesse no estudo de problemas conve-

xos, entre os quais encontram-se problemas de programação linear representativos de uma ampla classe de problemas de grande porte, resultados obtidos com hipóteses de convexidade fornecem a base teórica para o desenvolvimento e análise de métodos aplicáveis a problemas para os quais estas hipóteses não se verificam. Estes resultados, de forte interpretação geométrica, podem algumas vezes também ser estendidos, embora de forma parcial, mesmo a problemas não-convexos (Lasdon [13]).

Tendo em vista uma melhor caracterização de problemas de programação convexa, são apresentados os resultados seguintes, cujas demonstrações podem ser encontradas em Avriel [3], Hadley [12], Lasdon [13], Luenberger [14] e Bazarra [4], entre outros.

Teorema I-2 : A interseção de um número qualquer de conjuntos convexos define um conjunto convexo.

Teorema I-3 : Uma forma quadrática semi-definida positiva é convexa; uma forma quadrática definida positiva é estritamente convexa.

Teorema I-4 : Uma combinação linear positiva de funções convexas define uma função convexa.

Teorema I-5 [Funções Convexas Diferenciáveis] : Seja $f(x) : R^n \rightarrow R \in C^2$ uma função convexa definida sobre um domínio convexo X . Então, as seguintes proposições são equivalentes:

- i) $f(x_1) \geq f(x_2) + \nabla f(x_2)(x_1 - x_2)$, $\forall x_1, x_2 \in X$
 ii) $F(x) \geq 0$, $\forall x \in X$

onde $\nabla f(x) \in \mathbb{R}^n$ representa o vetor gradiente e $F(x) \in \mathbb{R}^{n \times n}$ a matriz Hessiana de $f(x)$ calculados em x .

Através dos métodos primais analisados neste capítulo, a solução para o problema de otimização é obtida iterativamente selecionando-se uma direção de pesquisa que, a partir de um ponto factível $x \in X$, apresenta as seguintes características:

- i) mantêm, para pequenas variações a partir de x , a factibilidade da nova solução;
 ii) decresce o valor da função objetivo.

O conceito-chave para estes métodos é, portanto, o de direção usável.

Definição I-5 : Uma direção $s \in \mathbb{R}^n$ é dita uma direção factível a partir de um dado ponto $x \in X$ se existir um escalar $\bar{\alpha} > 0$ tal que

$$x - \alpha s \in X , \quad \forall 0 \leq \alpha \leq \bar{\alpha} \quad (I.8)$$

Se a condição anterior for satisfeita e simultaneamente ocorrer, para α adequadamente escolhido no intervalo $0 \leq \alpha \leq \bar{\alpha}$,

$$f(x - \alpha s) < f(x) \quad (I.9)$$

diz-se então que s é uma direção usável de (I.1).

O decréscimo de uma função de n variáveis pode ser quantificado através do conceito de derivada direcional, uma generalização para o \mathbb{R}^n das noções de derivada de uma função. Segue então que para uma direção usável de (I.1),

$$Df(x, s) \triangleq \left. \frac{d}{d\alpha} f(x - \alpha s) \right|_{\alpha=0} = \nabla f(x)'s > 0, \quad (I.10)$$

onde $Df(x, s)$ é a derivada direcional de $f(x)$ em x na direção s .

O escalar $\bar{\alpha}$ consiste do máximo valor para α em que a factibilidade da nova solução é mantida. O valor do escalar α , o passo do algoritmo, é tomado como a solução ótima do problema unidimensional

$$\text{Min}_{\alpha} f(x - \alpha s) \quad (I.11)$$

$$0 \leq \alpha \leq \bar{\alpha}$$

Proposição I-1 [Condições Necessárias de 1ª Ordem] : Sejam $f(x) : \mathbb{R}^n \rightarrow \mathbb{R} \in C^1$ e $X \subseteq \mathbb{R}^n$. Se $x^* \in X$ é um mínimo local de $f(x)$ sobre X então, para toda direção factível s de (I.1) tomada a partir de x^* ,

$$\nabla f'(x^*)s \leq 0 \quad (I.12)$$

Interpretando-se este resultado, a equação (I.12) assegura que todas as direções a partir de $x^* \in X$ nas quais $f(x)$ decresce são infactíveis. Esta condição se reduz para problemas irrestritos e problemas em que x^* é interior a X ao conhecido resultado

$$\nabla f(x^*) = 0 \quad (I.13)$$

Proposição I-2 [Condições Necessárias de 2ª Ordem] : Sejam $f(x): \mathbb{R}^n \rightarrow \mathbb{R} \in C^2$ e $X \subseteq \mathbb{R}^n$. Se x^* é um mínimo local de $f(x)$ sobre X então, toda direção factível s que satisfaz

$$\nabla f(x^*)'s = 0, \quad (I.14)$$

satisfaz também,

$$s'F(x^*)s \geq 0. \quad (I.15)$$

A proposição anterior, que deriva diretamente da expansão em série de Taylor de $f(x)$ em torno de x^* impõe que toda direção factível que anule sua variação em primeira ordem, a matriz Hessiana de $f(x)$ seja semi-definida positiva. Segue também diretamente da proposição que, para problemas que têm $\nabla f(x) = 0$ como condição de 1ª ordem, (I.15) estabelece também uma condição suficiente para um mínimo local se

$$F(x^*) > 0. \quad (I.16)$$

Tendo sido introduzidas estas noções iniciais, considere agora o problema primal (I.1) reformulado no sentido de incluir também restrições funcionais:

$$\begin{aligned} \text{Min } f(x) \\ g_i(x) \leq 0, \quad i=1, \dots, m \\ x \in X \end{aligned} \quad (I.17)$$

onde $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, $i=1, \dots, m$.

O desenvolvimento de condições necessárias de 1ª ordem para o problema (I.17) deve agora levar em conta suas restrições funcionais. Nota-se claramente que as restrições ativas de (I.17) exercerão importante influência no desenvolvimento destas condições, visto que, para um dado ponto factível, são as restrições ativas do problema que, localmente, restringem seu domínio de factibilidade. As restrições inativas não tem influência local, pois pequenas perturbações em torno de um ponto factível não violam nenhuma destas restrições. A importância destas restrições surge apenas quando do cálculo do passo de bloqueio $\bar{\alpha}$.

Teorema I-6 : Sejam $g_i(x) : \mathbb{R}^n \rightarrow \mathbb{R} \in C^1$, $i=1, \dots, m$ e $X \subseteq \mathbb{R}^n$. Seja ainda x^* uma solução factível de (I.17) e $I(x^*) = \{i \mid g_i(x^*) = 0\}$, $i=1, \dots, m$. Se x^* é um mínimo local de (I.17), então toda direção factível s que satisfaz (Proposição I-1),

$$\nabla f(x^*)'s \leq 0, \quad (I.18)$$

deve também satisfazer

$$\nabla g_i(x^*)'s \geq 0, \quad i \in I(x^*) \quad (I.19)$$

Prova : Para pequenas perturbações em torno de x^* , a expansão em série de Taylor para as restrições ativas será dada por

$$g_i(x^* - \alpha s) = g_i(x^*) - \alpha \nabla g_i(x^*)'s + O(\alpha^2) \leq 0, \quad i \in I(x^*)$$

para todas as direções factíveis s . Como $g_i(x^*) = 0, \forall i \in I(x^*)$ e, para $\alpha \rightarrow 0^+$ os termos de ordem superior a 1 podem ser desprezados, conclui-se que

$$\nabla g_i(x^*)'s \geq 0, \quad i \in I(x^*)$$

Entretanto, mesmo direções infactíveis ao problema (I.17) podem verificar a expressão anterior. É comum então que se estabeleça algum tipo de qualificação de restrições que elimine esta possibilidade. Entre as mais utilizadas, está a proposta por Fiacco-McCormick (1968) que consiste em pedir que os gradientes de todas as restrições ativas sejam linearmente independentes em x^* (Avriel [3], Luenberger [14]).

I.3. Condições de Kuhn-Tucker

Grande parte da teoria envolvendo condições de otimalidade para problemas de otimização com restrições pode ser unificada a partir de 1950 através dos trabalhos de H.W. Kuhn e A.W. Tucker. O principal resultado destes trabalhos, um conjunto de 4 condições de otimalidade de 1ª ordem, aplica-se a uma grande variedade de problemas de programação não-linear e, dependendo do problema tratado fornecem, sob alguma qualificação de restrições, condições necessárias ou necessárias e suficientes para a existência de soluções locais ou globais. A abordagem simplificada e direta aqui apresentada segue de perto a proposta por Geromei [8] (*).

Considere a função Lagrangeano $L(x, \lambda) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ associada ao problema (I.17):

$$L(x, \lambda) = f(x) + \lambda'g(x) \quad (\text{I.20})$$

onde $\lambda \in \mathbb{R}^m$ representa o vetor de multiplicadores de Lagrange. Por simplicidade de notação, convencionou-se $g(x)' = [g_1(x) \ g_2(x) \ \dots \ g_m(x)]$.

(*) Um desenvolvimento algébrico mais rigoroso pode ser encontrado em Avriel [3] e Hadley [12], por exemplo.

Definição I-6 : Um ponto (x^*, λ^*) com $x^* \in X$ e $\lambda^* \geq 0$ é dito um Ponto de Sela de $L(x, \lambda)$ se satisfaz

$$\begin{aligned} \text{i) } & L(x^*, \lambda^*) \leq L(x, \lambda^*) , & \forall x \in X \\ \text{ii) } & L(x^*, \lambda^*) \geq L(x^*, \lambda) , & \forall \lambda \geq 0 \end{aligned}$$

O teorema a seguir fornece condições necessárias e suficientes para a existência de um Ponto de Sela de $L(x, \lambda)$.

Teorema I-7 : Sejam $\lambda^* \geq 0$ e $x^* \in X$. Então o ponto (x^*, λ^*) é um Ponto de Sela de $L(x, \lambda)$ se e somente se,

- i) x^* minimiza $L(x, \lambda^*)$ sobre X ;
- ii) $g(x^*) \leq 0$;
- iii) $\lambda^{*'} g(x^*) = 0$.

Teorema I-8 [Suficiência de Pontos de Sela] : Se (x^*, λ^*) é um Ponto de Sela de $L(x, \lambda)$, então x^* resolve o problema primal (I.17).

Prova [Lasdon [13]] : Diretamente da definição de Ponto de Sela,

$$f(x^*) + \lambda^{*'} g(x^*) \leq f(x) + \lambda^{*'} g(x) , \quad \forall x \in X \quad (\text{I.21})$$

Então, usando a condição iii) do Teorema I-7 e lembrando que para todo x factível, $\lambda^{*'} g(x) \leq 0$ resulta

$$f(x^*) \leq f(x) , \quad \forall x \in X , \quad (\text{I.22})$$

que pela Definição I-2 é um ponto de mínimo global de (I.17).

Observe-se que o teorema anterior estabelece uma condição suficiente de otimalidade pois, em geral, não é possível garantir a existência de $\lambda^* \geq 0$ tal que as condições do Teorema I-7 se verifiquem. Em outras palavras não é possível, para problemas não-convexos, garantir a existência de um Ponto de Sela da função Lagrangeano, $L(x, \lambda)$.

Por simplicidade, suponha $X \equiv \mathbb{R}^n$. Então, substituindo-se a minimização do Lagrangeano $L(x, \lambda)$, condição i) do Teorema I-7, pela sua estacionariedade, no caso de $L(x, \lambda)$ ser diferenciável, obtêm-se,

$$\nabla_x L(x^*, \lambda^*) \triangleq \left. \frac{\partial}{\partial x} L(x, \lambda^*) \right|_{x=x^*} = 0 \quad (\text{I.23})$$

As condições do Teorema I-7 reduzem-se finalmente às condições de otimalidade de Kuhn-Tucker para o problema (I.17):

$$\begin{aligned} \text{i)} \quad & \nabla_x L(x^*, \lambda^*) = 0 \\ \text{ii)} \quad & g(x^*) \leq 0 \\ \text{iii)} \quad & \lambda^{*i} g(x^*) = 0 \\ \text{iv)} \quad & \lambda^* \geq 0 \end{aligned} \quad (\text{I.24})$$

Note que se o problema primal (I.17) for convexo, isto é, se tanto $f(x)$ como $g_i(x)$, $i = 1, \dots, m$ forem funções convexas sobre um domínio convexo X , a função Lagrangeano $L(x, \lambda)$ consiste, para $\lambda^* \geq 0$, de uma combinação linear positiva de funções convexas e neste caso as condições de Kuhn-

Tucker são necessárias e suficientes, uma consequência imediata dos Teoremas I-1, I-4 e I-5 combinados com a Proposição I-2.

I.4. Dualidade

Ao problema primal (I.17) associa-se um outro, chamado problema dual relativo as restrições funcionais, definido por

$$\begin{array}{ll} \text{Max} & \psi(\lambda) \\ \lambda \geq 0 & \end{array} \quad (\text{I.25})$$

onde $\psi(\lambda)$, a função dual, é escrita como

$$\psi(\lambda) = \min_{x \in X} L(x, \lambda) \quad (\text{I.26})$$

e D, seu domínio de definição, é naturalmente dado por

$$D = \left\{ \lambda \in \mathbb{R}_+^m \mid \exists \min_{x \in X} L(x, \lambda) \right\} \quad (\text{I.27})$$

A formulação de um problema de programação não-linear em termos de seu correspondente problema dual pode trazer vantagens e/ou desvantagens. Assegurada a convexidade das funções envolvidas, os valores ótimos das funções objetivo primal e dual são iguais e neste caso é possível resolver indiretamente o problema primal através da solução do problema dual que em muitos casos é mais facilmente obtida. Os seguintes resultados evidenciam algumas características do problema dual (estes resultados e muitos outros são desenvolvidos de forma completa no Capítulo VIII de Lasdon [13]).

Teorema I-9 : A função dual $\psi(\lambda)$ é côncava sobre qualquer subconjunto convexo do seu domínio D .

Este resultado é importante pois é estabelecido sem nenhuma hipótese de convexidade sobre $f(x)$, $g(x)$ e X . Pelo teorema anterior, sendo D convexo, o problema dual consiste em maximizar uma função côncava sobre um domínio convexo e, portanto, todo máximo local é também máximo global.

Teorema I-10 [Teorema Fraco da Dualidade] : Seja $x \in X$ uma solução factível de (I.17) e $\lambda \in D$ uma solução factível de (I.25). Então,

$$\psi(\lambda) \leq f(x). \quad (I.28)$$

Decorre imediatamente deste resultado que se existe $x^* \in X$ factível primal e $\lambda^* \in D$ factível dual tal que

$$\psi(\lambda^*) = f(x^*), \quad (I.29)$$

então, x^* resolve o problema primal (I.17) e λ^* resolve o problema dual (I.25).

A igualdade de (I.29) é sempre verdadeira para os problemas de programação matemática cuja função Lagrangeano, apresenta um Ponto de Sela. Em particular, problemas convexos satisfazem este requisito.

É ainda possível mostrar (Lasdon [13]) que se $f(x)$ é uma função estritamente convexa e as restrições $g_i(x)$, $i=1, \dots, m$ são convexas sobre X convexo, então a função dual $\psi(\lambda)$ é continuamente diferenciável, viabili-

zando o emprego de métodos de máxima subida, do tipo gradiente, para a maximização da função dual. Este importante resultado será utilizado com frequência no decorrer deste trabalho.

1.5. O Problema de Otimização

Nos problemas tratados neste trabalho, não-linearidades estarão presentes apenas na função objetivo. As restrições funcionais $g_i(x) : R^n \rightarrow R \in C^1$ são assumidas lineares e, portanto, convexas sobre um subconjunto convexo e bem definido do R^n , representado por X . Excetuando-se o método de Geromel-Baptistella (1981), a totalidade dos métodos aqui analisados impõe como hipótese básica de aplicabilidade que as restrições $g_i(x)$, $i = 1, \dots, m$ além de compatíveis, sejam linearmente independentes. As restrições para este problema definem desta forma um politopo convexo de dimensão $n-m$, um sub-espço de natureza linear. Este fato é particularmente importante pois, como será visto em seguida, os métodos de projeção têm suas direções de pesquisa determinadas a partir da projeção ortogonal do gradiente da função objetivo sobre a interseção de um subconjunto das restrições do problema.

O problema de programação não-linear resultante seria então:

$$\text{Min } f(x)$$

$$Ax = b \tag{I.30}$$

$$x \in X$$

onde $x \in R^n$, $A \in R^{m \times n}$ ($m \leq n$), $b \in R^m$, $f(x) : R^n \rightarrow R \in C^1$ e

$$X = \{x \mid \underline{x} \leq x \leq \bar{x}\} \quad (I.31)$$

com \underline{x} e \bar{x} representando os limitantes superior e inferior de x , respectivamente. Problemas com restrições de desigualdade podem ser colocados nesta forma adicionando-se as variáveis de folga necessárias (Sakarovitch [18]). Um subconjunto X mais específico pode também ser obtido a partir de (I.31) ajustando-se convenientemente os limitantes \underline{x} e \bar{x} .

A primeira etapa para aplicação efetiva de métodos primais, consiste em encontrar uma solução factível inicial para o problema. Longe de constituir-se tarefa simples, a obtenção deste primeiro ponto demanda um tempo de computação em muitos casos comparável ao necessário à obtenção da própria solução ótima do problema. Nesta etapa, que para problemas de grande porte tem na FASE I do método SIMPLEX (Sakarovitch [18]) seu algoritmo mais eficiente, são também detetadas possíveis inconsistências e/ou redundâncias no conjunto de restrições do problema.

Outra maneira indicada para a eliminação de redundâncias faz uso indiretamente da Decomposição LU para inversão de matrizes simétricas de finidas positivas. Note que sendo A tal que $\text{rank}(A) < m$ então

$$AA' \geq 0 \quad (I.32)$$

e fazendo-se $V = AA'$, a Decomposição LU permite escrever

$$V = LL' \quad (I.33)$$

onde $L \in R^{m \times m}$ é uma matriz triangular inferior. Então, a partir de (I.33),

$$\det(AA') = \det(L)^2 = \left[\prod_{i=1}^m l_{ii} \right]^2 \quad (I.34)$$

sendo que os elementos ℓ_{ij} são obtidos resolvendo-se a equação matricial não-linear (I.33) recursivamente através de:

$$\ell_{ii} = \left(v_{ij} - \sum_{k=1}^{i-1} \ell_{ik}^2 \right)^{1/2} \quad i = 1, \dots, m$$

$$\ell_{ij} = \frac{v_{ij} - \sum_{k=1}^{j-1} \ell_{ik} \ell_{jk}}{\ell_{jj}} \quad \begin{array}{l} j = 1, \dots, m \\ i = (j+1), \dots, m \end{array}$$
(I.35)

Existindo para algum $i = 1, \dots, m$, um autovalor ℓ_{ii} de L tal que $\ell_{ii} = 0$, a redundância no conjunto de restrições poderá ser eliminada removendo-se a i -ésima linha de A .

Finalizando esta primeira parte o seguinte algoritmo identificará qualquer método de solução para (I.30) que utilize direções factíveis. Dada uma solução inicial factível x^0 de (I.30), faça $k = 0$. Uma iteração genérica k será como segue.

PASSO 1 : Encontre uma direção de pesquisa s^k tal que:

$$a) \exists \bar{\alpha} > 0 \mid x^k - \alpha_k s^k \in \gamma, \quad \forall 0 \leq \alpha_k \leq \bar{\alpha}$$

$$\gamma = \{x \mid Ax = b, x \in X\}$$

$$b) \nabla f(x^k)' s^k > 0$$

PASSO 2 : Determine o passo do algoritmo α_k resolvendo o problema unidimensional (I.11).

PASSO 3 : Atualize as variáveis primais

$$x^{k+1} = x^k - \alpha_k s^k$$

faça $k = k + 1$ e repita o procedimento até que um determinado critério de otimalidade seja satisfeito.

I.6. Gradiente Projetado

Os métodos de projeção podem ser encarados como uma generalização dos métodos de otimização sem restrições que utilizam informação do gradiente da função objetivo para gerar direções de pesquisa. Os métodos de projeção de gradiente também utilizam esta informação, levando porém em conta as restrições do problema. Isto é realizado tomando-se como direção de pesquisa não uma direção como a do gradiente, que em geral produz pontos infactíveis, mas sua projeção ortogonal sobre a interseção das restrições ativas do problema, que como foi anteriormente mencionado, restringem localmente seu domínio da factibilidade. O método do Gradiente Projetado proposto por Rosen (1960) - Rosen ([16]) foi o precursor desta classe de métodos que se diferenciam, basicamente, quanto a forma através da qual a projeção é determinada.

A direção assim obtida não pode ser considerada, em geral, como a localmente melhor, no sentido de fornecer a maior derivada direcional da função no ponto sendo, entretanto, mais simples de ser calculada.

Embora o método seja aplicável ao problema (I.30) diretamente, a apresentação será simplificada considerando-se $\tilde{A}' = [A' \ ; \ I \ ; \ -I]$, $\tilde{b}' = [b' \ ; \ \bar{x}' \ ; \ -\underline{x}']$ e o problema colocado na forma

$$\text{Min } f(x)$$

(I.36)

$$\bar{A}x \leq \bar{b},$$

onde $\bar{A} \in R^{(m+2n) \times n}$, $\bar{b} \in R^{(m+2n)}$, $x \in R^n$ e $f(x) : R^n \rightarrow R \in C^1$

Dado um ponto inicial factível x , determina-se então, quais as restrições ativas de (I.36), representadas através do conjunto de índices $I = \{i \mid \bar{a}_i x = \bar{b}_i\}$, com \bar{a}_i denotando a i -ésima linha de \bar{A} . Este conjunto será composto, evidentemente, pelas restrições de igualdade de (I.30) e suas restrições de desigualdade ativas em x . Seja \bar{A}_1 a matriz formada por estas restrições e \bar{A}_2 a matriz correspondente as restrições inativas. Então \bar{A} admite a seguinte partição:

$$\bar{A}_1 x = \bar{b}_1$$

(I.37)

$$\bar{A}_2 x < \bar{b}_2$$

O objetivo é então encontrar uma direção s que ao mesmo tempo em que decresça, o valor da função ($\nabla f(x)'s > 0$) satisfaça

$$\bar{a}_i s = 0, \quad \forall i \in I \quad (I.38)$$

ou equivalentemente, que pertença ao espaço nulo de \bar{A}_1 , $n(\bar{A}_1)$. A projeção do vetor gradiente se fará, portanto, sobre este sub-espaço, interseção das restrições ativas de (I.36). Lembrando agora que $\mathcal{R}(\bar{A}_1')$, o espaço range de \bar{A}_1' é ortogonal a $n(\bar{A}_1)$, é possível então escrever qualquer vetor, e em particular o vetor gradiente, como uma composição de sua projeção sobre estes sub-espaços ortogonais. No R^3 , por exemplo, ter-se-ia

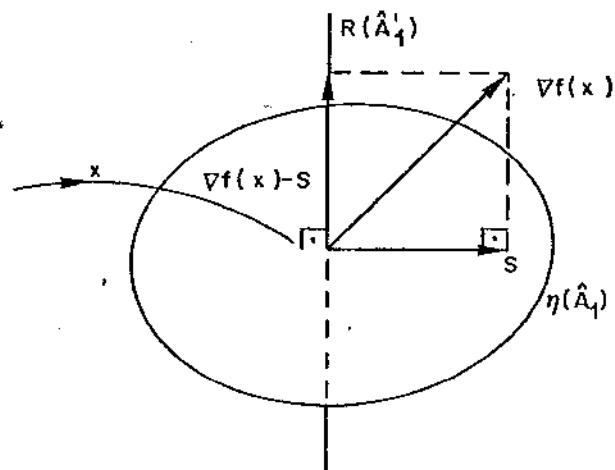


Fig. I.1 - Gradiente Projetado

Uma vez que $(\nabla f(x) - s) \in \mathcal{R}(\tilde{A}_1')$, existe um vetor $\beta \in \mathbb{R}^q$ com q sendo o número de restrições ativas em x , tal que

$$\tilde{A}_1' \beta = \nabla f(x) - s \quad (\text{I.39})$$

Pré-multiplicando (I.39) por \tilde{A}_1 e assumindo regularidade das restrições, isto é, assumindo que de um total máximo de $(m + n)$ restrições ativas, q é menor ou igual a n e \tilde{A}_1 tem rank completo, obtém-se o vetor :

$$\beta = (\tilde{A}_1 \tilde{A}_1')^{-1} \tilde{A}_1 \nabla f(x) , \quad (\text{I.40})$$

e levando-se (I.40) em (I.39) resulta para a direção de pesquisa s ,

$$s = P \nabla f(x) \quad (\text{I.41})$$

onde

$$P \triangleq I_n - \tilde{A}_1' (\tilde{A}_1 \tilde{A}_1')^{-1} \tilde{A}_1 , \quad P \in \mathbb{R}^{n \times n} \quad (\text{I.42})$$

\bar{e} é conhecida como matriz de projeção no espaço nulo de \tilde{A}_1 e que apresenta as seguintes propriedades.

Lema I-1 : Seja a matriz de projeção P , definida por (I.42); então

- i) $P = P' \geq 0$
- ii) $P'P = P$
- iii) $\lambda_i(P) = 0$ ou 1 , $i = 1, \dots, n$
- iv) a matriz $(I_n - P)$ satisfaz as propriedades acima e projeta qualquer vetor do R^n em $\mathcal{S}(\tilde{A}_1')$

A direção s assim obtida, caso seja diferente de zero, efetivamente decresce o valor da função, como pode ser facilmente demonstrado pré-multiplicando-se (I.41) por $\nabla f(x)'$,

$$\nabla f(x)'s = \nabla f(x)'P\nabla f(x) = \nabla f(x)'P'P\nabla f(x) = \|P\nabla f(x)\|^2 > 0, \quad (\text{I.43})$$

pois $s = P\nabla f(x) \neq 0$

A análise do caso $s = 0$ é uma característica do método de Rosen. Observe que $-\beta$ representa na verdade o vetor de multiplicadores de Lagrange associado às restrições ativas de (I.36). Isto pode ser visto claramente escrevendo-se as condições de Kuhn-Tucker de (I.36) em x :

- i) $\nabla f(x) + \tilde{A}_1' \pi = 0$;
 - ii) $\tilde{A}_1 x = \tilde{b}_1$;
 - iii) $\pi_i \geq 0$, $i = (m+1), \dots, q$
- (I.44)

É imediato então que fazendo-se $s = 0$ em (I.39) a correspondência se verifica para $\beta = -\pi$. Deste modo, se todas as componentes de β relativas às restrições de desigualdade forem não positivas, as condições de Kuhn-Tucker estarão satisfeitas. Por outro lado, se existir $\beta_i > 0$ para algum $i = (m+1), \dots, q$ é possível gerar uma nova direção usável removendo-se de \tilde{A}_1 a restrição associada a β_i (Rosen [16], Luenberger [14]).

Os procedimentos do método estão resumidos no Algoritmo I-1.

Algoritmo I-1 [Rosen, 1960] : Dados \tilde{A} , \tilde{b} e uma solução inicial factível x^0 , faça $k = 0$. Uma iteração genérica k será dada como segue.

PASSO 1 : Determine a matriz \tilde{A}_1 , formada pelas restrições ativas de (I.36) no ponto x^k .

PASSO 2 : Calcule $P = I - \tilde{A}_1'(\tilde{A}_1\tilde{A}_1')^{-1}\tilde{A}_1$ e a direção $s^k = P\nabla f(x^k)$.

PASSO 3 : Se $\|s^k\| < \epsilon$, arbitrariamente pequeno, encontre

$$\beta^k = (\tilde{A}_1\tilde{A}_1')^{-1}\tilde{A}_1'\nabla f(x^k)$$

Se $\beta_i < 0$, $\forall i > m$ então pare. O ponto x^k satisfaz as condições necessárias de otimalidade de Kuhn-Tucker. Senão, remova a restrição correspondente ao valor da componente mais positiva de β e retorne ao PASSO 2.

Se $\|s^k\| \geq \epsilon$, vá para o PASSO 4

PASSO 4 : Encontre o valor do passo α_k resolvendo

$$\text{Min}_{\alpha} f(x^k - \alpha s^k)$$

$$0 \leq \alpha \leq \bar{\alpha}$$

onde $\bar{\alpha} = \max\{\alpha \mid A(x^k - \alpha s^k) \leq b\}$. Atualize as variáveis primais,

$$x^{k+1} = x^k - \alpha_k s^k,$$

faça $k = k + 1$ e retorne ao PASSO 1

O critério utilizado no PASSO 3 para a retirada de restrições da matriz \bar{A}_1 atende apenas a necessidade de estabelecer-se algum tipo de sistematização para este procedimento não tendo, entretanto, qualquer justificativa teórica. Este critério é comumente encontrado na literatura.

A análise do algoritmo apresentado permite concluir que o principal esforço computacional associado ao método de Rosen consiste em inverter, a cada iteração, a matriz simétrica definida positiva \bar{A}_1^{-1} . Demonstra-se (Rosen [16]) que se em iterações consecutivas os sub-espacos correspondentes diferem de apenas uma restrição é possível, através de relações de recorrência, atualizar a matriz inversa a partir da matriz inversa anterior. Isto, entretanto, não ocorre para a maioria dos problemas práticos nos quais as composições de sub-espacos consecutivos podem diferir consideravelmente, sendo então necessárias projeções sucessivas para que a nova direção de pesquisa seja determinada. Para que estas projeções sejam possíveis é necessário ainda eliminar a possibilidade de degenerescência do método estabelecendo-se regras especiais para a detecção de restrições redundantes que podem surgir da combinação de restrições ativas do problema. A eliminação de restrições

redundantes é requisito básico para a aplicação do método pois permite que as hipóteses de não-degenerescência sejam satisfeitas.

Pelas razões apresentadas, o método de Rosen é pouco utilizado em problemas práticos de grande dimensão. Porém, a idéia de projeção do gradiente foi desde então continuamente aperfeiçoada por vários métodos no sentido de diminuir o esforço computacional associado à projeção e relaxar as hipóteses de não-degenerescência.

I.7. Gradiente Reduzido

O método do Gradiente Reduzido foi proposto e desenvolvido por Wolfe (1963) para problemas de otimização com restrições lineares e em trabalhos posteriores, generalizado para problemas com restrições não-lineares por Abadie e Carpentier (1969) (Abadie [1]).

Neste método, Wolfe explora ao máximo o caráter linear das restrições do problema, empregando vários dos procedimentos do método SIMPLEX para obter a projeção do vetor gradiente sobre um sub-espço de dimensão reduzida. Como em programação linear, o método de Wolfe subdivide as variáveis do problema em grupos de variáveis básicas (dependentes) e não-básicas (independentes). Eliminando-se as variáveis básicas, é possível considerar o problema de otimização resultante apenas em termos de variáveis não-básicas e, deste modo, reduzir a dimensão do vetor gradiente. A direção de pesquisa do método é então determinada a partir da projeção deste gradiente reduzido sobre as restrições de canalização associadas às variáveis não-básicas.

Considere o problema (I.30). Note que suas restrições estão colocados na forma canônica de programação linear:

$$\begin{aligned} \text{Min } & f(x) \\ \text{Ax} &= b \\ x &\in X \end{aligned} \quad (\text{I.45})$$

onde

$x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R} \in C^1$ e

$$X = \{x \mid \underline{x} \leq x \leq \bar{x}\} \quad (\text{I.46})$$

Por conveniência, as variáveis básicas $x^B \in \mathbb{R}^m$ serão representadas através do conjunto de m índices

$$I_B(x) = \{i \mid \underline{x}_i^B < x_i^B < \bar{x}_i^B\}, \quad i = 1, \dots, m \quad (\text{I.47})$$

e as não-básicas $x^N \in \mathbb{R}^{n-m}$ por

$$I_N(x) = I - I_B(x), \quad i = 1, \dots, n \quad (\text{I.48})$$

onde $I = \{i\}$, $i = 1, \dots, n$

Em relação ao problema (I.45), as seguintes hipóteses de não-degenerescência devem ser satisfeitas (Bazaraa [4], Lumberger [14]):

- i) Quaisquer m colunas de A constituem um conjunto linearmente independente, uma base B do \mathbb{R}^m ;
- ii) Toda solução básica factível obtida é não-degenerada, isto

é, qualquer solução factível do problema tem no máximo $(n-m)$ variáveis assumindo o valor dos seus limites.

Na prática estas hipóteses podem restringir bastante a aplicação do método. Admita, entretanto, que exista ao menos uma partição das variáveis satisfazendo as hipóteses i) e ii). O problema pode ser reescrito como

$$\begin{aligned} \text{Min } f(x^B, x^N) \\ Bx^B + Nx^N = b \\ \underline{x}^B \leq x^B \leq \bar{x}^B \\ \underline{x}^N \leq x^N \leq \bar{x}^N \end{aligned} \tag{I.49}$$

onde

$x^B \in \mathbb{R}^m$, $x^N \in \mathbb{R}^{(n-m)}$, $B = \{a^i \mid i \in I_B(x)\} \in \mathbb{R}^{m \times m}$, $N = \{a^i \mid i \in I_N(x)\} \in \mathbb{R}^{m \times (n-m)}$, e a^i representa a i -ésima coluna de A .

Usando-se então a hipótese i) é possível colocar x^B como função de x^N escrevendo

$$x^B = B^{-1}b - B^{-1}Nx^N \tag{I.50}$$

e assim resolver (I.49) apenas em função de x^N . Através de (I.50), as variações Δx^B e Δx^N das variáveis básicas e não básicas, respectivamente, estarão relacionadas de forma única através de

$$\Delta x^B = -B^{-1}N\Delta x^N \quad (I.51)$$

O vetor gradiente, relativo às variáveis não-básicas, pode então ser facilmente obtido, calculando-se a derivada total de $f(x^B(x^N), x^N)$ em relação a x^N , que será dada por

$$r = \frac{d f(x^B(x^N), x^N)}{dx^N} = \frac{\partial f(x^B, x^N)}{\partial x^N} + \left(\frac{dx^B}{dx^N} \right)' \frac{\partial f(x^B, x^N)}{\partial x^B} \quad (I.52)$$

Tomando-se em (I.51), o limite quando $\Delta x^N \rightarrow 0$, resulta finalmente para o vetor gradiente reduzido $r \in \mathbb{R}^{n-m}$ a expressão

$$r = \nabla_{x^N} f(x^B, x^N) - (B^{-1}N)' \nabla_{x^B} f(x^B, x^N) \quad (I.53)$$

O problema agora se resume em projetar este gradiente reduzido sobre as restrições de canalização de x^N , determinando sua direção de movimento, s^N e a partir de (I.51), a direção s^B associada às variáveis básicas. Esta projeção é realizada modificando-se ligeiramente o método do gradiente para otimização irrestrita por um procedimento que na verdade é uma especialização do método do Gradiente Projetado de Rosen a problemas que têm somente restrições de canalização (Bazaraa [4]).

O algoritmo associado ao método do Gradiente Reduzido é dado em seguida

Algoritmo I-2 [Wolfe, 1963] : Dados $A, b, \underline{x}, \bar{x}$ e x^0 factível inicial, faça

$k = 0$. Uma iteração genérica k será dada como segue:

PASSO 1 : Com $I_B(x^k)$ e $I_N(x^k)$, calcule o gradiente reduzido $r \in \mathbb{R}^{n-m}$

$$r^k = \nabla_{x^{N,k}} f(x^{B,k}, x^{N,k}) - (B^{-1}N)' \nabla_{x^{B,k}} f(x^{B,k}, x^{N,k})$$

PASSO 2 : Para todo $i \in I_N(x^k)$ obtenha (*)

$$s_i^{N,k} = \begin{cases} 0, & \text{se } x_i^{N,k} = \underline{x}_i^N \text{ e } r_i^k > 0 \text{ ou } x_i^{N,k} = \bar{x}_i^N \text{ e } r_i^k < 0 \\ r_i^k, & \text{caso contrário} \end{cases}$$

PASSO 3 : Se $\|s^{N,k}\| < \epsilon$, arbitrariamente pequeno, pare. O ponto corrente $x^{k'} = [x^{B,k'}, x^{N,k'}]$ satisfaz as condições necessárias de otimalidade de Kuhn-Tucker para a existência de um mínimo local. Senão, obtenha

$$s^{B,k} = -B^{-1}N s^{N,k}.$$

PASSO 4 : Encontre o valor ótimo de α_k , o passo do algoritmo, resolvendo

$$\min_{\alpha} f(x^k - \alpha s^k)$$

$$0 \leq \alpha \leq \min(\bar{\alpha}_1, \bar{\alpha}_2)$$

onde

(*) $s_i^{N,k}$ representa a i -ésima componente do vetor direção associado às variáveis não-básicas na iteração k .

$$\bar{\alpha}_1 = \max \{ \alpha \mid \underline{x}^N \leq x^{N,k+1} \leq \bar{x}^N \}$$

$$\bar{\alpha}_2 = \max \{ \alpha \mid \underline{x}^B \leq x^{B,k+1} \leq \bar{x}^B \}$$

Atualize as variáveis primais

$$x^{k+1} = x^k - \alpha_k s^k .$$

PASSO 5 : Se $\alpha_k < \bar{\alpha}_2$, faça $k = k + 1$ e retorne para o PASSO 2. Se $\alpha_k = \bar{\alpha}_2$, então pelo menos uma variável básica atingiu um dos limitantes. Neste caso, remova-a da base substituindo-a por uma variável não-básica inativa. Redefina $I_B(x^k)$ e $I_N(x^k)$, faça $k = k + 1$ e retorne para o PASSO 1.

No procedimento descrito no PASSO 5 residem as principais vantagens e desvantagens do método. Caso $\alpha_k < \bar{\alpha}_2$, isto é, caso nenhuma variável básica atinja um dos limitantes, os conjuntos $I_B(x)$ e $I_N(x)$ permanecem inalterados e o algoritmo prossegue recalculando o gradiente reduzido r . Caso $\alpha_k = \bar{\alpha}_2$, as seguintes situações podem ocorrer^(*) :

- i) uma variável básica atinge um dos limitantes e uma variável não-básica se torna inativa;
- ii) uma variável básica atinge um dos limitantes enquanto que uma nova variável não-básica se torna ativa;

(*) Analisa-se aqui o caso em que somente uma nova variável torna-se ativa.

iii) uma variável básica atinge um dos limitantes enquanto que uma variável não-básica se deslocando de um dos limitantes, atinge o limitante oposto.

A ocorrência da situação i) está perfeitamente inserida na filosofia do método. A variável básica que alcançou um dos limitantes é declarada não-básica e uma variável não-básica inativa é declarada básica, redefinindo-se os conjuntos $I_B(x)$ e $I_N(x)$. A atualização da base B é então eficientemente realizada utilizando-se técnicas desenvolvidas para problemas de programação linear com restrições de canalização (Lumberger [14]).

As duas últimas situações, entretanto, possibilitam o aparecimento de soluções degeneradas, dado que nas situações descritas não é possível garantir que o número de variáveis ativas seja menor ou igual a $(n-m)$ e, portanto, assegurar que as hipóteses de não-degenerescência sejam satisfeitas. Neste caso, procedimentos adicionais que detetem a ocorrência destas situações e modifiquem a direção de pesquisa s^N das variáveis não-básicas devem ser incorporados ao algoritmo, onerando-o computacionalmente.

I.8. Gradiente Projetado Via Decomposição

Os métodos que foram até agora analisados são também conhecidos como métodos diretos pois a direção de pesquisa é obtida exclusivamente com informações do problema primal. O método de Geromel-Baptistella (1981) é indireto, isto é, utiliza também informações do problema dual associado ao chamado problema de projeção, a partir do qual a direção de pesquisa é determinada. Com isto, o método proposto procura explorar as vantagens tanto da abordagem direta, obtendo-se a cada iteração uma solução factível, quanto da

abordagem indireta, resolvendo-se um problema de projeção estritamente convexo cujo dual tem solução imediata.

Problemas práticos de grande dimensão podem ser tratados de forma muito eficiente uma vez que o método utiliza uma técnica de decomposição dual, bastante difundida em programação matemática, e cuja convergência é assegurada para problemas estritamente convexos, como é o caso.

Por trabalhar com subproblemas independentes e não de forma global como nos métodos anteriores, o método de Geromel-Baptistella [10] possui a característica de não fazer nenhuma hipótese de regularidade das restrições, característica esta de fundamental importância em se tratando de problemas práticos.

Considere novamente o problema de programação matemática (I.30) reescrito aqui por conveniência:

$$\begin{aligned} \text{Min } f(x) \\ Ax = b \\ x \in X \end{aligned} \tag{I.54}$$

onde $x \in R^n$, $A \in R^{m \times n}$, $b \in R^m$, $f(x) : R^n \rightarrow R \in C^1$ e

$$X = \{x \mid \underline{x} \leq x \leq \bar{x}\} \tag{I.55}$$

Como anteriormente, assume-se conhecida uma solução inicial factível de (I.54) ou seja, supõe-se que existe ao menos um ponto $x \in R^n$ que satisfaz

$$x \in \gamma \stackrel{\Delta}{=} \{x \mid Ax = b ; x \in X\}. \quad (I.56)$$

A partir deste ponto inicial, Geromel-Baptistella [10] propõem que a direção de pesquisa do método seja a solução ótima de um problema quadrático estritamente convexo que será referenciado aqui como problema de projeção:

$$\begin{aligned} \text{Min}_s \quad & \frac{1}{2} \|\nabla f(x) - s\|^2 \\ \text{As} \quad & = 0 \end{aligned} \quad (I.57)$$

$$s \in S(x),$$

onde o conjunto $S(x)$ é escrito como

$$S(x) = \{s \mid s_i \leq 0, i \in I_-(x) ; s_i \geq 0, i \in I^-(x)\}, \quad (I.58)$$

e no qual os conjuntos $I_-(x)$ e $I^-(x)$ são dados por

$$\begin{aligned} I_-(x) \quad & \{i \mid x_i = \underline{x}_i, i = 1, \dots, n\} \\ I^-(x) \quad & \{i \mid x_i = \bar{x}_i, i = 1, \dots, n\}, \end{aligned} \quad (I.59)$$

com x_i representando a i -ésima componente de $x \in R^n$. Note que as restrições de (I.57) foram obtidas a partir de (I.54) assegurando-se a existência de um tamanho mínimo de passo $\bar{\alpha} > 0$, tal que, se $x \in \gamma$ então

$$x - \alpha s \in \gamma, \quad \forall 0 \leq \alpha \leq \bar{\alpha}, \quad (I.60)$$

sendo s a solução ótima de (I.57) e, conseqüentemente, uma direção factível de (I.54). Procura-se, portanto, obter no sub-espaço das direções factíveis de (I.57),

$$\Omega(x) = \{s \mid As = 0 \ ; \ s \in S(x)\}, \quad (\text{I.61})$$

a direção s que minimiza o ângulo θ formado entre o vetor gradiente em x , $\nabla f(x)$, direção de máximo decréscimo de $f(x)$ e em geral inactível, e este sub-espaço, obtido a partir das restrições ativas de (I.54). A figura I.2 ilustra geometricamente o problema de projeção.

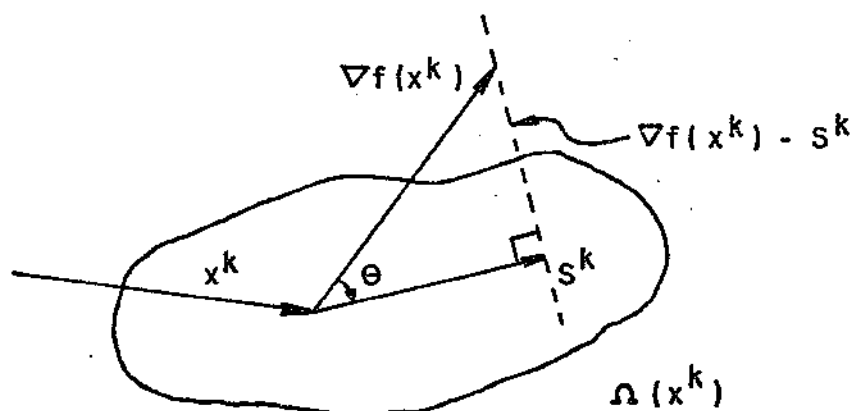


Fig. I.2 - Interpretação Geométrica de (I.57)

A natureza linear do sub-espaço definido por $\Omega(x)$ determina que s^k , a solução ótima de (I.57) seja dado, evidentemente, pela projeção ortogonal de $\nabla f(x)$ sobre este sub-espaço.

Considere a função Lagrangeano $L(s, \lambda) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$.

$$L(s, \lambda) = \frac{1}{2} \|\nabla f(x) - s\|^2 + \lambda' As, \quad (\text{I.62})$$

onde $\lambda \in \mathbb{R}^m$ representa o vetor de multiplicadores de Lagrange relativo às restrições de igualdade. A função dual $\psi(\lambda) : \mathbb{R}^m \rightarrow \mathbb{R}$ será então dada por

$$\psi(\lambda) = \min_{s \in S(x)} L(s, \lambda), \quad (\text{I.63})$$

e o correspondente problema dual de (I.57) por

$$\text{Max}_{\lambda} \psi(\lambda) \quad (\text{I.64})$$

Note, entretanto, que sendo $L(s, \lambda)$ completamente separável em s , (I.63) pode ser reescrita como

$$\text{Max}_{\lambda} \sum_{i=1}^n \min_{s_i \in S(x)} L_i(s_i, \lambda) \quad (\text{I.65})$$

onde

$$\text{Min}_{s_i \in S(x)} L_i(s_i, \lambda), \quad i = 1, \dots, n \quad (\text{I.66})$$

representa o i -ésimo subproblema,

$$L_i(s_i, \lambda) = \frac{1}{2} \left(\frac{\partial f(x)}{\partial x_i} - s_i \right)^2 + \lambda' a_i s_i, \quad i = 1, \dots, n \quad (\text{I.67})$$

sua i -ésima função objetivo e a^i , a i -ésima coluna de A . As soluções dos n subproblemas são facilmente obtidas através de

$$s_i = \begin{cases} \min\{0, \frac{\partial f(x)}{\partial x_i} - a_i^1 \lambda\}, & i \in I_-(x) \\ \max\{0, \frac{\partial f(x)}{\partial x_i} - a_i^1 \lambda\}, & i \in I^+(x) \\ \frac{\partial f(x)}{\partial x_i} - a_i^1 \lambda, & \text{caso contrário.} \end{cases} \quad (I.68)$$

Esta estrutura caracteriza os métodos de decomposição dual em dois níveis; um nível inferior no qual as soluções analíticas dos n subproblemas são determinadas por (I.68) e um nível superior onde, através da variável dual λ , o problema dual (I.64) as coordena forçando a convergência aos valores ótimos λ^* e s_i^* , $i=1, \dots, n$. A convergência de tal procedimento é assegurada dado que o problema de projeção (I.57) é estritamente convexo. Este fato é também suficiente para garantir a diferenciabilidade da função dual e viabilizar o uso de métodos do tipo gradiente para a determinação de sua solução global λ^* , que como mencionado anteriormente é tal que

$$\psi(\lambda^*) = \frac{1}{2} \|\nabla f(x) - s^*\|^2 \quad (I.69)$$

O algoritmo relativo ao método de Geromel-Baptistella é fornecido em seguida.

Algoritmo I-3 [Geromel-Baptistella, 1981] : Dados $A, b, \underline{x}, \bar{x}$ e uma solução inicial factível x^0 , faça $k=0$. Uma iteração genérica será dada como segue.

PASSO 1 : Encontre $\nabla f(x^k)$, $I_-(x^k)$ e $I^-(x^k)$

PASSO 2 : Encontre a solução do problema de projeção (I.57). Fixado um valor inicial para λ , uma iteração genérica ℓ do algoritmo de decomposição seria:

PASSO 2.1 : Com λ^ℓ fixo, encontre as soluções s_i dos n subproblemas (I.66) através de (I.68).

PASSO 2.2 : Encontre o gradiente da função dual, $\nabla\psi(\lambda^\ell)$. Se $\|\nabla\psi(\lambda^\ell)\| < \epsilon_2$, o procedimento convergiu, sendo as variáveis correntes soluções do problema de projeção; vá para o PASSO 3. Senão, atualize λ

$$\lambda^{\ell+1} = \lambda^\ell + \sigma_\ell d^\ell,$$

sendo d^ℓ gerada pelo método de direções conjugadas de Fletcher-Reeves (Luenberger [14]) e σ_ℓ solução ótima de

$$\text{Max}_{\sigma \geq 0} \psi(\lambda^\ell + \sigma d^\ell),$$

Faça $\ell = \ell + 1$ e volte para o PASSO 2.1.

PASSO 3 : Se $\|s^k\| < \epsilon_1$, arbitrariamente pequeno, pare: x^k satisfaz as condições necessárias de Kuhn-Tucker. Senão, vá para o PASSO 4.

PASSO 4 : Encontre o passo α_k do algoritmo resolvendo

$$\text{Min}_{\alpha} f(x^k - \alpha s^k)$$

$$, 0 \leq \alpha \leq \min\{\bar{\alpha}_1, \bar{\alpha}_2\},$$

onde

$$\bar{\alpha}_1 = \min_{1 \leq i \leq n} \{(x_i^k - \underline{x}_i) / s_i^k, s_i^k > 0\}$$

$$\bar{\alpha}_2 = \min_{1 \leq i \leq n} \{(x_i^k - \bar{x}_i) / s_i^k, s_i^k < 0\},$$

Atualize as variáveis primais,

$$x^{k+1} = x^k - \alpha_k s^k,$$

faça $k = k + 1$ e retorne ao PASSO 1.

A justificativa teórica para o PASSO 4 é imediata pois o problema de projeção (I.57) tem a propriedade de sempre gerar como solução ótima, uma direção usável de (I.54). Assim, um vetor gradiente $\nabla f(x)$ não-ortogonal a $\Omega(x)$ gera uma direção usável dado que $\nabla f(x)'s > 0$; caso $\nabla f(x) \perp \Omega(x)$, resultando em $s = 0$, demonstra-se facilmente (Geromel-Baptistella [10]) que x satisfaz as condições necessárias de Kuhn-Tucker. Os métodos de Wolfe e Geromel-Baptistella fornecem, portanto, uma maneira simples de detectar-se a otimalidade.

O principal esforço computacional do método de Geromel-Baptistella está associado ao PASSO 2 do algoritmo e em particular à solução iterativa do nível de coordenação do problema dual. Por outro lado, a dimensão do nível coordenador depende apenas do número de restrições de igualdade do

problema tornando o cálculo da projeção independente da introdução de limitantes nas variáveis primais, o que não ocorre, por exemplo, com o método de Rosen onde estas restrições, quando ativas, contribuem para dificultar o cálculo da matriz de projeção (Geromel-Baptistella [10]).

Este método é entre todos aqueles que foram apresentados o de mais difícil implementação, principalmente quando aplicado a problemas de otimização dinâmica.

O método primal proposto no capítulo seguinte procurará manter a maior parte de suas vantagens introduzindo, porém, um procedimento computacional mais simples de ser implementado.

I.9. Conclusão

Neste primeiro capítulo, procurou-se inicialmente apresentar os fundamentos teóricos de programação matemática utilizados neste, e nos capítulos seguintes. As condições necessárias de otimalidade de Kuhn-Tucker, como pode ser constatado, serão particularmente importantes na maioria das demonstrações. Propriedades envolvendo dualidade, também frequentemente utilizadas na sequência deste trabalho foram apresentadas e alguns de seus resultados diretamente empregados na discussão do método de Geromel-Baptistella. Deixaram de ser aqui expostos, por fugirem aos objetivos do capítulo, resultados que dizem respeito ao conteúdo do Capítulo III (Sistemas Dinâmicos, Cálculo Variacional, Princípio do Mínimo de Pontryagin, etc.) que trata da aplicação em Controle Ótimo da metodologia proposta no Capítulo II, e que são cobertos por Athans [2] e Sage [17], entre outros.

No que se refere aos métodos apresentados (Rosen, Wolfe e Gero-

mel-Baptistella) objetivou-se, principalmente, introduzir a filosofia de trabalho dos métodos primais e analisar os algoritmos descritos sob dois aspectos: hipóteses que os métodos assumem válidas e esforço computacional necessário à projeção.

CAPÍTULO II

II.1. Introdução

Este capítulo é destinado à proposição e à análise de um novo método primal, para resolução de problemas de programação matemática com função objetivo diferenciável e restrições lineares. O método proposto pode ser entendido como uma forma alternativa de abordar o problema de projeção, resolvido pelo método de Geromel-Baptistella (1981), através de Decomposição Dual. Aqui, a proposição é exatamente no sentido inverso.

Utilizando ainda a Teoria da Dualidade, a direção de pesquisa do método é obtida globalmente, mas por uma expressão que separa a contribuição das restrições de igualdade, permanentemente ativas, da contribuição das restrições de desigualdade ativas, cuja composição é variável a cada iteração do algoritmo.

Este procedimento permite, em última análise, considerar no processo de otimização apenas as restrições de desigualdade ativas uma vez que o problema pode agora ser resolvido, *a-priori*, para as restrições de igualdade.

Separando ainda a contribuição das restrições de igualdade e de desigualdade à direção de pesquisa do método, evitam-se as hipóteses de não-degenerescência. A única hipótese que surge, independência linear das restrições de igualdade, é suficientemente fraca para que esteja satisfeita após a Fase I e antes, portanto, da aplicação efetiva do método.

II.2. Desenvolvimento Teórico do Método: Caso Estático

O problema de programação matemática que mais uma vez se coloca é:

$$\text{Min } f(x)$$

$$Ax = b \quad (\text{II.1})$$

$$x \in X,$$

onde $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$; $\text{rank}(A) = m \leq n$, $f(x) : \mathbb{R}^n \rightarrow \mathbb{R} \in C^1$ e

$$X = \{x \mid \underline{x} \leq x \leq \bar{x}\}. \quad (\text{II.2})$$

A direção de pesquisa para este problema é obtida, através do método de Geromel-Baptistella, resolvendo-se o problema de projeção formulado no capítulo anterior, na forma dada por (I.57). Uma maneira alternativa de abordar este problema parte do princípio de que é também possível escrevê-lo como (Ferreira-Geromel [7]):

$$\text{Min } \frac{1}{2} \|\nabla f(x) - s\|^2 \quad (\text{II.3})$$

$$As = 0$$

$$B(x)s \leq 0,$$

onde a matriz $B(x)$, introduzida em (II.3), é formada pelos elementos -1, 0 e 1, seu número de linhas é igual ao número de componentes ativas de x e associa, por linha, a i -ésima componente de s o valor 1, se $i \in I_-(x)$ ou -1, se $i \in I^+(x)$, de modo a satisfazer

$$S(x) = \{s \mid s_i \leq 0, i \in I_-(x); s_i \geq 0, i \in I^+(x)\} \quad (\text{II.4})$$

Sendo q o número de componentes ativas de x , então $B(x) \in R^{q \times n}$. Ainda com relação à matriz $B(x)$, verifica-se diretamente que

$$B(x)B(x)' = I_q, \quad \forall x \in \gamma, \quad (II.5)$$

propriedade esta que se mostrará útil em desenvolvimentos futuros.

No método proposto por Geromel-Baptistella, a função dual é construída dualizando-se apenas as restrições de igualdade de (II.3), dando origem, em consequência, a um problema dual sem qualquer restrição sobre $\lambda \in R^m$, mas com a minimização do Lagrangeano $L(s, \lambda)$ restrita à (II.4). Considere agora as implicações decorrentes da dualização também das restrições de desigualdade de (II.3). Neste caso, se $\mu \in R^q$ for a variável dual associada, o novo problema dual ficará restrito à $\mu \geq 0$ e a minimização da função Lagrangeano $L(s, \lambda, \mu)$ sem restrição sobre s . Dado então que a função $L(s, \lambda, \mu)$ é estritamente convexa em s , é possível expressar a solução ótima do problema de projeção diretamente como função das variáveis duais λ e μ . Para tanto, seja a função dual $\psi(\lambda, \mu) : R^m \times R^q \rightarrow R$ relativa ao problema (II.3):

$$\psi(\lambda, \mu) = \min_s L(s, \lambda, \mu), \quad (II.6)$$

cujo domínio de definição é fornecido por $D = \{\lambda \in R^m, \mu \in R_+^q \mid \exists \min_s L(s, \lambda, \mu)\}$. A função Lagrangeano indicada em (II.6) é escrita como

$$L(s, \lambda, \mu) = \frac{1}{2} \|\nabla f(x) - s\|^2 + \lambda'As + \mu'B(x)s, \quad (II.7)$$

e o correspondente problema dual dado por

$$\begin{array}{l} \text{Max} \\ \lambda \\ \mu \geq 0 \end{array} \min_s L(s, \lambda, \mu). \quad (\text{II.8})$$

Segue então, que para quaisquer λ e μ fixos, o problema irrestrito

$$\min_s L(s, \lambda, \mu) \quad (\text{II.9})$$

tem solução única, podendo ser facilmente determinada impondo-se a condição de 1ª ordem, necessária e suficiente neste caso, sobre $L(s, \lambda, \mu)$:

$$\left. \frac{\partial L(s, \lambda, \mu)}{\partial s} \right|_{s=s^*} = 0. \quad (\text{II.10})$$

Efetuando-se a operação indicada, a solução ótima s^* será obtida como

$$s^* = \nabla f(x) - A' \lambda^* - B(x)' \mu^*, \quad (\text{II.11})$$

sendo, que λ^* e μ^* são as soluções ótimas do problema dual (II.8). Note, entretanto, que (II.8) pode ser separado nas variáveis λ e μ , tomando assim a forma:

$$\begin{array}{l} \text{Max} \\ \mu \geq 0 \end{array} \max_{\lambda} L(s^*, \lambda, \mu), \quad (\text{II.12})$$

e que o problema de maximização em λ , estritamente côncavo, pode igualmente ser resolvido, para μ fixo, impondo-se a condição de 1ª ordem sobre

$L(s^*, \lambda, \mu)$:

$$\left. \frac{\partial L(s^*, \lambda, \mu)}{\partial \lambda} \right|_{\lambda=\lambda^*} = 0, \quad (\text{II.13})$$

obtendo-se, do mesmo modo,

$$\lambda^* = (AA')^{-1}A[\nabla f(x) - B(x)' \mu^*]. \quad (\text{II.14})$$

Uma direção de pesquisa s^* , função apenas de μ^* , é obtida substituindo-se (II.14) em (II.11) e após alguma manipulação algébrica será dada por

$$s^* = P[\nabla f(x) - B(x)' \mu^*], \quad (\text{II.15})$$

onde $P \triangleq I_n - A'(AA')^{-1}A$ é a matriz de projeção no espaço nulo de A , definida no capítulo anterior. A direção de pesquisa assim obtida guarda estreita relação com aquela utilizada pelo método de Rosen. Observe que se, a cada iteração, as restrições de desigualdade ativas de (II.1) forem incorporadas à matriz A , o método proposto reduz-se ao método do Gradiente Projetado (Rosen, 1960).

Portanto, a direção de pesquisa (II.15) dependerá apenas de μ^* , a solução ótima do problema dual que resulta da substituição de (II.14) em $L(s^*, \lambda, \mu)$:

$$\text{Max}_{\mu \geq 0} \psi(\mu), \quad (\text{II.16})$$

onde $\psi(\mu)$, uma função quadrática, é escrita como

$$\psi(\mu) = \frac{1}{2} \mu' Q(x) \mu + c'(x) \mu + e(x), \quad (\text{II.17})$$

e onde

$$Q(x) \triangleq -B(x)PB(x)' \quad (\text{II.18.1})$$

$$c(x) \triangleq B(x)P\nabla f(x) \quad (\text{II.18.2})$$

$$e(x) \triangleq \nabla f(x)' A' (AA')^{-1} A \nabla f(x) \quad (\text{II.18.3})$$

$$Q(x) \in \mathbb{R}^{q \times q}, \quad c(x) \in \mathbb{R}^q \quad \text{e} \quad e(x) \in \mathbb{R}$$

Uma primeira consequência, decorrente da forma na qual o problema dual foi colocado em (II.12) é a possibilidade de que a variável dual λ seja eliminada, desacoplando totalmente as restrições de igualdade da parte que efetivamente varia a cada iteração, as restrições de desigualdade ativas do problema. A contribuição desta parte ao cálculo de s^* aparece em (II.15) na forma do vetor do \mathbb{R}^n , $B(x)' \mu^*$. Deste modo, o método proposto evita as hipóteses de regularidade que surgem quando as restrições do problema são tratados de forma não-diferenciada, como no método de Rosen.

O valor de μ^* , a solução ótima de (II.16), pode ser determinada através de qualquer método de gradiente, dado que a estrita convexidade de (II.3) garante a diferenciabilidade da função dual (II.17).

Antes da análise do método dual-clássico utilizado para o cálculo de μ^* , serão evidenciadas algumas propriedades da matriz $Q(x)$ definida por (II.18.1).

Lema II-1 : Seja a matriz $Q(x)$ definida por (II.18.1); então,

$$i) Q(x) = Q(x)' \leq 0 ;$$

$$ii) -1 \leq \lambda_i(Q(x)) \leq 0 , \quad i = 1, \dots, q$$

Prova : A prova da parte i) é imediata e utiliza a parte ii) do Lema I-1:

$$\mu'Q(x)\mu = -\mu'B(x)PB(x)'\mu = -\mu'B(x)P'PB(x)'\mu = -\|PB(x)'\mu\|^2 \leq 0 \quad (II.19)$$

A parte ii) é demonstrada lembrando-se que o i -ésimo autovalor de $Q(x)$ satisfaz

$$Q(x)\mu_i = \lambda_i(Q(x))\mu_i , \quad (II.20)$$

ou

$$\lambda_i(Q(x)) = \frac{\mu_i'Q(x)\mu_i}{\mu_i'\mu_i} , \quad (II.21)$$

que ainda pode ser escrito como

$$\lambda_i(Q(x)) = - \frac{\mu_i'B(x)P'PB(x)'\mu_i}{\mu_i'B(x)B(x)'\mu_i} , \quad (II.22)$$

na qual utilizou-se, simultaneamente, a prova de i) e a propriedade verificada em (II.5). Finalmente,

$$\begin{aligned} \lambda_i(Q(x)) &= - \frac{\|PB(x)'\mu_i\|^2}{\|B(x)'\mu_i\|^2} \geq \frac{-\|P\|^2 \|B(x)'\mu_i\|^2}{\|B(x)'\mu_i\|^2} \\ &\geq -\|P\|^2 = -\lambda_M(P) \\ &\geq -1 , \end{aligned} \quad (II.23)$$

o que completa a prova da parte ii) do Lema II-1.

II.3. Método Dual-Clássico

A análise da equação (II.15), que define a direção de pesquisa s^* , evidencia claramente que a eficiência do método proposto fica, em grande parte, condicionada ao grau de dificuldade encontrado para resolver o problema dual (II.16). Nesta seção, descreve-se o método dual-clássico utilizado e demonstram-se algumas propriedades relacionadas com o problema dual (II.16) que implicam em simplificações no algoritmo final.

Como observado por Lasdon [13], o método dual-clássico, como será aqui apresentado, é simplesmente uma adaptação do método do Gradiente Projetado de Rosen a problemas onde as únicas restrições são as de não-negatividade. A forma na qual será apresentado leva ainda em conta todas as propriedades da função quadrática a ser maximizada.

Seja, portanto, o problema dual anteriormente formulado:

$$\begin{array}{l} \text{Max } \psi(\mu) , \\ \mu \geq 0 \end{array} \quad (II.24)$$

onde

$$\psi(\mu) = \frac{1}{2} \mu' Q(x) \mu + c'(x) \mu + e(x) , \quad (II.25)$$

com $Q(x)$, $c(x)$ e $e(x)$ definidas por (II.18.1), (II.18.2) e (II.18.3), respectivamente. Com um valor inicial $\mu^0 \geq 0$, faça $k = 0$. Uma iteração genérica k do algoritmo será dada como segue.

PASSO 1 : Encontre $\psi(\mu^k)$ e $\nabla\psi(\mu^k)$, o vetor gradiente de $\psi(\mu)$ no ponto μ^k ,

$$\nabla\psi(\mu^k) = Q(x)\mu^k + c(x). \quad (II.26)$$

PASSO 2 : Defina a direção usável de pesquisa d^k por

$$d_i^k = \begin{cases} \frac{\partial \psi(\mu^k)}{\partial \mu_i^k} & \text{se } \mu_i^k > 0 \\ \text{Max}\{0, \frac{\partial \psi(\mu^k)}{\partial \mu_i^k}\} & \text{se } \mu_i^k = 0 \end{cases} \quad (II.27)$$

PASSO 3 : Se $\|d^k\| < \epsilon_2$, arbitrariamente pequeno, pare. O valor corrente de μ é ótimo. Senão vá para o PASSO 4.

PASSO 4 : Atualize a variável dual μ ,

$$\mu^{k+1} = \mu^k + \sigma_k d^k, \quad (II.28)$$

com σ_k , solução do problema,

$$\text{Max}_{\sigma} \psi(\mu^k + \sigma d^k) \quad (II.29)$$

$$0 \leq \sigma \leq \bar{\sigma},$$

onde

$$\bar{\sigma} = \min_{1 \leq i \leq q} \left\{ -\frac{\mu_i^k}{d_i^k}, d_i^k < 0 \right\}. \quad (II.30)$$

Faça $k = k + 1$, retornando em seguida ao PASSO 1

O Lema seguinte é uma justificativa teórica para o critério de otimalidade utilizado no PASSO 3 do algoritmo.

Lema II-2 : Se d , como definida por (II.27) é tal que $d = 0$, então μ satisfaz as condições necessárias e suficientes de otimalidade de Kuhn-Tucker. Senão, d é uma direção usável de (II.24).

Prova : Se $d = 0$, então, por construção,

- i) $\mu \geq 0$;
- ii) $\nabla\psi(\mu) \leq 0$; (II.31)
- iii) $\mu' \nabla\psi(\mu) = 0$,

ou seja, μ satisfaz as condições necessárias e suficientes de otimalidade de Kuhn-Tucker para a existência de um máximo global de (II.24). Se $d \neq 0$, então mais uma vez, por construção,

$$\nabla\psi(\mu)'d = d'd = \|d\|^2 > 0 \quad (II.32)$$

e, portanto, d é uma direção usável de (II.24).

A parte suficiente das condições de otimalidade de Kuhn-Tucker, quando aplicadas ao problema, deve-se ao fato de que a matriz Hessiano da função Lagrangeano

$$L_D(\mu, \xi) = \psi(\mu) + \xi' \mu , \quad (II.33)$$

dada por $Q(x)$, ser semi-definida negativa no sub-espço tangente às restri-

ções (componentes) ativas em μ^* (Luemberger [14], Avriel [3]). Entretanto, esta condição é suficiente apenas para garantir que o valor $\psi(\mu^*)$ é único, podendo porém ser obtido através de um incontável número de soluções. Através do Lema II-3 demonstra-se que a matriz $Q(x)$ é na verdade estritamente negativa no sub-espço das direções usáveis (II.27) pelo qual o algoritmo se desenvolve, assegurando-se deste modo, que no seu sub-espço de trabalho o algoritmo encontra uma única solução.

Lema II-3 : A matriz $Q(x)$ é definido negativa no sub-espço das direções usáveis (II.27).

Prova : Por hipótese, a direção d é usável. Assim,

$$\nabla\psi(\mu)'d > 0, \quad (\text{II.34})$$

e explicitando-se o vetor gradiente, vem

$$[Q(x)\mu + c(x)]'d > 0. \quad (\text{II.35})$$

Usando-se então (II.18.1) e (II.18.2) resulta

$$[\nabla f(x) - B(x)'\mu]'PB(x)'d > 0, \quad (\text{II.36})$$

que implica em $PB(x)'d \neq 0$. Entretanto,

$$d'Q(x)d = -d'B(x)PB(x)'d = -d'B(x)P'PB(x)'d, \quad (\text{II.37})$$

e, conseqüentemente,

$$d'Q(x)d = -\|PB(x)'d\|^2 < 0, \quad \forall d \text{ usável} \quad (\text{II.38})$$

É claro que este resultado é verdadeiro não apenas para as direções definidas através de (II.27), mas também para qualquer método que gere direções usáveis, isto é, que gere direções satisfazendo (II.34) para uma dada direção factível d .

Como consequência imediata deste resultado, o procedimento de busca unidimensional tratado no PASSO 4 fica simplificado, pois é possível agora obter-se uma expressão analítica para σ^* , o valor ótimo irrestrito para o passo σ , impondo-se a estacionariedade de (II.29):

$$\left. \frac{d\psi(\mu + \sigma d)}{d\sigma} \right|_{\sigma=\sigma^*} = 0, \quad (\text{II.39})$$

levando a

$$\sigma^* = - \frac{d'd}{d'Q(x)d}, \quad (\text{II.40})$$

uma expressão perfeitamente válida pois pelo Lema II-3, o algoritmo não desenvolve-se no sub-espço gerado pelos autovetores associados aos autovalores nulos de $Q(x)$. Deste modo, o valor do passo do algoritmo na k -ésima iteração fica sendo dado por

$$\sigma^k = \min\{\sigma^*, \bar{\sigma}\}. \quad (\text{II.41})$$

Convém ressaltar que o algoritmo desenvolvido para a obtenção do valor ótimo da variável dual μ pode vir a comportar-se como um método de gradiente puro, nos casos em que o valor ótimo de μ for igual à solução irrestrita do problema, apresentando, portanto, lenta convergência. A manei-

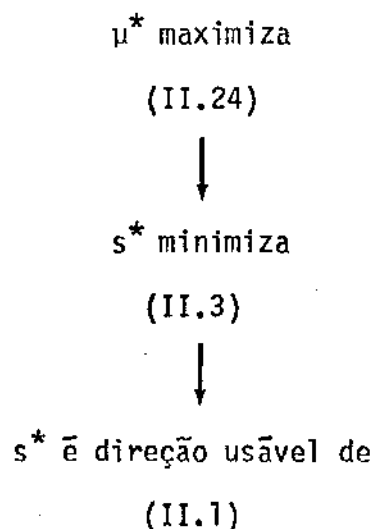
ra encontrada para acelerar esta convergência consiste em fornecer esta solução irrestrita, calculada através do método de direções conjugadas de Fletcher-Reeves, como inicialização ao método dual clássico, atribuindo-se antes o valor 0 a qualquer componente negativa desta solução. Assim, o algoritmo convergirá em q iterações sempre que a solução irrestrita não contiver componentes negativas. A inicialização para o método de Fletcher-Reeves será sempre $\mu^0 = 0$ (vetor nulo).

A eficácia deste procedimento foi constatada em todos os problemas tratados neste trabalho.

II.4. Condições de Otimalidade

Nesta seção demonstram-se as condições de otimalidade relativas aos problemas primal (II.1) e de projeção (II.3), tratados nas seções anteriores. Apresenta-se também o procedimento computacional do método proposto.

Na demonstração será obedecida a seguinte ordem lógica:



A primeira implicação será demonstrada escrevendo-se as condições de Kuhn-Tucker do problema dual (II.24):

$$\begin{aligned} \text{i) } \mu^* &\geq 0 ; \\ \text{ii) } \nabla\psi(\mu^*) &\leq 0 ; \\ \text{iii) } \mu^{*'} \nabla\psi(\mu^*) &= 0 . \end{aligned} \tag{c1}$$

e comparando-as em seguida com as condições de Kuhn-Tucker do problema de projecção (II.3):

$$\begin{aligned} \text{i) } \mu^* &\geq 0 ; \\ \text{ii) } As^* &= 0 ; \quad B(x)s^* \leq 0 ; \\ \text{iii) } \mu^{*'} B(x)s^* &= 0 ; \\ \text{iv) } -\nabla f(x) + s^* + A'\lambda^* + B(x)'\mu^* &= 0 . \end{aligned} \tag{c2}$$

Assim, como μ^* maximiza (II.24), a condição i) de (c2) fica automaticamente satisfeita. Com relação à factibilidade das restrições de igualdade, as condições (c2) são também satisfeitas, bastando para isto, que se utilize a definição de s^* , equação (II.15).

A condição ii) de (c1) é equivalente à condição de factibilidade das restrições de desigualdade de (c2) pois

$$\nabla\psi(\mu^*) = Q(x)\mu^* + c(x) = B(x)P[\nabla f(x) - B(x)'\mu^*] = B(x)s^* \leq 0 \tag{II.42}$$

A folga complementar iii) de (c1) é equivalente à de (c3):

$$\mu^{*'} \nabla\psi(\mu^*) = \mu^{*'} B(x)s^* = 0 \tag{II.43}$$

Finalmente, a estacionariedade do Lagrangeano, condição iv) de (c2) será satisfeita se usadas as definições de s^* , equação (II.15), e λ^* , equação (II.14), o que completa a prova da primeira implicação.

Resta, portanto, demonstrar que s^* , a solução do problema de projeção é uma direção usável para o problema primal (II.1) e sob que condições a otimalidade será alcançada.

Lema II-4 : Seja s^* , a solução ótima do problema de projeção; então,

- i) se $s^* = 0$, então x satisfaz as condições necessárias de otimalidade de Kuhn-Tucker para a existência de um mínimo local de (II.1);
- ii) se $s^* \neq 0$, então s^* é uma direção usável de (II.1).

Prova : O primeiro item do Lema pode ser demonstrado pela simples comparação entre as condições de Kuhn-Tucker dos problemas primal e de projeção (Geromel-Baptistella [10]):

P. Primal	P. Projeção
i) $\nabla f(x^*) + A'\beta^* + \pi^* = 0$;	i) $s^* - \nabla f(x) + A'\lambda^* + B(x)'\mu^* = 0$;
ii) $Ax^* = b$, $x^* \in X$;	ii) $As^* = 0$; $B(x)s^* \leq 0$;
iii) $\pi_i^* \geq 0$, se $i \in I^-(x^*)$;	iii) $\mu^{*'}B(x)s^* = 0$
$\pi_i^* < 0$, se $i \in I_-(x^*)$;	iv) $\mu^* \geq 0$
$\pi_i^* = 0$, caso contrário	

Os vetores $\beta \in R^m$ e $\pi \in R^n$ são os multiplicadores de Lagrange associados às

restrições de igualdade e de desigualdade de (II.1), respectivamente. Note que a cada iteração do método, a projeção s^* é uma direção factível do problema primal (II.1) e, como $s^* = 0$ é também uma solução factível do problema de projeção, as condições necessárias de Kuhn-Tucker do problema primal estarão satisfeitas para

$$\begin{aligned}\beta^* &= -\lambda^* ; \\ \pi^* &= -B(x)' \mu^*\end{aligned}\tag{II.44}$$

Verifica-se facilmente que os sinais dos multiplicadores estão em completo acordo. O item ii) do Lema II-4 pode ser demonstrado utilizando-se a mesma idéia proposta em Geromel-Baptistella [10]. Para $s^* \neq 0$,

$$\frac{1}{2} \|\nabla f(x) - s^*\|^2 \leq \frac{1}{2} \|\nabla f(x)\|^2 ,\tag{II.45}$$

uma desigualdade verdadeira visto que $s^* = 0$ é também um solução factível de (II.3). Colocando em seguida, as normas de (II.45) em termos de produtos escalares, encontra-se facilmente que

$$\nabla f(x)' s^* \geq \frac{1}{2} \|s^*\|^2 > 0 ,\tag{II.46}$$

e, portanto, s^* é uma direção usável de (II.1).

Um critério bastante utilizado para detetar-se este tipo de condição de otimalidade ($s^* = 0$) consiste em fazer com que a norma direção de pesquisa s fique abaixo de um valor arbitrariamente pequeno, ϵ_1 .

Entretanto, este critério não leva em consideração o valor relativo da norma do vetor gradiente da função objetivo, o que em geral acaba

provocando um número de iterações maior do que aquele efetivamente necessária. Neste trabalho, propõe-se o seguinte critério:

$$\frac{\|s\|}{1 + \|\nabla f(x)\|} < \epsilon_1, \quad (\text{II.47})$$

e sua interpretação é bastante simples. Para soluções factíveis x em que a norma do vetor gradiente é muito pequena, a otimalidade será detetada em função, principalmente, da norma da direção de pesquisa s . A medida em que a norma do vetor gradiente cresça, a otimalidade será função do valor relativo entre a norma da direção de pesquisa e a norma do vetor gradiente.

O procedimento computacional do método pode, então, ser resumido como segue. Dados A , b , \underline{x} , \bar{x} e x^0 factível inicial, encontre $P = I_n - A'(AA')^{-1}A$ e faça $k = 0$. Uma iteração genérica k do algoritmo será dada por:

PASSO 1 : Encontre $\nabla f(x^k)$ e $B(x^k)$.

PASSO 2 : Com P , $\nabla f(x^k)$ e $B(x^k)$ encontre μ^* resolvendo o problema dual (II.24). Em seguida, obtenha

$$s^k = P[\nabla f(x^k) - B(x^k)'\mu^*] \quad (\text{II.48})$$

PASSO 3 : Se $\frac{\|s^k\|}{1 + \|\nabla f(x^k)\|} < \epsilon_1$, pare: x^k satisfaz as condições necessárias

de otimalidade de Kuhn-Tucker. Senão, vá para o PASSO 4.

PASSO 4 : Encontre o passo α_k do algoritmo resolvendo

$$\text{Min}_{\alpha} f(\bar{x}^k - \alpha s^k) \quad (II.49)$$

$$0 \leq \alpha \leq \min\{\bar{\alpha}_1, \bar{\alpha}_2\}$$

onde

$$\bar{\alpha}_1 = \min_{1 \leq i \leq n} \{(x_i^k - \underline{x}_i) / s_i^k, s_i^k > 0\} \quad (II.50)$$

$$\bar{\alpha}_2 = \min_{1 \leq i \leq n} \{(x_i^k - \bar{x}_i) / s_i^k, s_i^k < 0\}$$

Atualize as variáveis primais,

$$x^{k+1} = x^k - \alpha_k s^k, \quad (II.51)$$

faça $k = k + 1$ e retorne ao PASSO 1.

Apresentado o algoritmo, é importante agora analisar de que forma e qual a ordem de grandeza que um erro de $\pm \Delta\mu$ na solução do sub-problema dual provoca no cálculo da projeção. Para um erro de $\pm \Delta\mu$ ter-se-ia,

$$\begin{aligned} s \pm \Delta s &= P[\nabla f(x) - B(x)'(\mu^* \pm \Delta\mu)] \\ &= P[\nabla f(x) - B(x)'\mu^*] \mp PB(x)'\Delta\mu. \end{aligned} \quad (II.52)$$

Colocado em termos de sua norma, o erro introduzido será dado por

$$\|\Delta s\| \leq \|P\| \|B(x)'\Delta\mu\| = \|P\| \|\Delta\mu\|, \quad (II.53)$$

e como

$$\|P\| = \max_{x \neq 0} \sqrt{\frac{x'P'Px}{x'x}} = \lambda_M(P) = 1, \quad (\text{II.54})$$

conclui-se então que

$$\|\Delta s\| \leq \|\Delta \mu\|, \quad (\text{II.55})$$

ou seja, o erro cometido no cálculo da projeção está limitado superiormente pelo erro cometido na solução do subproblema dual. Isto sugere que os critérios ϵ_1 para o problema primal e ϵ_2 para o problema dual devam ter ordens de grandeza distintas. Computacionalmente é necessário impor que os critérios ϵ_1 e ϵ_2 sejam tais que

$$\epsilon_2 < \epsilon_1, \quad (\text{II.56})$$

II.5. Exemplos

Com o objetivo de demonstrar a eficiência do método, bem como validar o programa computacional desenvolvido para implementá-lo, foram realizados inúmeros testes, utilizando-se diversos tipos de problemas. Em seguida serão apresentados dois destes exemplos.

EXEMPLO Nº 1 : Seleção de Atividades de um Agricultor em Presença de Risco

Um agricultor dispõe de uma certa quantidade de terras e deseja

saber como selecionar o conjunto de atividades que deverão ser desenvolvidas para que o risco de seu empreendimento seja minimizado em relação a uma renda esperada R. Silveira [20] apresenta uma análise detalhada sobre o modelamento e proposição deste problema.

É possível formulá-lo como um problema de programação quadrática:

$$\begin{aligned} \text{Min } x'Qx \\ Ax \leq b \\ x \geq 0 \end{aligned} \tag{II.57}$$

onde $x \in R^5$ e suas componentes x_1, x_2, x_3, x_4 e x_5 são as áreas em hectare onde deverão ser plantados algodão, arroz, feijão, milho e mandioca, respectivamente, $Q \in R^{5 \times 5}$ é a matriz de covariância para estas culturas, $A \in R^{4 \times 5}$ é a matriz tecnológica do agricultor e $b \in R^4$ é formado pela renda esperada R e os recursos disponíveis para o plantio.

Utilizando-se os dados fornecidos por Silveira [20]*,

$$Q = \begin{bmatrix} 2,3939 & 4,0666 & 2,3431 & 1,8039 & 1,4329 \\ 4,0666 & 9,5703 & 4,3505 & 2,4916 & 2,7912 \\ 2,3431 & 4,3505 & 2,7333 & 2,0979 & 1,9803 \\ 1,8039 & 2,4916 & 2,0979 & 2,0617 & 1,4827 \\ 1,4329 & 2,7912 & 1,9803 & 1,4827 & 1,6692 \end{bmatrix} \tag{II.58}$$

(*) Para efeito de melhor condicionamento numérico, a matriz de covariância Q foi dividida por 10^6 e a primeira linha de A por 10^3 .

$$A = \begin{bmatrix} 2,8774 & 4,0706 & 3,5436 & 2,0518 & 7,6398 \\ 0,0 & 1,0 & 1,0 & 0,0 & 0,0 \\ 1,0 & 0,0 & 0,0 & 1,0 & 1,0 \\ 4,65 & 21,47 & 8,79 & 9,13 & 10,81 \end{bmatrix} \quad (II.59)$$

$$b' = [R/1000 \quad 1,86 \quad 2,75 \quad 300,0] \quad (II.60)$$

Com estes dados e através do método proposto, gerou-se a tabela a seguir, que associa a uma certa renda esperada o valor do risco que o agricultor terá que enfrentar ao realizar seu empreendimento. As precisões utilizadas foram $\epsilon_1 = 10^{-2}$ (primal) e $\epsilon_2 = 10^{-3}$ (dual).

Tabela II.1 EXEMPLO Nº 1 - Resultados Obtidos

RENDA (Cr\$)	k	r	ℓ	CPU(s)	RISCO (Cr\$)
2500	3	4	7	1,73	423,08
5000	3	4	8	1,78	845,81
7500	3	4	8	1,67	1268,70
10000	13	3	4	2,58	1694,60
12500	11	3	4	2,70	2117,60
15000	14	3	4	2,68	2538,70
17500	7	3	5	1,97	2962,10
20000	7	3	5	2,06	3386,00

Fonte: Calculada - Sistema PDP-10/UNICAMP

k = nº total de iterações do problema primal

r = nº máximo de restrições de desigualdade ativas

ℓ = nº médio de iterações do problema dual por iteração do primal.

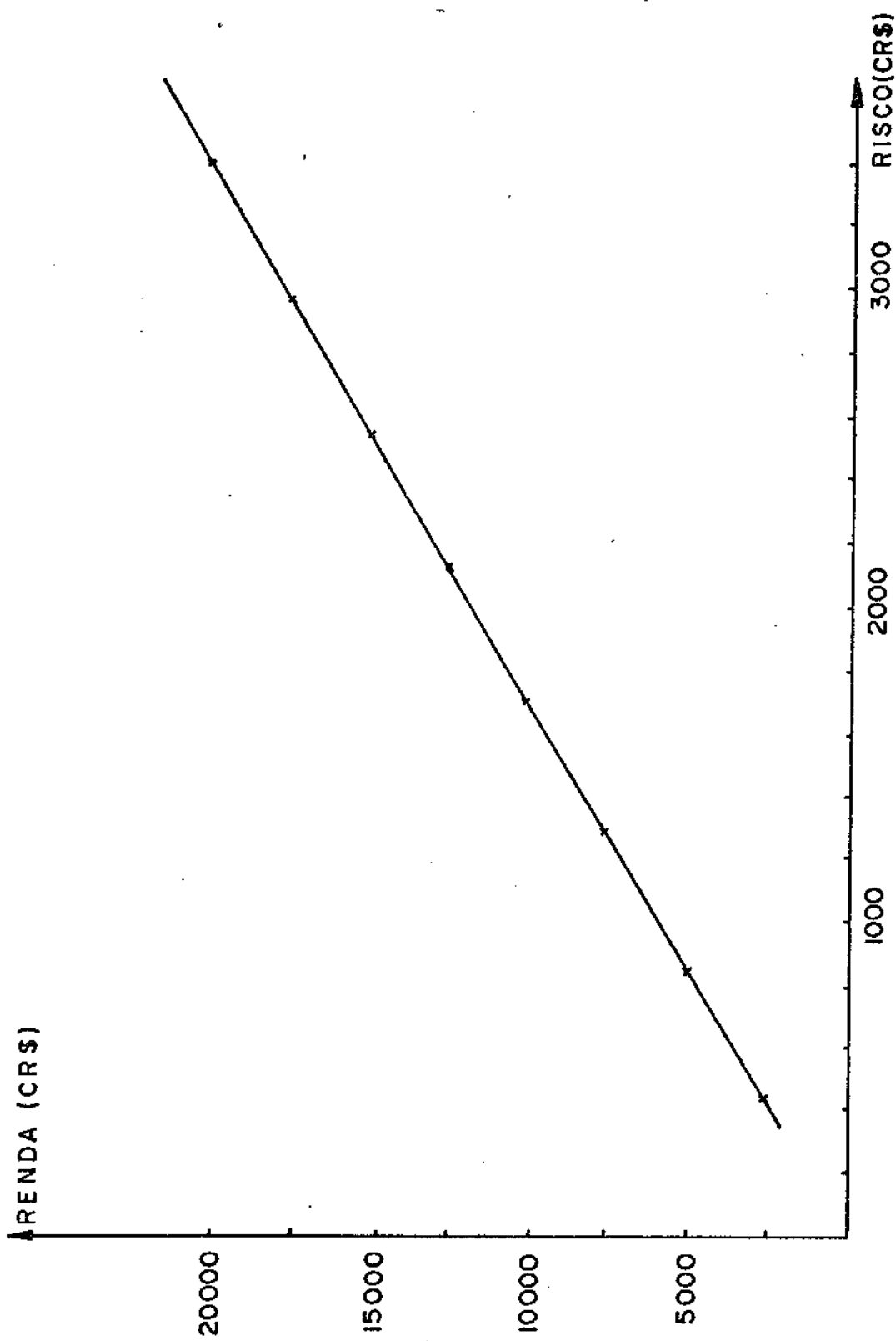


Fig. II.1 - Fronteira de Eficiência Risco-Renda

Da tabela II.1 é construída a chamada Fronteira de Eficiência Risco-Renda. Esta curva está ilustrada na Fig. II.1 e é linear na faixa de renda considerada.

EXEMPLO Nº 2 : Geração Hidroelétrica de Energia

Este problema consiste em determinar uma política ótima de operação para uma usina hidroelétrica de modo a maximizar sua produção de energia ao longo de 6 meses de operação com discretização quinzenal. O Problema da Geração Hidroelétrica de Energia pode ser colocado de forma simplificada como:

$$\begin{aligned} \text{Max}_{u(t), x(t)} \quad & \sum_{t=0}^{11} \phi(x(t), u(t)) \\ x(t+1) = & x(t) - u(t) + y(t) \quad ; \quad x(0) = x_0 \\ \underline{x} \leq & x(t) \leq \bar{x} \\ \underline{u} \leq & u(t) \leq \bar{u} \end{aligned} \tag{II.61}$$

onde

$x(t)$ = volume armazenado no reservatório no instante t

$u(t)$ = volume de água turbinada no instante t

$y(t)$ = afluições no instante t

Embora existam técnicas bem mais eficientes para resolver este problema de otimização, na verdade um problema de controle ótimo discreto com tempo final fixo, este problema será abordado como sendo de otimização

estática. Para tanto, utiliza-se um conhecido resultado que basicamente estabelece que qualquer problema de otimização dinâmica pode ser colocado como um problema de otimização estática (Canon-Cullurn-Polak [6]). O problema (II.61) pode então ser reformulado como

$$\text{Max}_{x(t), u(t)} \sum_{t=0}^{11} \phi(x(t), u(t))$$

$$\begin{aligned} x(1) + u(0) &= y(0) + x_0 \\ -x(1) + x(2) + u(1) &= y(1) \\ -x(2) + x(3) + u(2) &= y(2) \\ &\vdots \\ -x(11) + x(12) + u(11) &= y(11) \end{aligned} \quad (\text{II.62})$$

$$\underline{x} \leq x(t) \leq \bar{x}$$

$$\underline{u} \leq u(t) \leq \bar{u}$$

A restrição dinâmica foi eliminada, surgindo no seu lugar uma matriz $A \in \mathbb{R}^{12 \times 24}$ cuja estrutura é conhecida na literatura como sendo do tipo "staircase". Convencionando-se em seguida

$$z' = [x(1) \ u(0) \ x(2) \ u(1) \ \dots \ x(12) \ u(11)] \quad (\text{II.63})$$

o problema resultante será

$$\begin{aligned} \text{Max}_z f(z) &= \sum_{i=1}^{23} \phi(z_i, z_{i+1}) \\ Az &= b \\ \underline{z} &\leq z \leq \bar{z} \end{aligned} \quad (\text{II.64})$$

A função de geração $\phi(\cdot)$ é não-côncava e dada por

$$\phi(z_i, z_{i+1}) = \pi z_{i+1} (1 - e^{-\rho z_i}) \quad (\text{II.65})$$

Os valores constantes, obtidos de Geromel-Baptistella [10] são os seguintes:

$$\begin{aligned} \pi &= 1,0 ; \\ \rho &= 0,020 ; \\ \underline{z}' &= [8 \ 0 \ 8 \ 0 \ \dots \ 8 \ 0] \\ \bar{z}' &= [80 \ 20 \ 80 \ 20 \ \dots \ 80 \ 20] \end{aligned} \quad (\text{II.66})$$

As afluências a cada instante de tempo estão indicadas na Tabela II.1, a seguir,

Tabela II.2 EXEMPLO Nº 2 - Afluências

t	0	1	2	3	4	5	6	7	8	9	10	11
y(t)	5	10	10	10	20	25	5	5	5	5	5	5

Fonte: Geromel-Baptistella [10]

Na Tabela II.3 estão reproduzidos os resultados obtidos através do método proposto, utilizando-se precisões de $\epsilon_1 = 10^{-2}$ (primal) e $\epsilon_2 = 10^{-3}$ (dual).

Tabela II.3 EXEMPLO Nº 2 - Resultados Obtidos

ITERAÇÃO PRIMAL	RESTRIÇÕES ATIVAS	ITERAÇÕES DUAL	FUNÇÃO OBJETIVO
1	7	9	-31,8966
2	4	3	-45,1600
3	5	4	-52,7234
4	6	4	-54,3875
5	6	5	-55,6816
6	9	27	-65,1516
7	9	5	-65,3564
8	10	27	-65,5989
9	10	6	-68,1513
10	11	11	-68,4259

Fonte: Calculada - Sistema PDP-10/UNICAMP

É interessante notar que na maioria das iterações necessárias para alcançar-se a otimalidade, o número de iterações do algoritmo para o subproblema dual foi inferior à ordem das funções quadráticas. Isto foi possível devido a inicialização da variável dual μ através da solução irrestrita do problema dual obtida com a técnica de direções conjugadas de Fletcher-Reeves. Esta solução irrestrita coincidiu na maioria dos casos com a própria solução ótima restrita do problema dual. O tempo de CPU foi de aproximadamente 11,26 seg.

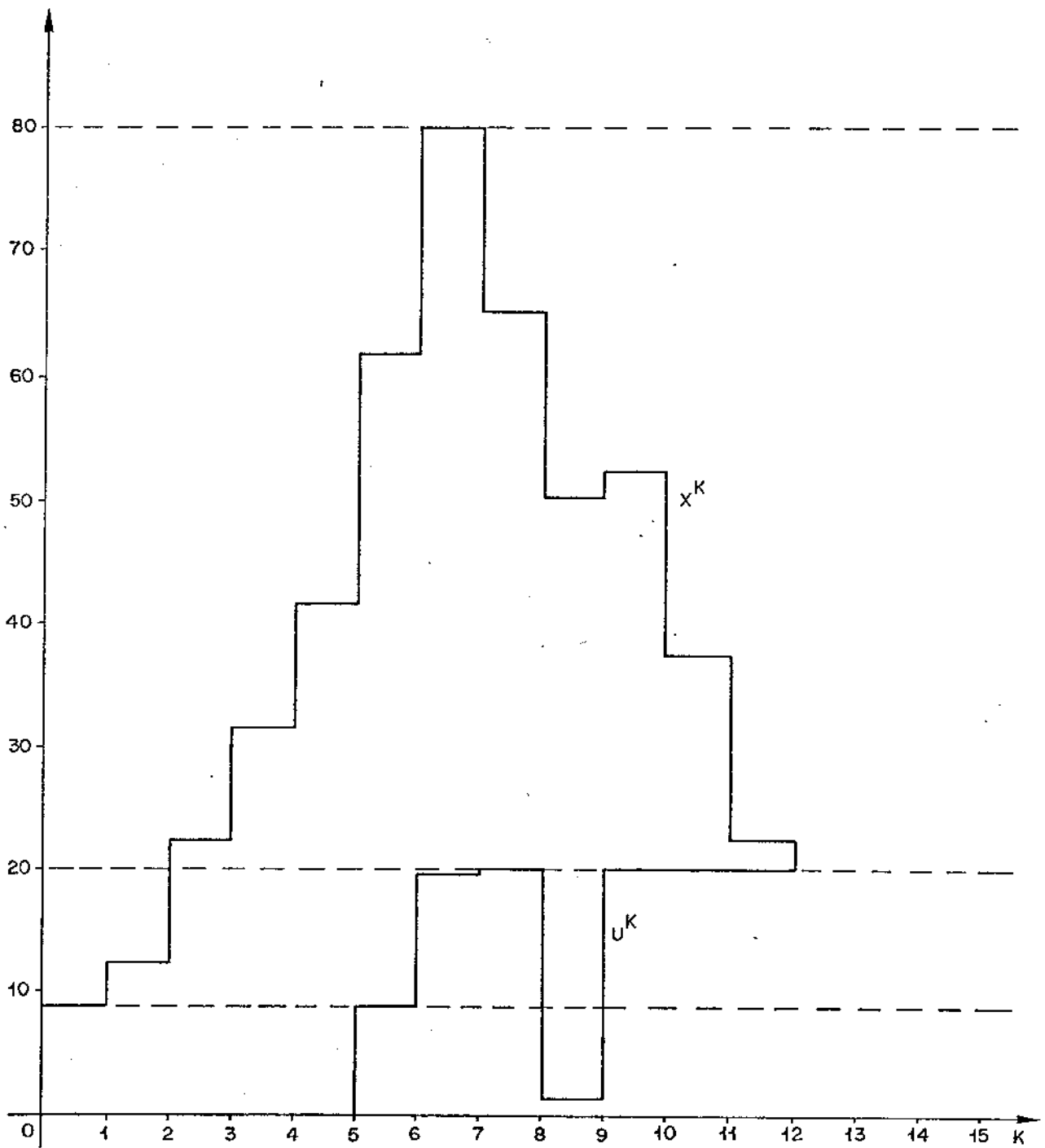


Fig. II.2 - Trajetórias de Estado e Controle para o Exemplo 2

A Fig. II.2 ilustra as trajetórias de estado e controle da usina hidroelétrica considerada.

No Capítulo IV um Problema de Geração Hidroelétrica de Energia bem mais complexo será formulado e resolvido através da extensão ao caso dinâmico do método primal proposto neste capítulo.

II.6. Conclusão

Analisando o caráter puramente estático do método primal proposto, algumas importantes conclusões podem ser obtidas.

O método apresentado tem como sua principal característica a forma diferenciada com a qual manipula os conjuntos de restrições de igualdade e desigualdade. A Teoria da Dualidade quando aplicada ao problema de projeção, permite que a contribuição destes dois conjuntos de restrições seja tomada de forma independente. Assim, formulado o problema primal, a contribuição das restrições de igualdade à direção de pesquisa do método é imediatamente conhecida através da matriz de projeção P . Em seguida, o vetor gradiente da função objetivo é modificado pela variável dual μ^* tendo por base a informação contida na matriz $B(x)$ e são então este gradiente modificado é projetado sobre as restrições de igualdade.

Desaparecem com este procedimento os problemas de degenerescência provocados pela combinação linear de restrições ativas e que constituem um obstáculo teórico à obtenção do gradiente reduzido pelo método de Wolfe e à inversão da matriz de projeção do método de Rosen. O método proposto substitui esta inversão pela necessidade de resolver-se um subproblema de otimização para o cálculo da variável dual μ^* , cuja dimensão é igual ao número de

restrições de desigualdade ativas, em geral, muito menor que o número de variáveis do problema e de fácil solução, como ficou demonstrado através de exemplos.

Em relação ao método de Geromel-Baptistella, o método proposto fornece uma maneira global de determinar a direção de pesquisa, mais eficiente do ponto de vista computacional. Como este último, o método proposto fornece ainda um modo simples de detectar a otimalidade.

Embora este aspecto não tenha sido estudado, o método primal proposto deve exibir, como o método de Rosen, uma taxa de convergência, linear uma vez que estes dois métodos são completamente iguais se não existirem restrições de desigualdade associadas às variáveis do problema.

CAPÍTULO III

III.1. Introdução

O Exemplo Nº 2 do capítulo anterior exemplificou a relação existente entre problemas de programação matemática e problemas de controle ótimo de sistemas dinâmicos discretos. O problema de controle ótimo originalmente formulado foi reescrito como um problema de otimização estática e então resolvido, de maneira ineficiente do ponto de vista de controle, via programação matemática. A proposta da moderna teoria de controle ótimo é empregar a teoria de otimização na resolução do problema de controle e ao mesmo tempo explorar a estrutura particular das restrições - a dinâmica do sistema.

Resolver um problema de controle ótimo significa encontrar uma lei de controle $u^*(t)$, $t \in [0, T-1]$ tal que o desempenho do sistema é otimizado segundo um critério pré-estabelecido e, em geral, separável no tempo, satisfeita a sua dinâmica e o espaço das funções U que define as trajetórias de controle factíveis.

Entretanto, um modelamento físico realista quase sempre introduz restrições também sobre alguns ou mesmo todos os estados do sistema, hipóteses sob as quais o Princípio do Mínimo de Pontryagin, um dos mais importantes resultados disponíveis em teoria de controle ótimo não pode ser diretamente aplicado, pois restrições desta natureza conduzem ao acoplamento temporal da lei de controle (Athans [2], Sage [17]).

Como principal alternativa para o tratamento de problemas desta classe, surgem então os métodos primais. Este capítulo estende a metodologia proposta no capítulo anterior, a problemas de otimização com dinâmica linear e restrições tanto nas variáveis de estado, quanto nas variáveis de controle. Somente o caso discreto, melhor adaptado ao uso de computadores digitais será considerado.

III.2. Desenvolvimento Teórico do Método: Caso Dinâmico

Considere o seguinte problema de controle ótimo discreto:

$$\text{Min}_{x(t), u(t)} \sum_{t=0}^{T-1} \psi(x(t), u(t)) + \psi_T(x(T)) = J(x(t), u(t))$$

$$x(t+1) = Ax(t) + Bu(t) + y(t); \quad x(0) = x_0$$

$$x(t) \in X = \{x(t), t \in [0, T] \mid \underline{x} \leq x(t) \leq \bar{x}\} \quad (\text{III.1})$$

$$u(t) \in U = \{u(t), t \in [0, T-1] \mid \underline{u} \leq u(t) \leq \bar{u}\},$$

onde o instante final T é fixo e $x(T)$ livre. Por hipótese $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ e $y(t) \in \mathbb{R}^n$ representa uma entrada conhecida ao sistema. As funções $\psi(x(t), u(t)) : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ e $\psi_T(x(T)) : \mathbb{R}^n \rightarrow \mathbb{R}$ são assumidas de classe C^1 em relação aos seus argumentos. Assume-se ainda que a equação de diferenças que descreve a dinâmica do sistema tem solução única $x(t+1) \in X$ para quaisquer $u(t) \in U$ e x_0 dados.

Com esta última hipótese, para uma dada lei de controle admissível^(*) $u(t)$, $t \in [0, T-1]$, o valor do funcional em (III.1) fica completamente determinado, ou seja, (III.1) é na verdade equivalente a

$$\text{Min } J(u(t)) \quad (\text{III.2})$$

$$u(t) \in U / x(t+1) \in X, \quad t \in [0, T-1]$$

onde $J(\cdot) : \mathbb{R}^m \times [0, T-1] \rightarrow \mathbb{R}$.

(*) Por admissível deve-se entender toda lei de controle factível ($u(t) \in U$) para a qual a correspondente trajetória de estado é também factível ($x(t+1) \in X$).

Interpretado desta forma, o problema de controle ótimo (III.1) pode então ser resolvido através de métodos primais variando-se a lei de controle segundo a projeção do gradiente de $J(u(t))$ sobre as suas restrições ativas. Note que o vetor

$$\frac{dJ(u(t))}{du(t)}, \quad t \in [0, T-1] \quad (\text{III.3})$$

representa um gradiente reduzido em relação ao espaço das variáveis do problema. O Teorema a seguir apresenta um resultado bastante conhecido em teoria de controle ótimo e através do qual é possível obter-se este gradiente reduzido de forma eficiente.

Teorema III-1 : Seja a função Hamiltoniano $H(x(t), u(t), p(t)) : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R}$ associada a (III.1),

$$H(x(t), u(t), p(t)) \triangleq \psi(x(t), u(t)) + p(t)' [Ax(t) + Bu(t) + y(t)], \quad (\text{III.4})$$

onde $p(t) \in \mathbb{R}^n$, $t \in [0, T-1]$ representa o vetor de multiplicadores de Lagrange ou co-estado associado a restrição dinâmica. Então, o gradiente reduzido será dado por

$$\frac{dJ(u(t))}{du(t)} = \frac{\partial H(x(t), u(t), p(t))}{\partial u(t)} \quad (\text{III.5})$$

onde os vetores $x(t)$, $t \in [1, T]$ e $p(t)$, $t \in [0, T-1]$ satisfazem,

$$\frac{\partial H(x(t), u(t), p(t))}{\partial x(t)} = p(t-1); \quad p(T-1) = \frac{d\psi_T(x(T))}{dx(T)} \quad (\text{III.6.1})$$

$$\frac{\partial H(x(t), u(t), p(t))}{\partial p(t)} = x(t+1); \quad x(0) = x_0 \quad (\text{III.6.2})$$

Prova (Geromel [8]) : Considere $u(t) \in [0, T-1]$ uma trajetória admissível de controle e $u(t) - \epsilon r(t)$ uma trajetória também admissível, mas perturbada em torno de $u(t)$. Seja ainda $x(t) - \epsilon g(t)$, $t \in [1, T]$ a correspondente trajetória perturbada do sistema. Em primeira ordem, o funcional de (III.1) escrito em torno das trajetórias nominais fica sendo^(*) :

$$J(u(t) - \epsilon r(t)) = J(u(t)) + \sum_{t=0}^{T-1} \epsilon \frac{\partial \psi' g(t)}{\partial x(t)} + \epsilon \frac{\partial \psi' r(t)}{\partial u(t)} + \epsilon \frac{d\psi_T' g(T)}{dx(T)} \quad (\text{III.7})$$

Também em primeira ordem, considere as mesmas variações na equação dinâmica do sistema (III.6.2):

$$g(t+1) = Ag(t) + Br(t); \quad g(0) = 0 \quad (\text{III.8})$$

O restante da prova visará eliminar o vetor $g(t)$. Transpondo ambos os lados de (III.6.1) e pós-multiplicando a equação resultante por $g(t)$, vem

$$\frac{\partial \psi' g(t)}{\partial x(t)} + p(t)' Ag(t) = p(t-1)' g(t), \quad (\text{III.9})$$

com

$$p(T-1)' g(T) = \frac{d\psi_T' g(T)}{dx(T)} \quad (\text{III.10})$$

Em seguida, substituindo-se o vetor $Ag(t)$, obtido através de (III.8) em (III.9), resulta

$$\frac{\partial \psi' g(t)}{\partial x(t)} = p(t-1)' g(t) - p(t)' [g(t+1) - Br(t)], \quad (\text{III.11})$$

(*) Serão omitidos os argumentos das funções envolvidas.

O funcional (III.7) pode agora ser reescrito na seguinte forma:

$$\begin{aligned}
 J(u(t) - \epsilon r(t)) &= J(u(t)) + \sum_{t=0}^{T-1} \epsilon \{p(t-1)'g(t) - p(t)'[g(t+1) - Br(t)]\} \\
 &+ \epsilon \frac{\partial \psi' r(t)}{\partial u(t)} + \epsilon p(T-1)'g(T) \quad (III.12)
 \end{aligned}$$

Note entretanto que,

$$\begin{aligned}
 \sum_{t=0}^{T-1} \{p(t-1)'g(t) - p(t)'g(t+1)\} &= p(-1)'g(0) - p(0)'g(1) + p(0)'g(1) - \\
 &- p(1)'g(2) + \dots + p(T-2)'g(T-1) - \\
 &- p(T-1)'g(T) = -p(T-1)'g(T), \quad (III.13)
 \end{aligned}$$

onde $p(-1) = 0$. Levando então este resultado em (III.12), obtêm-se

$$\begin{aligned}
 J(u(t) - \epsilon r(t)) &= J(u(t)) + \sum_{t=0}^{T-1} \epsilon \left\{ \frac{\partial \psi' r(t)}{\partial u(t)} + p(t)'Br(t) \right\} \\
 &= J(u(t)) + \sum_{t=0}^{T-1} \epsilon \left[\frac{\partial \psi}{\partial u(t)} + B'p(t) \right]' r(t) \\
 &= J(u(t)) + \sum_{t=0}^{T-1} \epsilon \frac{\partial H(x(t), u(t), p(t))' r(t)}{\partial u(t)}, \quad (III.14)
 \end{aligned}$$

Finalmente, reescrevendo (III.14) como

$$\frac{J(u(t) - \epsilon r(t)) - J(u(t))}{\epsilon} = \sum_{t=0}^{T-1} \frac{\partial H(x(t), u(t), p(t))' r(t)}{\partial u(t)}, \quad (III.15)$$

tomando o limite de (III.15) para $\epsilon \rightarrow 0^+$, e lembrando que o resultado deve valer para toda direção $r(t)$, $t \in [0, T-1]$, verifica-se a igualdade,

$$\frac{dJ(u(t))}{du(t)} = \frac{\partial H(x(t), u(t), p(t))}{\partial u(t)} \quad (\text{III.16})$$

Calculado o gradiente reduzido, um problema de projeção para o caso dinâmico, análogo ao problema de projeção para o caso estático, pode então ser formulado. Dada uma trajetória admissível de controle $u(t)$, $t \in [0, T-1]$ e a correspondente trajetória do sistema $x(t+1)$, $t \in [0, T-1]$, o problema de projeção consistirá em minimizar-se a cada instante de tempo o ângulo formado entre o vetor gradiente reduzido e o sub-espço obtido a partir das restrições ativas de (III.1):

$$\text{Min } \frac{1}{2} \sum_{t=0}^{T-1} \|r(t) - \frac{\partial H}{\partial u(t)}\|^2$$

$$g(t+1) = Ag(t) + Br(t); \quad g(0) = 0$$

$$D(x(t+1))g(t+1) \leq 0 \quad (\text{III.17})$$

$$D(u(t))r(t) \leq 0, \quad t \in [0, T-1].$$

As matrizes $D(x(t+1))$ e $D(u(t))$ são construídas seguindo-se o mesmo procedimento descrito no Capítulo II para a obtenção da matriz $B(x)$. Sejam q_1 e q_2 o número de variáveis de estado e controle ativas, respectivamente. Então, $D(x(t+1)) \in R^{q_1 \times n}$ e $D(u(t)) \in R^{q_2 \times m}$.

O que segue são manipulações em (III.17) com o objetivo de colocar o problema de projeção numa forma mais compacta. Assim, para a condição inicial dada, a restrição dinâmica de (III.17), fica sendo,

$$D(x)g \leq 0 \quad (\text{III.22})$$

$$D(u)r \leq 0 \quad (\text{III.23})$$

onde agora $D(x) \in R^{q_1 \times nT}$ e $D(u) \in R^{q_2 \times mT}$. É ainda possível eliminar o vetor g substituindo-se (III.20) em (III.22):

$$D(x)Gr \leq 0 \quad (\text{III.24})$$

Por último, lembrando que

$$\|r - \frac{\partial H}{\partial u}\|^2 = \sum_{t=0}^{T-1} \|r(t) - \frac{\partial H}{\partial u(t)}\|^2, \quad (\text{III.25})$$

o problema de projeção para o caso dinâmico poderá ser finalmente formulado como

$$\text{Min } \frac{1}{2} \|r - \frac{\partial H}{\partial u}\|^2 \quad (\text{III.26})$$

$$M(x, u)r \leq 0$$

onde $M \in R^{q \times mT}$ ($q = q_1 + q_2$) é dada por

$$M(x, u) \triangleq \begin{bmatrix} D(x)G \\ \dots \\ D(u) \end{bmatrix} \quad (\text{III.27})$$

Formulado o problema de projeção, o mesmo procedimento adotado

no Capítulo II para estudar o caso estático será observado.

Seja então $\mu \in \mathbb{R}^q$ a variável dual associada às restrições de (III.26). O problema dual correspondente será portanto,

$$\text{Max}_{\mu \geq 0} \min_r L(r, \mu), \quad (\text{III.28})$$

onde $L(r, \mu) : \mathbb{R}^{m^T} \times \mathbb{R}^q \rightarrow \mathbb{R}$ é a função Lagrangeano associada a (III.26):

$$L(r, \mu) = \frac{1}{2} \|r - \frac{\partial H}{\partial u}\|^2 + \mu^T M(x, u)r, \quad (\text{III.29})$$

e, como no caso estático, o problema de minimização em r é estritamente convexo. Assim, impondo-se a estacionariedade de (III.29) em relação à variável primal,

$$\left. \frac{\partial L(r, \mu)}{\partial r} \right|_{r=r^*} = 0, \quad (\text{III.30})$$

resulta para a direção de pesquisa r ,

$$r^* = \frac{\partial H}{\partial u} - M(x, u)^T \mu^* \quad (\text{III.31})$$

A função dual é então facilmente encontrada bastando para isto, substituir (III.31) em (III.29):

$$\psi(\mu) = \frac{1}{2} \mu^T Q(x, u)\mu + c(x, u)^T \mu \quad (\text{III.32})$$

onde

$$Q(x, u) \triangleq -M(x, u)M(x, u)' \quad (\text{III.33.1})$$

$$c(x, u) \triangleq M(x, u) \frac{\partial H}{\partial u} \quad (\text{III.33.2})$$

$$Q(x, u) \in R^{q \times q}, \quad c(x, u) \in R^q$$

Se observada a grande semelhança existente entre o problema dual obtido para o caso dinâmico e o correspondente ao caso estático, não constituirá surpresa alguma verificar que todas as propriedades anteriormente demonstradas são também válidas, com poucas modificações, também para o caso dinâmico. Desta forma, o algoritmo desenvolvido para a obtenção da variável dual μ^* poderá ser diretamente empregado.

Lema III-1 : A matriz $Q(x, u)$ é definida negativa no sub-espço das direções usáveis d do problema dual (III.28).

Prova : Por direção usável, tem-se

$$\nabla \psi(\mu)' d < 0, \quad (\text{III.34})$$

e mais uma vez, explicitando-se o vetor gradiente de $\psi(\mu)$, vem

$$[Q(x, u)\mu + c(x, u)]' d < 0, \quad (\text{III.35})$$

equivalente também a

$$\left[\frac{\partial H}{\partial u} - M(x, u)' \mu \right]' M(x, u)' d < 0, \quad (\text{III.36})$$

implicando claramente em $M(x, u)' d \neq 0$. Fazendo agora,

$$d' Q(x, u) d = -d' M(x, u) M(x, u)' d = -\|M(x, u)' d\|^2, \quad (\text{III.37})$$

resulta, como esperado,

$$d' Q(x, u) d < 0, \quad \forall d \text{ usável} \quad (\text{III.38})$$

Tendo sido concluído que é possível estender ao caso dinâmico o algoritmo desenvolvido para o cálculo de μ^* , o passo seguinte será estabelecer a faixa onde se encontram os autovalores de $Q(x, u)$.

Lema III-2 : Seja a matriz $Q(x, u)$, definida por (III.33.1); então,

$$-(1 + \|G\|^2) \leq \lambda_i(Q(x, u)) \leq 0, \quad (\text{III.39})$$

Prova : A matriz simétrica semi-definida positiva $M(x, u)M(x, u)'$ é, em termos de suas sub-matrizes, escrita como:

$$M(x, u)M(x, u)' = \begin{bmatrix} D(x)GG'D(x)' & D(x)GD(u)' \\ \hline D(u)G'D(x) & Iq_2 \end{bmatrix} \quad (\text{III.40})$$

Note que foi utilizada a propriedade

$$D(u)D(u)' = Iq_2 \quad (\text{III.41})$$

Considere agora a seguinte forma de reescrever (III.40):

$$\begin{bmatrix} D(x) & 0 \\ 0 & D(u) \end{bmatrix} \begin{bmatrix} G \\ I_{mT} \end{bmatrix} \begin{bmatrix} G' & I_{mT} \end{bmatrix} \begin{bmatrix} D(x)' & 0 \\ 0 & D(u)' \end{bmatrix} = \\ = E(x, u)FF'E(x, u)'. \quad (\text{III.42})$$

Omitindo-se os argumentos x e u e associando um autovetor $z_i \in R^q$ ao i -ésimo autovalor de (III.42), vem

$$\lambda_i(MM') = \frac{z_i'EFF'E'z_i}{z_i'EE'z_i}, \quad i = 1, \dots, q \quad (\text{III.43})$$

Mais uma vez utilizou-se a propriedade

$$EE' = I_q. \quad (\text{III.44})$$

Em seguida, fazendo $Ez_i = y_i$ resulta para o autovalor máximo de MM' ,

$$\lambda_M(MM') = \max_{y_i \neq 0} \frac{y_i'FF'y_i}{y_i'y_i} = \lambda_M(FF'). \quad (\text{III.45})$$

Usando então a propriedade $\lambda_M(FF') = \lambda_M(F'F)$ obtêm-se,

$$\lambda_M(MM') = \lambda_M(I_{mT} + G'G) = 1 + \lambda_M(G'G) = 1 + \|G\|^2, \quad (\text{III.46})$$

e como $Q = -MM'$, (III.39) fica, portanto, demonstrada

Como uma importante particularização do problema, considere que em dada iteração do algoritmo não existem restrições de estado ativas. Neste caso, existindo possivelmente apenas restrições de controle ativas,

$$M(x, u) = D(u) \quad (\text{III.47})$$

e, conseqüentemente,

$$Q(x, u) = -M(x, u)M(x, u)' = -D(u)D(u)' = -Iq_2 \quad (\text{III.48})$$

As implicações de (III.18) são claras. Com um único passo de gradiente é possível obter a solução ótima do problema dual (III.28), acarretando considerável simplificação do método. Uma inevitável pergunta então se colocaria : qual seria então a vantagem do método proposto em relação, por exemplo, ao método do Gradiente Reduzido (Wolfe, 1964) se em dada iteração não existem restrições de estado ativas ? Para responder a esta pergunta, algumas características do método do Gradiente Reduzido devem ser apontadas.

Usando a unicidade de solução da equação dinâmica do sistema, o método de Wolfe elege naturalmente as variáveis de controle com variáveis não-básicas e as variáveis de estado como variáveis básicas. Assim, inexistindo restrições sobre as variáveis de estado, o gradiente reduzido pode ser projetado apenas sobre as variáveis de controle e o método é então perfeitamente aplicável.

Existindo, entretanto, restrições sobre as variáveis de estado podem surgir os usuais problemas de degenerescência associados ao método quando muitas destas restrições se ativarem em uma dada iteração. Evidentemente, estes problemas serão tanto mais críticos quanto maior for a dimensão do problema tratado.

A propriedade evidenciada por (III.48) seria então importante no sentido de fazer o procedimento computacional do método proposto tão simples quanto o do método de Wolfe na ausência de restrições sobre os estados do sistema. Na verdade, sob esta condição, os dois métodos são absolutamente

Tabela III.1 - Solução do EXEMPLO Nº 2 através do Algoritmo Proposto neste Capítulo

k	q	ℓ	CRITÉRIO
1	7	4	35.11
2	1	1	42.91
3	2	1	49.13
4	3	1	53.23
5	4	1	56.92
6	5	1	61.84
7	5	1	62.14
8	5	1	64.39
9	6	1	65.03
10	7	1	67.26

FONTE : Calculada - Sistema S-700 Prológica

k = iteração do problema primal

q = nº de restrições de desigualdade ativas

ℓ = nº de iterações do problema dual

equivalentes, podendo esta equivalência ser constatada através da simples observação de como o método dual-clássico trabalha e de como a direção de pesquisa do método proposto é determinada na ausência de restrições sobre variáveis de estado.

O método proposto permite ainda a inclusão de restrições desta natureza sem incorrer em quaisquer hipóteses além da unicidade de solução da equação dinâmica. Neste caso, deverá ser introduzido um esforço de cálculo adicional, necessário para obter-se a projeção do gradiente reduzido.

Para ilustrar a importância da propriedade demonstrada em (III.48), considere o Exemplo Nº 2 resolvido no Capítulo II através do algoritmo para o caso estático, agora novamente resolvido com a metodologia apresentada neste capítulo (todos os parâmetros e condições iniciais são os mesmos).

Na Tabela III-1 dois fatos são particularmente importantes. Em primeiro lugar, a óbvia presença da propriedade já mencionada em quase todas as iterações do algoritmo. O fato de existirem apenas restrições de controle ativas em 9 das 10 iterações do algoritmo, pode ser parcialmente explicado, lembrando-se que a variável de controle está muito mais restrita pelos seus limitantes do que a correspondente variável de estado o que, neste exemplo, tem inclusive uma justificativa física. O segundo fato importante é o melhor aproveitamento da restrição dinâmica tornando o algoritmo bem mais eficiente. A diferença entre os valores ótimos do critério fica por conta das precisões utilizadas para resolver o exemplo no Micro-Computador CP-700 da Prologica ($\epsilon_1 = 0,1$ e $\epsilon_2 = 0,01$).

O algoritmo utilizado será agora resumido. Dada uma lei de controle admissível $u(t) \in U$, $t \in [0, T-1]$ encontre G e faça $k = 0$. Uma iteração genérica k do algoritmo será como segue.

PASSO 1 : Encontre o gradiente reduzido

$$\frac{dJ}{du(t)^k} = \frac{\partial H}{\partial u(t)^k}$$

com $x(t)^k$ e $p(t)^k$ satisfazendo

$$\frac{\partial H}{\partial x(t)^k} = p(t-1)^k ; \quad p(T-1)^k = \frac{d\psi_T(x(T)^k)}{dx(T)^k}$$

$$\frac{\partial H}{\partial p(t)^k} = x(t+1)^k ; \quad x(0) = x_0$$

PASSO 2 : Determine $D(x^k)$ e $D(u^k)$. Encontre ainda

$$M(x^k, u^k) = \begin{bmatrix} D(x^k)G \\ \hline D(u^k) \end{bmatrix}$$

PASSO 3 : Com $M(x^k, u^k)$ e $\frac{\partial H}{\partial u^k}$ resolva o problema dual obtendo μ^* . Em seguida, determine

$$r^k = \frac{\partial H}{\partial u^k} - M(x^k, u^k)' \mu^*$$

PASSO 4 : Se $\|r^k\| < \epsilon_1$, arbitrariamente pequeno, pare. Os valores correntes de $u(t)$, $t \in [0, T-1]$ e $x(t)$, $t \in [1, T]$ satisfazem as condições

necessárias de otimalidade de Kuhn-Tucker. Senão, vá para o PASSO 5

PASSO 5 : Encontre um passo α_k para o algoritmo tal que

$$J(u(t)^k - \alpha_k r(t)^k) < J(u(t)^k),$$

com $u(t)^{k+1} = u(t)^k - \alpha_k r(t)^k \in U$ e $x(t+1)^{k+1} \in X$, para $t \in [0, T-1]$. Faça $k = k+1$ e volte ao PASSO 1.

Neste ponto é importante ressaltar que as matrizes $D(x)$, $D(u)$ e $M(x, u)$ aparecem explicitamente durante todo o desenvolvimento do método proposto para efeito ilustrativo. Na prática, a matriz $Q(x, u)$ e o vetor $c(x, u)$ são obtidos diretamente a partir de vetores que armazenam compactamente a informação contida nas matrizes $D(x)$ e $D(u)$ e sem que seja necessário armazenar a matriz $M(x, u)$.

A próxima seção analisará alguns aspectos relacionados à matriz de projeção G .

III.3. A Matriz de Projeção G

Em primeiro lugar, (III.21) é simplesmente a forma matricial encontrada para representar a restrição dinâmica do problema de projeção e assim relacionar a direção de pesquisa das variáveis de controle com a direção de pesquisa das variáveis de estado do problema.

Na verdade, esta matriz tem uma função inteiramente análoga à matriz P do caso estático. Obtida a direção de pesquisa associada às variáveis de controle, que corresponde à projeção do gradiente reduzido sobre as restrições de factibilidade de estado e controle modificada convenientemente pela variável dual μ^* , a matriz G se encarregará de projetar este gradiente reduzido modificado sobre a dinâmica do sistema e assim permitir que a direção associada às variáveis de estado seja encontrada.

Esta última direção será importante apenas quando da determinação do passo do algoritmo.

Como no caso estático, a matriz G é conhecida a-priori, isto é, dados o sistema a ser controlado (A e B) e o tempo final T , G fica completamente caracterizada. Entretanto, o fato de sua dimensão depender explicitamente do tempo final especificado, torna inviável a tarefa de armazená-la na forma em que é definida.

Neste caso, a única saída possível é explorar sua estrutura especial em termos computacionais, tentando usar o fato de que o conhecimento de A e B é suficiente para que qualquer elemento de G seja determinado.

Em consequência, a eficiência em manipular a matriz de projeção está diretamente relacionada com o sistema de controle ao qual o algoritmo é aplicado. Em outras palavras, sistemas de controle que apresentem algum tipo de estrutura especial possibilitam a manipulação de G de forma mais eficiente.

A aplicação numérica do método a um sistema dinâmico real no capítulo seguinte contribuirá para tornar mais claros estes pontos.

III.4. Conclusão

O presente capítulo foi basicamente uma extensão do método pri-

mal proposto no capítulo anterior a problemas de controle ótimo discreto com horizonte de tempo finito, dentro de toda a generalidade que o método admite: critério diferenciável, dinâmica linear e restrições de canalização nas variáveis de estado e controle.

Como esperado, a grande maioria dos resultados obtidos no capítulo anterior são também aplicáveis ao caso dinâmico com poucas modificações. A unicidade de solução da equação dinâmica é uma hipótese equivalente a independência linear das restrições no caso estático e o algoritmo para a obtenção da variável dual μ^* é diretamente aplicável no caso dinâmico.

Novas e importantes propriedades do método também foram demonstradas. Em particular, foi possível mostrar que o algoritmo para o subproblema dual convergirá em apenas uma iteração quando não existir qualquer variável de estado ativa. O procedimento computacional do método é neste caso tão simples quanto o do método do Gradiente Reduzido, com a vantagem de poder considerar restrições sobre as variáveis de estado sem fazer qualquer hipótese adicional.

Por outro lado, a inclusão de restrições sobre as variáveis de estado deverá implicar num aumento de esforço computacional necessário à projeção do gradiente reduzido e, por conseguinte, da direção de pesquisa do método.

CAPITULO IV

IV.1. Introdução

O objetivo deste capítulo é apresentar uma aplicação numérica do método primal, proposto a um problema de controle ótimo real.

A opção pelo Problema da Geração Hidroelétrica de Energia é motivada em primeiro lugar pela importância do parque hidráulico no contexto de um problema consideravelmente mais geral, o chamado Problema da Geração Hidrotérmica de Energia. Além de sua importância intrínseca, que decorre do baixo custo marginal da energia hidroelétrica quando comparado ao custo proporcionado pela operação de uma usina térmica (Soares [21], Ohishi [15]), este problema em particular permite um elevado nível de especialização do método proposto.

Em termos genéricos, a estrutura ou topologia de um sistema hidroelétrico, composto por n usinas localizadas em uma dada bacia hidrográfica, caracteriza-se pela distribuição espacial destas usinas numa configuração do tipo árvore. Esta estrutura faz com que a dinâmica do reservatório da i -ésima usina, basicamente uma equação de balanço de massa, seja descrita pela expressão

$$x_i(t+1) = x_i(t) - u_i(t) + \sum_{j \in J_i} u_j(t) + y_i(t) ; x_i(0) = x_{0i}, t \in [0, T-1] \quad (IV.1)$$

indicando que o volume de água armazenado no reservatório (estado) de uma usina em determinado instante de tempo é função apenas do volume do reservatório no instante anterior, do volume de água turbinada (controle) pela usina e da somatória dos volumes turbinados pelas usinas imediatamente a montante da i -ésima, representadas pelo conjunto J_i , e de afluências (entrada) à usina cuja natureza é estocástica mas que para efeito de aplicação serão aqui representadas por seus valores médios. Como consequência imediata deste

fato, a dinâmica do sistema hidroelétrico será regida por uma equação linear a diferenças de ordem n .

$$x(t+1) = Ax(t) + Bu(t) + y(t); \quad x(0) = x_0, \quad t \in [0, T-1] \quad (\text{IV.2})$$

em que

1. A é a matriz identidade de ordem n ;
2. B é uma matriz triangular inferior de ordem n .

Um problema de otimização que apresente estas características torna particularmente simples a tarefa de manipular a matriz de projeção G , escrita neste caso como

$$G = \begin{bmatrix} \begin{array}{c|c} B & \\ \hline B & B \end{array} & & & & \\ & & & & 0 \\ & \vdots & & & \\ & \vdots & & & \\ \begin{array}{c|c} B & \\ \hline B & B \end{array} & & \dots & & \begin{array}{c|c} B & \\ \hline & \end{array} \end{bmatrix} \quad (\text{IV.3})$$

Neste capítulo serão abordados apenas os aspectos relativos à aplicação do método ao problema, sem maiores considerações sobre sua proposição e modelagem. Estes dois itens poderão ser plenamente cobertos pela já extensa literatura, dedicada ao assunto. Ohishi [15] por exemplo, fornece uma descrição detalhada deste problema com base no sistema hidroelétrico brasileiro.

O modelo aqui utilizado é suficientemente exato para permitir o tratamento numérico do problema com algum realismo.

IV.2. Formulação do Problema da Geração Hidroelétrica de Energia

Com base nas considerações realizadas na introdução deste capítulo, a dinâmica dos reservatórios do sistema hidroelétrico será descrita por

$$x(t+1) = x(t) + Bu(t) + y(t); \quad x(0) = x_0 \quad (\text{IV.4})$$

para $t \in [0, T-1]$. Por hipótese $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^n$ e $y(t) \in \mathbb{R}^n$. A matriz $B \in \mathbb{R}^{n \times n}$, formada pelos elementos -1, 0 e 1, estabelece a estrutura de interligação entre as n usinas e para uma numeração adequada destas usinas é tal que

$$b_{ij} = \begin{cases} 0, & \text{se } i < j \\ -1, & \text{se } i = j \\ 1, & \text{se } j \in J_i \end{cases} \quad (\text{IV.5})$$

com b_{ij} denotando o i - j -ésimo elemento de B .

Através da solução deste problema, objetiva-se encontrar uma política ótima de operação, uma lei de controle $u^*(t)$, $t \in [0, T-1]$ para cada uma das n usinas de forma a maximizar a produção de energia do sistema global no horizonte de tempo T especificado. O critério deste problema, aditivamente separável no tempo, será então dado por

$$J(x(t), u(t)) = \sum_{t=0}^{T-1} \phi(x(t), u(t)) + \phi_T(x(T)) , \quad (IV.6)$$

onde

$$\phi(x(t), u(t)) = \sum_{i=1}^n \phi_i(x_i(t), u_i(t)) , \quad (IV.7)$$

e

$$\phi_T(x(T)) = \sum_{i=1}^n \phi_{iT}(x_i(T)) . \quad (IV.8)$$

A função $\phi_i(x_i(t), u_i(t)) : R \times R \rightarrow R$, diferenciável em relação a seus argumentos, representa a função de geração da i -ésima usina e $\phi_{iT}(x_i(T))$, diferenciável em relação a $x_i(T)$, representa uma função que pondera o seu estado final.

O sistema global está ainda sujeito às restrições operativas de reservatório e de turbinagem de cada usina, representadas aqui por limitantes nas variáveis de estado e controle:

$$x_i(t) \in X_i = \{x_i(t) \mid \underline{x}_i \leq x_i(t) \leq \bar{x}_i, \quad t \in [1, T]\} \quad (IV.9)$$

$$u_i(t) \in U_i = \{u_i(t) \mid \underline{u}_i \leq u_i(t) \leq \bar{u}_i, \quad t \in [0, T-1]\} \quad (IV.10)$$

A energia gerada pela i -ésima usina em determinado instante de tempo t é obtida em função do volume de água turbinada $u_i(t)$ (10^9 m^3) pela usina e da cota ou altura de queda de seu reservatório (m), que por sua vez é uma função $h_i(x_i(t))$ do volume de água armazenado, $x_i(t)$ (10^9 m^3). Seja π_i

o rendimento com que a energia potencial da água é transformada em energia elétrica pelas turbinas da i -ésima usina, g a aceleração da gravidade (m/s^2) e ρ o peso específico da água (kg/m^3). A função de geração da i -ésima usina será, então, matematicamente expressa por

$$\phi_i(x_i(t), u_i(t)) = \pi_i \rho u_i(t) g h_i(x_i(t)) \quad (J) \quad (IV.11)$$

A função $h_i(x_i(t))$ relacionando volume do reservatório com altura de queda é em geral representada por um polinômio de quarta ordem (polinômio cota-volume) do tipo,

$$h_i(x_i(t)) = c_{i0} + c_{i1}x_i(t) + c_{i2}x_i^2(t) + c_{i3}x_i^3(t) + c_{i4}x_i^4(t) \quad (IV.12)$$

e embora seja nitidamente côncava, a função de geração que resulta de (IV.11) não apresenta qualquer característica de concavidade ou convexidade (Salviano [19]). Os coeficientes de (IV.12) dependem do perfil topológico de cada reservatório.

A não-concavidade das funções de geração e a dimensão dos sistemas hidroelétricos reais tornam o problema de difícil tratamento. As técnicas de programação matemática que tem sido utilizadas na obtenção da política ótima de operação de sistemas hidroelétricos procuram não contabilizar estes dois fatores, não-concavidade e dimensão, ao mesmo tempo.

Assim, os métodos de agregação que utilizam Programação Dinâmica procuram representar os reservatórios de todas as usinas através de um único reservatório equivalente e o problema resultante, com apenas uma variável de estado e uma variável de controle, pode ser então resolvido com toda a complexidade de modelamento, porém, geralmente com erros de aproximação (agregação/desagregação).

Os métodos de decomposição por dualidade procuram explorar a estrutura geograficamente distribuída dos sistemas hidroelétricos. Embora estes métodos possam tratar grandes sistemas, apresentam por sua vez problemas de convergência devido a não-concavidade das funções de geração envolvidas. Faz-se, então, necessário aproximá-las por funções côncavas e com isto, introduzir imprecisões no modelo.

Os métodos primais baseados em técnicas de gradiente projetado, entre os quais encontra-se o método proposto neste trabalho, permitem que sejam tratadas funções de geração não-côncavas mas a um preço computacional, refletido no tempo necessário para obter-se a política de operação ótima, crescente com a dimensão do sistema hidroelétrico considerado.

Antes da aplicação efetiva do método, o Problema da Geração Hidroelétrica de Energia será traduzido matematicamente no seguinte problema de controle ótimo:

$$\text{Max}_{x(t), u(t)} \sum_{t=0}^{T-1} \phi(x(t), u(t)) + \phi_T(x(T))$$

$$x(t+1) = x(t) + Bu(t) + y(t) ; \quad x(0) = x_0$$

$$x(t) \in X = \{x(t), t \in [1, T] \mid \underline{x} \leq x(t) \leq \bar{x}\} \quad (\text{IV.13})$$

$$u(t) \in U = \{u(t), t \in [0, T-1] \mid \underline{u} \leq u(t) \leq \bar{u}\} .$$

IV.3. Aplicação a um Sistema Real Brasileiro

Nesta seção, o algoritmo desenvolvido para tratar problemas de

D208/BC

controle ótimo pelo método primal proposto será aplicado a um sistema hidroelétrico real. O sistema em questão é um importante segmento do parque hidráulico do Sudeste Brasileiro e constitui por sua topologia um excelente exemplo de aplicação. A figura IV.1 representa esquematicamente o sistema hidroelétrico em estudo.

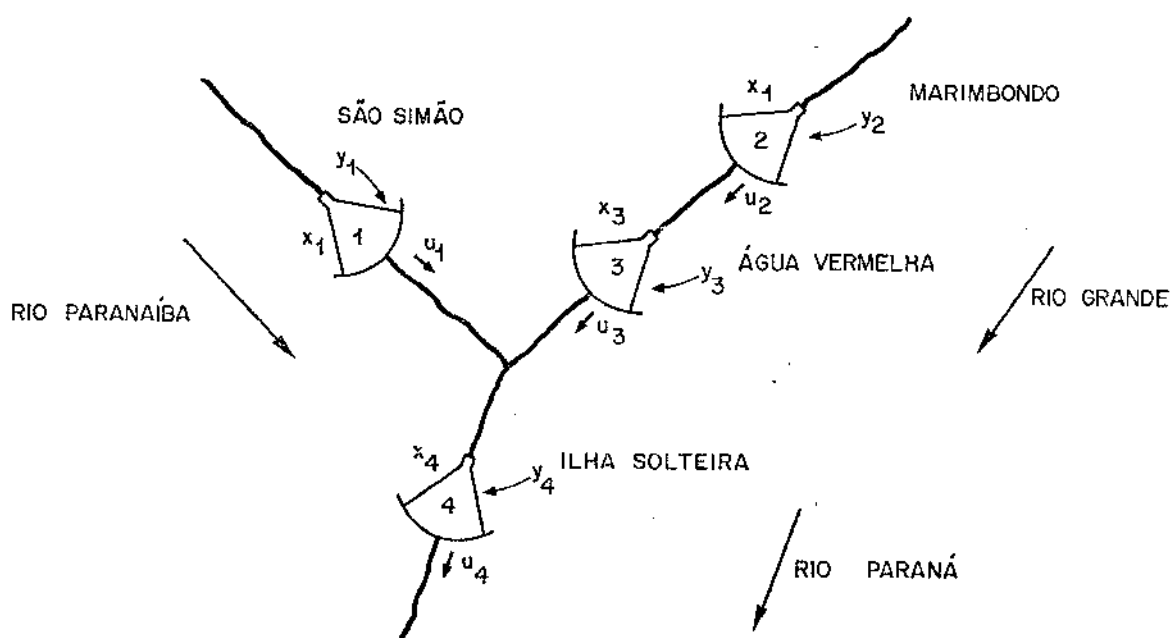


Fig. IV.1 - Sistema Hidroelétrica em Estudo

O horizonte de otimização é de um ano e o intervalo de discretização, de um mês ($T = 12$). Os valores médios dos aportes independentes são os relativos ao ano de 1954 e estão indicados na Tabela IV.1. Na Tabela IV.2 encontram-se os coeficientes dos polinômios cota-volume para as usinas em questão. Os limitantes de capacidade de reservatório e turbinagem bem como o rendimento das turbinas de cada usina estão relacionados na Tabela IV.3. Finalmente, na Tabela IV.4 encontra-se a lei inicial de controle. Os dados aqui utilizados foram obtidos de (Geromel-Luna-Sizenando [11]).

Tabela IV.1 - Aportes Independentes (10^9 m^3)

TEMPO	Y1	Y2	Y3	Y4
0	6.30	5.13	.42	1.10
1	5.76	6.38	.76	1.53
2	5.06	4.00	.45	1.27
3	5.58	3.47	.38	1.59
4	5.17	1.79	.41	.66
5	4.50	3.87	.31	1.81
6	4.18	3.90	.23	1.47
7	4.13	3.79	.19	1.58
8	3.99	3.42	.16	1.41
9	3.61	3.17	.19	1.25
10	4.34	1.97	.21	.46
11	5.93	2.98	.32	.31

Tabela IV.2 - Coeficientes dos Polinômios Cota-Volume

USINA	C0	C1	C2	C3	C4
1	30.03000	8.45670	-.90713	.06007	-.00161
2	33.06000	11.67100	-2.16150	.21912	-.00824
3	17.77000	8.21910	-.92300	.06241	-.00169
4	13.25000	3.27620	-.15157	.00448	-.00005

- 1 - SÃO SIMÃO
- 2 - MARIMBONDO
- 3 - A. VERMELHA
- 4 - I. SOLTEIRA

Tabela IV.3 - Limitantes (10^9 m^3) e Rendimento

USINA	VOLUME MINIMO	VOLUME MAXIMO	TURBINAGEM MINIMA	TURBINAGEM MAXIMA	RENDIMENTO
1	7.00	12.50	1.79	6.63	.89
2	.90	6.15	1.18	6.94	.88
3	4.40	11.00	1.25	7.57	.88
4	12.74	21.16	3.63	20.71	.89

Tabela IV.4 - Lei Inicial de Controle (10^9 m^3)

TEMPO	U1	U2	U3	U4
0	6.30	5.13	5.13	12.53
1	5.76	6.38	6.38	13.67
2	5.06	4.00	4.00	10.33
3	5.58	3.47	3.47	10.60
4	5.17	1.79	1.79	7.62
5	4.50	3.87	3.87	10.18
6	4.18	3.90	3.90	9.55
7	4.13	3.79	3.79	9.50
8	3.99	3.42	3.42	8.52
9	3.61	3.17	3.17	8.00
10	4.30	1.97	1.97	9.01
11	5.93	2.98	2.98	9.22

O algoritmo proposto foi implementado em FORTRAN IV num Micro-Computador CP-700 Prologica e para as condições de contorno

$$x(0) = (9,75 \quad 3,52 \quad 5,80 \quad 16,90)'$$

$$p(11) = (0,00 \quad 0,00 \quad 0,00 \quad 0,00)'$$

obteve-se uma solução ótima do problema (96 variáveis, 240 restrições) após 21 minutos e 8,60 segundos de execução (*). A evolução do critério a cada iteração do algoritmo está ilustrada na Tabela IV.5.

O cálculo do gradiente reduzido foi responsável por grande parte do tempo de CPU obtido, inviabilizando desta forma, o emprego de métodos de busca unidimensional que utilizam informação de gradiente. A lenta convergência do algoritmo, observada na Tabela IV.5 é decorrente do procedimento de busca linear então adotado.

O elevado tempo de CPU necessário para o cálculo do gradiente reduzido motivou uma análise numérica dos polinômios cota-volume visando sobretudo diminuir a ordem destes polinômios. Esta análise é relatada no Apêndice A deste trabalho.

Um aspecto importante a observar com relação a Tabela IV.5 é que o número máximo de restrições ativas numa iteração correspondeu à apenas 17,36% (9,72% em média) do total de restrições do problema que eventualmente poderiam estar ativas (48 restrições de igualdade +96 restrições de desigualdade). Este fato, aliado ao bom condicionamento numérico da função

(*) O mesmo algoritmo implementado em FORTRAN IV no PDP-10/UNICAMP e sob condições idênticas, consumiu apenas 40,17 segundos de CPU.

Tabela IV.5 - Evolução do Algoritmo - Polinômios Cota-Volume de Ordem 4

* EVOLUCAO DO ALGORITMO : (EPS1 = .1500 ; EPS2 = .0500)

ITERACAO PRIMAL	RESTRICCOES ATIVAS	ITERACOES DUAL	POTENCIA (GW)
1	0	0	48.11
2	2	2	48.68
3	4	3	48.99
4	5	4	49.17
5	5	3	49.33
6	7	4	49.43
7	8	5	49.48
8	10	5	49.80
9	11	5	49.98
10	11	5	50.02
11	13	5	50.18
12	13	6	50.34
13	14	8	50.68
14	14	8	50.77
15	16	8	51.00
16	18	7	51.40
17	21	8	51.66
18	20	10	51.74
19	21	11	52.07
20	21	10	52.17
21	23	11	52.45
22	22	10	52.47
23	24	12	52.75
24	22	9	52.82
25	25	10	52.85

dual, tornou possível obter o valor ótimo da variável μ para cada iteração do problema primal, sempre com relativamente poucas iterações (7 em média) no subproblema dual associado ao problema de projeção.

Observe ainda que embora a variável dual μ estivesse restrita, o subproblema dual foi, em termos práticos, resolvido pela técnica de gradientes conjugados de Fletcher-Reeves através de um número de iterações ainda bem inferior à ordem das funções quadráticas envolvidas.

Em consequência, o tempo necessário ao cálculo da projeção do gradiente reduzido não cresceu de maneira significativa ao longo da evolução do algoritmo.

Nas páginas seguintes são apresentadas as trajetórias ótimas de estado, controle, co-estado e potência da usina de São Simão e em seguida é ilustrada a trajetória do sistema em termos de potência total. Em particular, a variável co-estado admite importante interpretação econômica, refletindo o valor da água armazenadas no reservatório.

IV.4. Conclusão

Neste capítulo, o método primal desenvolvido ao longo deste trabalho foi aplicado numericamente na resolução do Problema de Geração Hidroelétrica de Energia.

Os resultados experimentais obtidos utilizando-se um Micro-Computador (8 bits) CP-700 Prológica permitem concluir pela eficiência do método quando aplicado a um sistema hidroelétrico real como o tratado neste capítulo e ao mesmo tempo indicam que o método proposto é também viável para sistemas hidroelétricos de dimensões maiores.

Estes resultados foram possíveis em grande parte devido a estrutura extremamente favorável do problema tratado, a qual permite um elevado nível de especialização do método.

A influência da estrutura do problema sobre o desempenho do algoritmo deve fazer-se notar sobretudo no tempo de processamento necessário à obtenção da solução ótima, justificando deste modo a afirmação de que o método tem melhor desempenho quando aplicado a problemas com algum tipo de estrutura particular como por exemplo, a do problema resolvido neste capítulo.

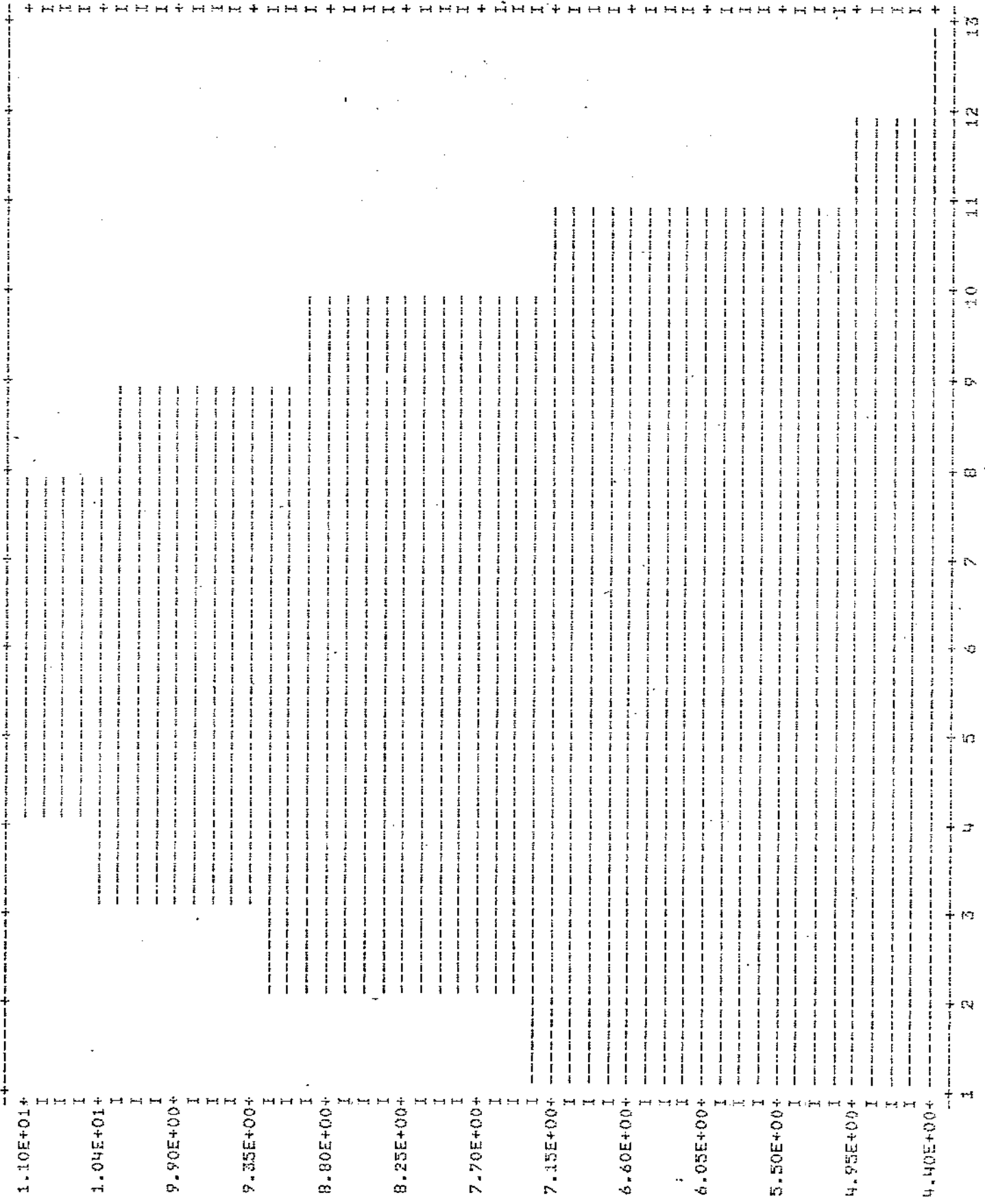


Fig. IV.1 - Trajetória Ótima de Estado (10⁹ m³) da Usina de S. Simão

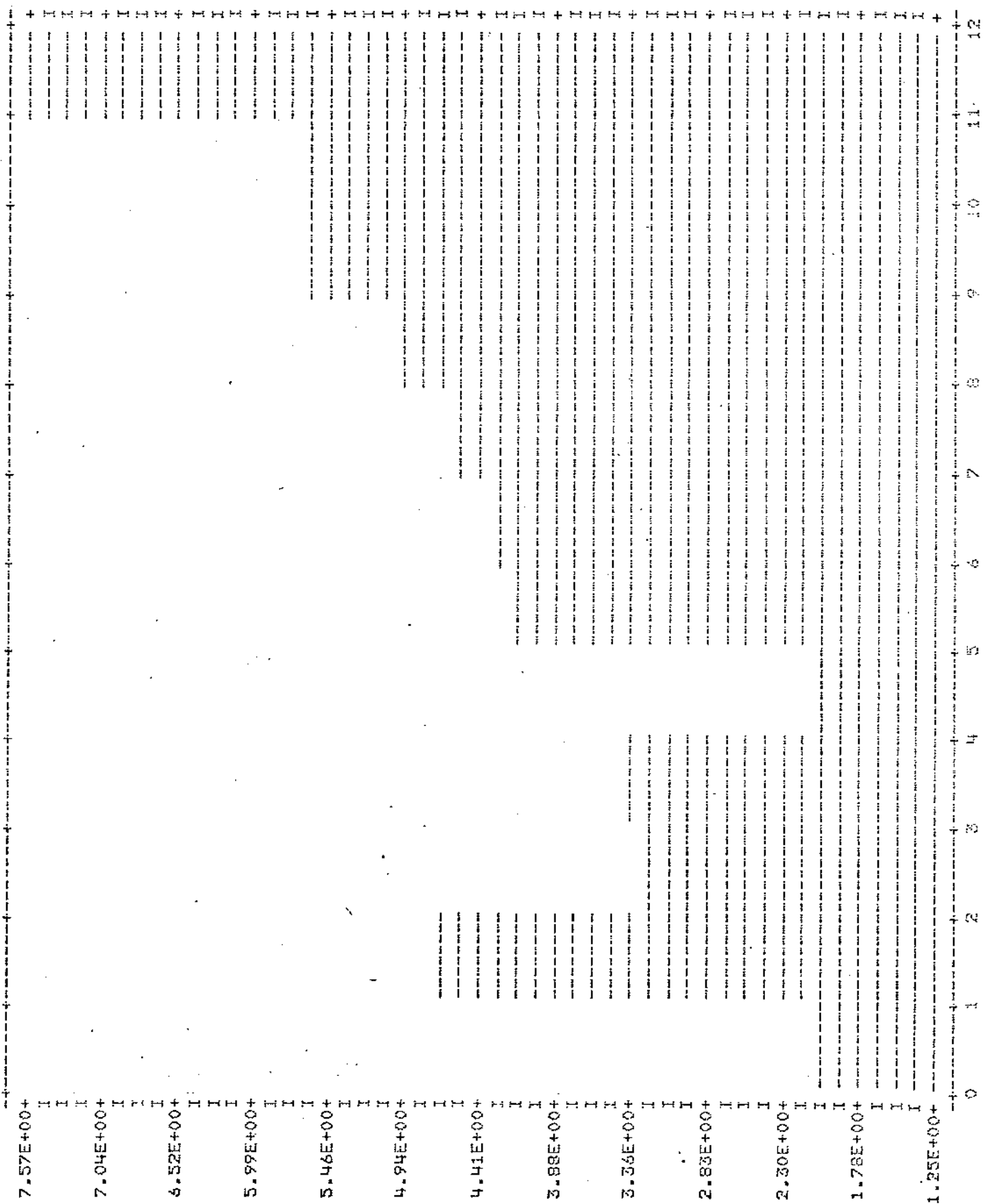
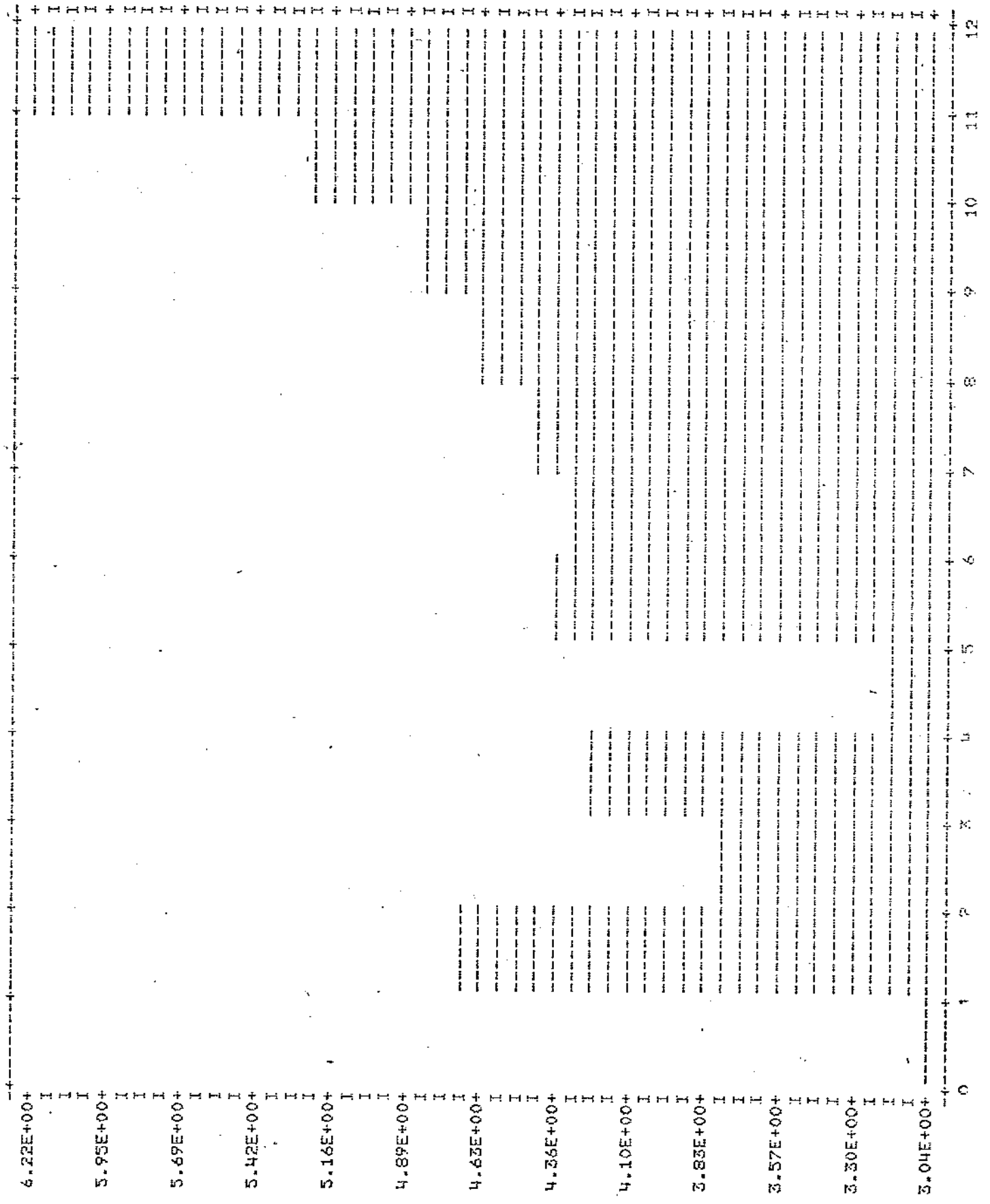


Fig. IV.2 - Trajetória Ótima de Controle (10^9 m^3) da Usina de S. Simão

Fig. IV.5 - Trajetória Ótima de Potência (GW) do Sistema Hidroelétrico



CONCLUSÃO GERAL

Neste trabalho propomos um novo método primal, baseado em técnica de projeção de gradiente para problemas de programação matemática de grande porte, que apresentem restrições lineares. O método proposto tem como principal característica, a forma na qual, manipula os diferentes conjuntos de restrições do problema.

A Teoria da Dualidade, quando aplicada a um problema de projeção estritamente convexo, permite obter uma direção de pesquisa para o método, em que a contribuição das restrições de igualdade pode ser completamente desacoplada da contribuição das restrições de desigualdade ativas, variável a cada iteração do algoritmo.

Em consequência, o problema de programação matemática pode ser em última análise, resolvido a-priori para as restrições de igualdade, ficando o processo de otimização dependente apenas das restrições de desigualdade ativas. A influência destas restrições é diretamente exercida sobre o problema de maximização de uma função dual quadrática, cuja dimensão é igual ao número de restrições de desigualdade ativas, e a partir do qual é obtida a variável dual μ^* . Com base no valor desta variável, o vetor gradiente é então modificado e em seguida projetado sobre as restrições de igualdade do problema.

Quer pelo número de restrições de desigualdade ser bem inferior ao número de variáveis do problema, quer pelas suas próprias características particulares, a maximização da função dual representa um problema de fácil solução.

Outra importante característica do método é evitar, por construção, a necessidade de hipóteses de não-degenerescência, comuns a maioria dos métodos de projeção de gradiente.

Esta característica é particularmente importante, no caso de

problemas de controle ótimo discreto com horizonte de tempo finito. A principal opção em termos de métodos primais hoje disponível para resolução deste tipo de problema, é o método do Gradiente Reduzido proposto por Wolfe (1964) cujo algoritmo, pode apresentar degenerescência se forem consideradas restrições sobre as variáveis de estado do sistema.

Esperamos com este trabalho, haver contribuído para o estudo e desenvolvimento de novos métodos primais baseados em projeção de gradiente e antes de finalizá-lo, gostaríamos de fazer algumas propostas de continuação:

- a) Salviano [19] propõe um método para resolução de problemas de controle ótimo estocástico, com aplicações em sistemas hidroelétricos de energia. Seu trabalho recai em um problema de determinação de trajetórias médias de estado e controle cujas características estatísticas são agora conhecidas. A proposta de continuação do trabalho é construir um simulador do processo estocástico e assim, comprovar o desempenho estatístico da lei de controle obtida.
- b) Utilizar o simulador em sistemas hidroelétricos de grande porte e comparar seu desempenho em relação às técnicas adotadas tradicionalmente no país.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Abadie, J. (Editor), Integer and Nonlinear Programming, North-Holland Publishing Co. - Amsterdam - London, 1970.
- [2] Athans, M. & Falb, P.L., Optimal Control - An Introduction to the Theory and its Applications, McGraw-Hill, 1966.
- [3] Avriel, M., Nonlinear Programming: Analysis and Methods, Prentice-Hall, London, England, 1976.
- [4] Bazaraa, S.M. & Shetty, C.M., Nonlinear Programming Theory and Algorithms, John Wiley and Sons, Inc., 1979.
- [5] Belloni, M.R., Decomposição em Programação Matemática Via Projeção - Análise e Aplicação, Tese de Mestrado, IMECC/UNICAMP, 1982.
- [6] Canon, M.D., Collum, C.D. & Polak, E., Theory of Optimal Control and Mathematical Programming, McGraw-Hill Book Co., New York, 1970.
- [7] Ferreira, P.A.V. & Geromel, J.C., Um Novo Método Primal para Otimização de Problemas de Programação Matemática de Grande Porte, Nota Interna Nº 10/83, DEE/FEC/UNICAMP, 1983.
- [8] Geromel, J.C., Técnicas de Decomposição em Programação Matemática - Anais do III Simpósio Castelan em Matemática Aplicada - Florianópolis, Abril, 1983.
- [9] Geromel, J.C., Controle Ótimo Determinístico, IV Escola de Matemática Aplicada, 23/01/84 a 10/02/84, Rio de Janeiro - RJ.

- [10] Geromel, J.C. & Baptistella, L.F.B., Feasible Direction Method for Large Scale Nonconvex Programs: Decomposition Approach, Journal of Optimization Theory and Applications, Vol. 35, No 2, 1981.
- [11] Geromel, J.C., Luna, H.P.L. & Sizenando, L.C., Saving Fuel for Electrical Energy Through Optimal Control of Hydraulic Resources, IFAC Congress VIII, Vol. XX, Kyoto, 1981.
- [12] Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley Publishing Company, Reading, Massachusetts, 1964.
- [13] Lasdon, L.S., Optimization Theory for Large Systems, The MacMillan Company, London, England, 1970.
- [14] Luemberger, D.G., Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Co., Reading, Massachusetts, 1973.
- [15] Ohishi, T., Aspectos da Otimização da Escala de Geração em Sistemas de Potência Hidrotêrmicos, Tese de Mestrado, DEE/FEC/UNICAMP, 1981.
- [16] Rosen, J.B., The Gradient Projection Method for Nonlinear Programming, I. Linear Constraints, SIAM Journal of Applied Mathematics, Vol. 9, 1960.
- [17] Sage, A.P. & White III, C.C., Optimum Systems Control, 2nd Edition, Prentice-Hall, Inc., New Jersey, 1977.

- [18] Sakarovich, M., Notes on Linear Programming, Van Nostrand Reinhold Co, New York, 1971.
- [19] Salviano, O., Modelamento Linear e Controle Ótimo Estocástico de Usinas Hidráulicas com Representação Individualizada, Tese de Mestrado, DEE/FEC/UNICAMP, 1982.
- [20] Silveira, J.V., Programação Quadrática Aplicada à Análise de Rentabilidade e Risco na Agricultura, Tese de Mestrado, DCC/ICEX/UFMG, 1982.
- [21] Soares, Fº, Secundino, Programação Matemática de Grande Porte: Uma Aplicação à Sistemas Hidrotérmicos de Potência, Tese de Doutorado, DEE, FEC/UNICAMP, 1978.

APÉNDICES

APÊNDICE A

ANÁLISE NUMÉRICA DOS POLINÔMIOS COTA-VOLUME
USINAS DE S. SIMÃO, MARIMBONDO, A. VERMELHA E I. SOLTEIRA

Como mencionado no Capítulo IV deste trabalho, a dependência entre a cota ou altura de queda d'água e o volume do reservatório de uma usina do parque hidráulico brasileiro é expressa por uma função côncava, um polinômio de quarta ordem do tipo

$$h_i(x_i(t)) = c_{0i} + c_{1i}x_i(t) + c_{2i}x_i(t)^2 + c_{3i}x_i(t)^3 + c_{4i}x_i(t)^4, \quad (\text{A.1})$$

onde $x_i(t)$ é o volume armazenado no reservatório da i -ésima usina no instante t e $c_{0i} \dots c_{4i}$ são constantes que dependem do perfil topológico de cada reservatório.

Neste Apêndice analisa-se a viabilidade de expressar (A.1) com certa margem tolerável de erro, através de um polinômio de menor ordem. Obviamente, esta análise parte da hipótese de que (A.1) representa o processo físico mencionado com exatidão. Na redução de ordem do polinômio será utilizada a técnica de Ajuste Ótimo de Parâmetros.

Seja para tanto $\alpha' = [\alpha_1 \quad \alpha_2 \quad \dots \quad \alpha_m]$ um vetor que contem m valores para a cota da i -ésima usina, calculadas a partir de (A.1), para m valores distintos de volume entre \underline{x}_i e \bar{x}_i .

Deseja-se, então, encontrar um vetor de parâmetros $a' = [a_{0i} \quad a_{1i} \quad \dots \quad a_{ri}]$ sendo r , a ordem arbitrada para o polinômio

$$h_{ai}(x_i(t)) = a_{0i} + a_{1i}x_i(t) + \dots + a_{ri}x_i(t)^r \quad (\text{A.2})$$

em que $a^* \in R^r$ seja a solução ótima do seguinte problema quadrático estritamente convexo:

$$\text{Min}_a \frac{1}{2} (\alpha - H a)' Q^{-1} (\alpha - H a) \quad (\text{A.3})$$

com

$$H \in R^{m \times r} \triangleq \begin{bmatrix} 1 & x_i & \dots & x_i^r \\ 1 & x_i & \dots & x_i^r \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 1 & x_i & \dots & x_i^r \end{bmatrix} \quad (\text{A.4})$$

e

$$Q^{-1} \in R^{m \times m} \triangleq \begin{bmatrix} \frac{1}{\alpha_1^2} & & 0 \\ & \frac{1}{\alpha_2^2} & \\ 0 & & \frac{1}{\alpha_m^2} \end{bmatrix} \quad (\text{A.5})$$

A solução ótima a^* do problema (A.3) minimiza, portanto, o erro percentual quadrático entre os valores obtidos para a cota através do polinômio (A.1) e aqueles obtidos através de (A.2) para os mesmos valores de volume entre \underline{x}_i e \bar{x}_i . Esta solução ótima é facilmente encontrada impondo-se a condição de estacionariedade de (A.3) em a^* . Obtêm-se então:

$$a^* = (H'Q^{-1}H)^{-1}H'Q^{-1} \quad (\text{A.6})$$

De posse das Tabelas IV.2 e IV.3 que definem os coeficientes do polinômio de 4ª ordem e os limitantes dos reservatórios para as usinas consi

deradas - S. Simão, Marimbondo, A. Vermelha e I. Solteira e através de um programa computacional implementado num Micro-Computador CP-700, foram geradas 50 medidas de cota em 50 valores uniformemente distribuídos de volume. Os resultados obtidos através de Ajuste Ótimo de Parâmetros para cada uma das usinas encontram-se relacionados a seguir.

A análise dos resultados obtidos através de Ajuste Ótimo de Parâmetros indicaram ser possível, substituir o polinômio de ordem 4 por uma função linear, com erro percentual associado inferior a 6%. Estes resultados, confirmam inteiramente os obtidos por Ohishi [15].

Os polinômios cota-volume passam, então, a ser representados por:

$$h_{a_i}(x_i(t)) = a_{0i} + a_{1i}x_i(t) \quad (A.7)$$

Com os parâmetros a_{0i} e a_{1i} obtidos para as 4 usinas em questão, o Problema de Geração Hidroelétrica de Energia foi novamente resolvido e desta feita, para as mesmas condições iniciais, constantes e critérios de otimalidade encontrou-se uma política ótima de operação para o sistema hidroelétrico após 19 minutos de CPU, tendo havido, portanto, uma economia de 2 minutos, aproximadamente.

A evolução do algoritmo para a representação dos polinômios cota-volume através de funções lineares está descrita na Tabela A.5. Observe que o valor ótimo do critério foi superior ao obtido com os polinômios de ordem 4, embora a diferença percentual seja inferior a 1%.

A trajetória do sistema em termos de potência total está ilustrada na Fig. A.5 e é essencialmente igual a obtida para o caso anterior a menos de uma pequena diferença provocada pelas aproximações.

Tabela A.1 - Usina São Simão

	COEFICIENTES					
	EPM*	A0	A1	A2	A3	A4
* ORDEM 0 :	8.477 %	67.01289				
* ORDEM 1 :	.610 %	48.48879	1.91744			
* ORDEM 2 :	.057 %	41.18476	3.45705	-.07897		
* ORDEM 3 :	.026 %	43.75880	2.64027	.00599	-.00290	
* ORDEM 4 :	.000 %	30.03000	8.45670	-.90713	.06007	-.00161

Tabela A.2 - Usina Marimbondo

	COEFICIENTES					
	EPM*	A0	A1	A2	A3	A4
* ORDEM 0 :	25.059 %	53.55040				
* ORDEM 1 :	5.734 %	41.59181	3.66532			
* ORDEM 2 :	1.390 %	36.67203	7.22928	-.51667		
* ORDEM 3 :	.136 %	33.75591	10.57984	-1.59888	.10318	
* ORDEM 4 :	.000 %	33.06000	11.67100	-2.16150	.21912	-.00826

Tabela A.3 - Usina Água Vermelha

	COEFICIENTES					
	EPM*	A0	A1	A2	A3	A4
* ORDEM 0 :	16.670 %	48.01890				
* ORDEM 1 :	1.790 %	32.38264	2.09883			
* ORDEM 2 :	.353 %	26.97889	3.60847	-.09881		
* ORDEM 3 :	.077 %	22.81377	5.36853	-.33665	.01032	
* ORDEM 4 :	.000 %	17.77000	8.21910	-.92300	.06241	-.00169

Tabela A.4 - Usina Ilha Solteira

	COEFICIENTES					
	EPM*	A0	A1	A2	A3	A4
* ORDEM 0 :	10.695 %	42.65359				
* ORDEM 1 :	.356 %	25.27652	1.03756			
* ORDEM 2 :	.074 %	22.28326	1.39886	-.01068		
* ORDEM 3 :	.007 %	17.18939	2.32283	-.06583	.00108	
* ORDEM 4 :	.000 %	13.25000	3.27620	-.15157	.00448	-.00005

(*) EPM = Erro Percentual Máximo

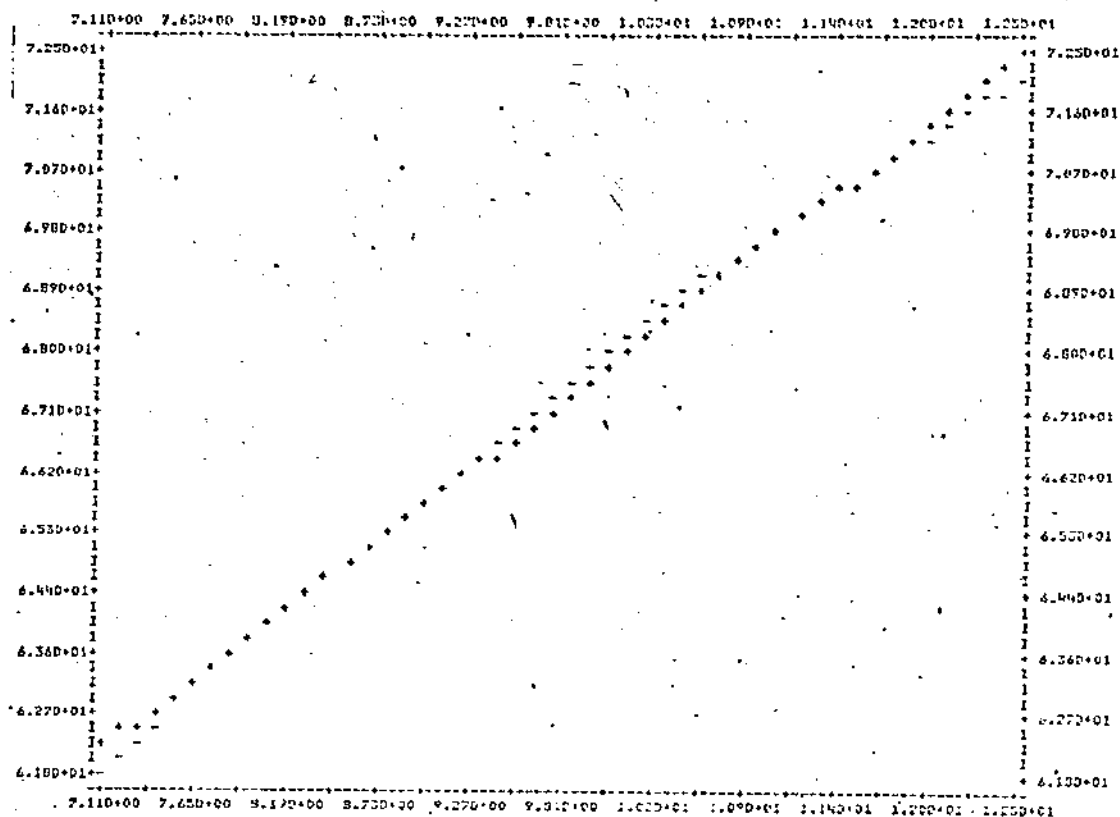


Fig. A.1 - Usina São Simão

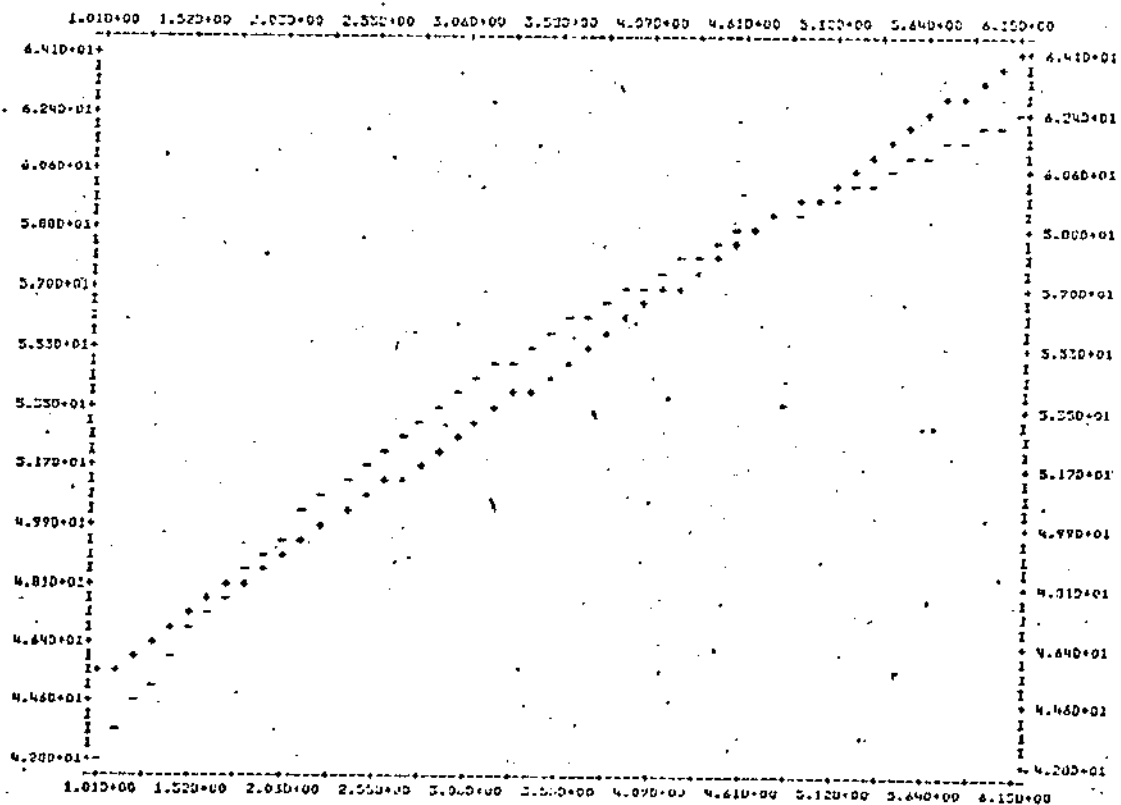


Fig. A.2 - Usina Marimbondo

+ polinômio de ordem 1 ; - polinômio de ordem 4

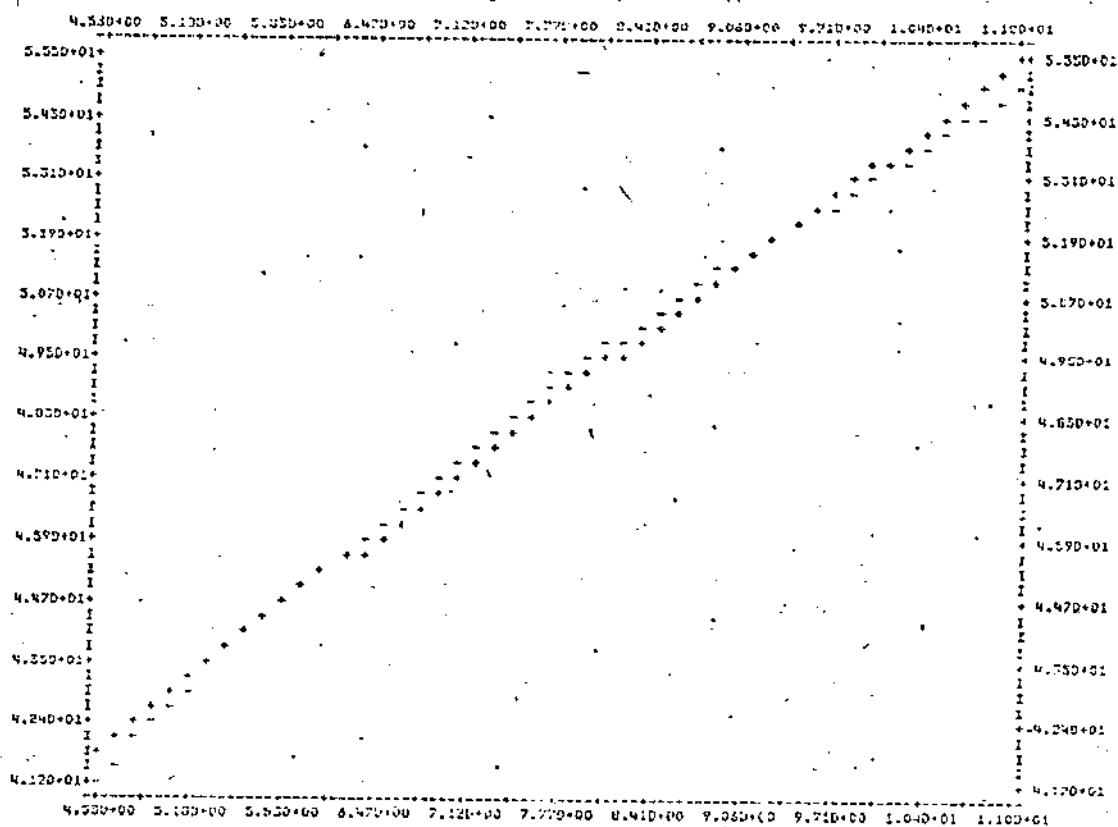


Fig. A.3 - Usina Água Vermelha

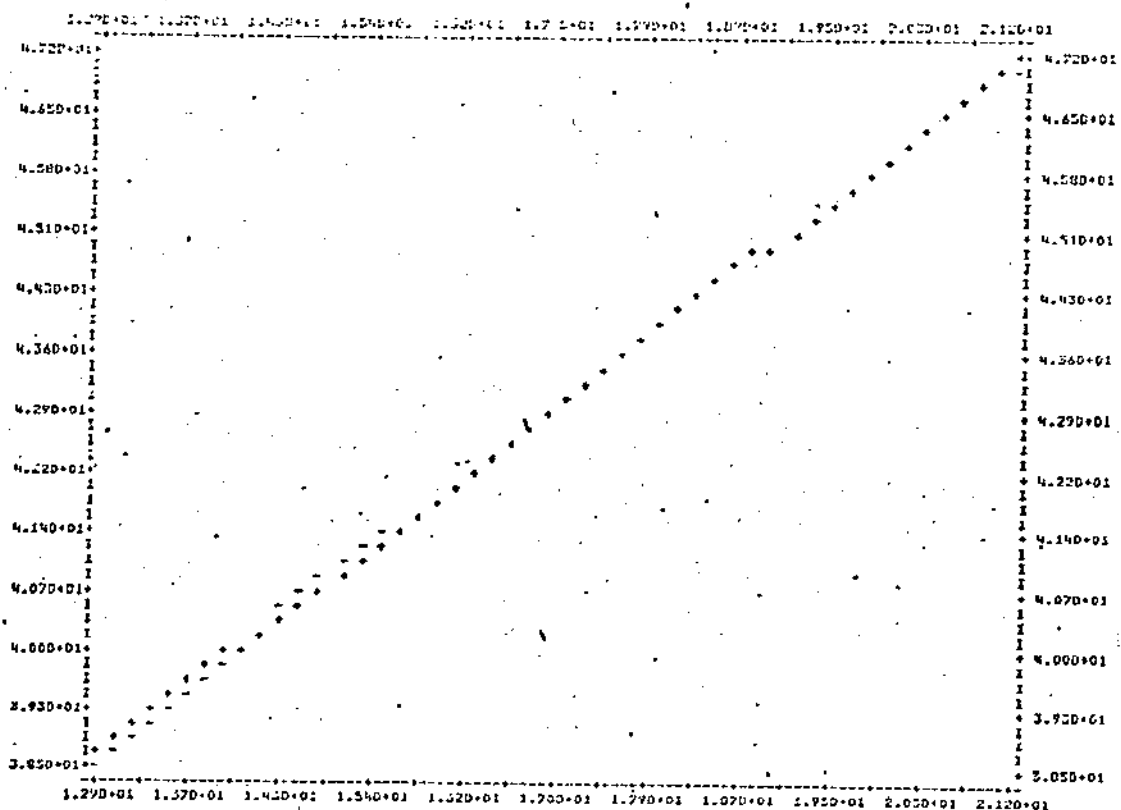


Fig. A.4 - Usina Ilha Solteira

+ polinômio de ordem 1 ; - polinômio de ordem 4

Tabela A.5 - Evolução do Algoritmo para Polinômios Cota-Volume Lineares

* EVOLUCAO DO ALGORITMO : (EPS1 = .1500 ; EPS2 = .0500)

ITERACAO PRIMAL	RESTRICOES ATIVAS	ITERACOES DUAL	POTENCIA (GW)
1	0	0	47.69
2	2	2	47.85
3	3	2	48.23
4	5	3	48.53
5	6	4	48.97
6	8	5	49.12
7	7	3	49.27
8	9	4	49.34
9	10	5	49.58
10	11	6	49.82
11	13	6	49.90
12	14	16	50.15
13	15	6	50.40
14	17	7	50.52
15	18	7	51.18
16	17	7	51.32
17	18	7	51.38
18	20	8	51.61
19	21	9	51.78
20	23	11	52.42
21	20	12	52.48
22	22	15	52.73
23	23	12	52.83
24	25	6	52.99
25	27	21	53.06
26	23	8	53.10
27	24	11	53.21

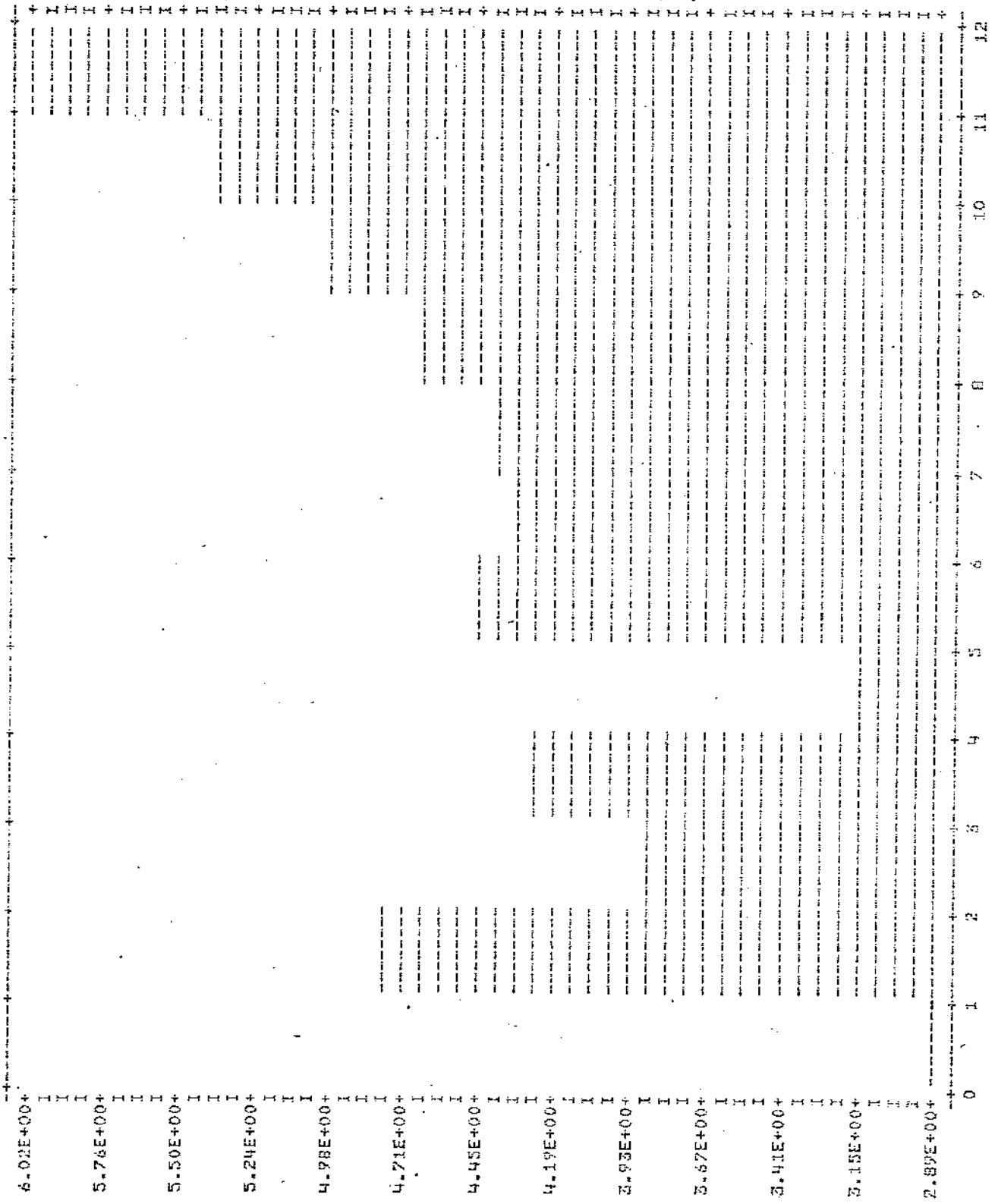


Fig. A.5 - Trajetória Ótima de Potência (GW) do Sistema

É, entretanto, importante ressaltar que embora a aproximação dos polinômios cotá-volume através de funções lineares traga vantagens do ponto de vista computacional, esta aproximação traz ao mesmo tempo uma mudança qualitativa: os polinômios deixam de ser estritamente côncavos (ordem 4) para passar a ser apenas côncavos (ordem 1).

Note finalmente que não houve mudança qualitativa nas funções de geração. Estas funções continuam a apresentar características de não-concavidade.

APÊNDICE B

ANÁLISE PÓS-OTIMIZAÇÃO

Este apêndice estabelece uma figura de mérito para a análise de resultados computacionais obtidos com o método primal proposto neste trabalho e é aplicável a problemas de programação matemática na forma:

$$\begin{aligned} \text{Min } f(x) \\ Ax = b \\ x \in X. \end{aligned} \tag{B.1}$$

onde $x \in R^n$, $b \in R^m$, $A \in R^{m \times n}$ e $f(x) : R^n \rightarrow R \in C^1$

Uma vez encontrada a solução ótima x^* de (B.1), a questão que se coloca e este apêndice tenta responder é: qual o erro percentual cometido no cálculo do valor ótimo da função objetivo para uma dada escolha do critério de parada ϵ_1 do algoritmo? Note que a igualdade de (B.1) só poderá ser satisfeita, para ϵ_1 pequeno mas ainda significativa, a menos de um determinado erro

$$Ax^* - b = \delta. \tag{B.2}$$

A análise da influência do erro δ sobre o valor ótimo da função objetivo permite avaliar o grau de sub-otimalidade da solução, proporcionado pela escolha do critério de parada ϵ_1 . Esta análise faz-se mediante o emprego da função perturbação associada a (B.1):

$$v(y) = \left\{ \text{Min } f(x) \mid Ax = y \right\}_{x \in X}. \tag{B.3}$$

Idealmente ($\delta = 0$), a solução ótima exata \bar{x} de (B.1) faria

(B.3) assumir o valor $v(b)$. Entretanto, a introdução de um erro δ diferente de zero faz (B.3) assumir o valor $v(b + \delta)$ que corresponde à solução x^* obtida iterativamente, segundo um determinado critério ϵ_1 .

Supondo diferenciabilidade de $v(y)$ é possível então escrever, em primeira ordem,

$$v(b) \approx v(b + \delta) - \left(\frac{dv(y)}{dy} \Big|_{y=b+\delta} \right) \delta \quad (\text{B.4})$$

onde

$$\frac{dv(y)}{dy} \Big|_{y=b+\delta} = -\lambda^* \quad (\text{B.5})$$

com $\lambda^* \in \mathbb{R}^m$ representando o vetor de multiplicadores de Lagrange associado às restrições de igualdade de (B.1) e obtido através do algoritmo. Tomando o erro percentual introduzido por δ em (B.4) resulta finalmente

$$\left| \frac{v(b) - v(b + \delta)}{v(b + \delta)} \right| = \left| \frac{\lambda^{*T} \delta}{f(x^*)} \right| \times 100 \quad (\text{B.6})$$

Esta figura de mérito foi utilizada nos exemplos do Capítulo II deste trabalho. O maior erro percentual obtido entre todas as rendas R do EXEMPLO 1 foi de 0,13%. Para o EXEMPLO 2, o erro percentual foi de 0,02%.

APÊNDICE C

LISTAGEM DOS PROGRAMAS COMPUTACIONAIS

DIMENSION A(500), AT(500), X(50), XI(50), XS(50), RAT(500), TOL(50)
 DIMENSION ATVP(500), P(500), S(500), BS(50,50), ST(500)
 DIMENSION BF(500), GF(500), C(50), BTM(100), GBTMI(50), S(50)
 DIMENSION V(500), VA(500), PBT(500), Q(500), M(50), IDEN(100)
 DIMENSION EPSION(50), BS(50), FIM(50), d(50,50), AX(50)
 REAL MI(50), LAMBDA(50)

```

*****
*
*   PPPP  RRRR  III  M   M   AA   L
*   P   P  R   R   I   M   M   A   A   L
*   PPPP  RRRR  I   MM  MM  AAAA  L
*   P   P  R   R   I   M  MM  R   A   L
*   P   P   R   R  III  M   M   A   A  LLLL
*
*****

```

OBJETIVO :

RESOLVER PROBLEMAS DE OTIMIZACAO DO TIPO :

$$\text{Min } f(x)$$

$$\text{s.t.}$$

$$A_1 x = b \quad , \quad A_2 x \leq c \quad ; \quad x \geq 0$$

$$x_1 \leq x \leq x_2 \quad , \quad x \in \mathbb{R}^n$$

METODO :

FARREIRO, P.E.V. & GERGUEL, J.C., UM NOVO METODO
 PRIMAL PARA OTIMIZACAO DE PROBLEMAS DE PROGRAMACAO
 MATEMATICA DE GRANDE PORTE, PUBLICACAO INTERNA NO.
 19/53 - FVC - UNICAMP - 1953.

TIPO DE SAÍDA :

n / d / A / B / X0 / XI / XS

PARÂMETROS DE ENTRADA :

EPS1 - CRITÉRIO DE PARADA DO PROC. PRIMAL

EPS2 - CRITÉRIO DE PARADA DO PROC. DUAL

OPÇÕES DE UTILIZACAO :

CALL / SUBR / OPEN / CLOSE / PRINT / (500)

PRINT / UNID / INV / DUAL / XGORMA / GRDF / F

SETRUM / SUBTITM (FAURON)

```
*****
*                                     *
* PROGRAM PRINCIPAL *
*                                     *
*****
```

```
INTEGER R,RR
CALL SETRUM
WRITE(S,1)
PURCAT(/IOX,'* ARCHIVO DE ENTRADA = ',S)
READ(S,4) IR
WRITE(S,2)
PURCAT(/IOX,'* ARCHIVO DE SALIDA = ',S)
READ(S,4) IA
READ(IR,3)(IDFN(I),I=1,SN)
WRITE(S,3) IDFN(I)
READ(IR,4) N,*
READ(IA,3) L=1, L=2
WRITE(S,4)
READ(IA,5) (X(I),I=1,N)
READ(IR,5) (X(I),I=1,N)
READ(IR,5) (XS(I),I=1,N)
PURCAT(200,/IOG)
WRITE(S)
WRITE(S,2)
IF(.EQ.0) GO TO 101
DO 6 D=1,N
L1=(L-1)*D+1
L2=L+D
READ(IR,7) (X(I),I=L1,L2)
CONTINUE
READ(IR,7) (X(I),I=L1,L2)
PURCAT(SGG)
CALL PRIN(10,4,0,0,0,0,X,X),X0,n,de,IOG)
IF(.EQ.0) GO TO 101
CALL PRIN(4,10,0,0,0,0)
CALL APRD(1,10,0,0,0,0,0)
CALL ARRAY(1,10,0,0,0,0,X,T,0)
CALL INV(0,0,0)
CALL ARRAY(2,10,0,0,0,0,V,0)
CALL APRD(2,10,0,0,0,0,0)
CALL APRD(AT,VA,ATVA,n,0,0,0,0)
DO 10 I=1,n
P(I)=-ATVA(I)
DO 20 J=1,n
K=(J-1)*n+I
P(K)=1.-ATVA(K)
DO 100 I=1,n
P(I)=0
IF(X(I)-XI(I)).GT.TOL(I)) P(I)=1.
```

```

30 IF((XS(I)-X(I)).LT.POL(I)) SI(I)=-1,
CONTINUE
R=R+1
L=L+1
DO 35 J=1,N
IF(SI(J).EQ.0.) R=R+1
35 CONTINUE
IF(R.EQ.0) GO TO 57
CALL GRDF(X,GF)
DO 100 I=1,N
100 GF(I)=GF(I)
GO TO 37
57 DO 36 I=1,R
DO 36 J=1,N
36 S(I,J)=0.
I=0
J=0
37 I=I+1
IF(I.GE.N) GO TO 39
38 J=J+1
IF(S(I,J).EQ.0.) S(I,J)=SI(J)
IF(S(I,J).EQ.0.) GO TO 37
GO TO 38
39 CONTINUE
CALL MTRN(C,GF,SP,SR,0,0)
CALL MTRN(S,SP,SR,0)
CALL MPRD(P,GF,SP,SR,0,0)
CALL MPRD(Q,SP,SR,SR,0,0)
SR=PR*2
DO 40 I=1,SR
40 S(I)=0(I)
CALL MTRN(PR,SP,SR,R,0)
CALL GRDF(X,GF)
CALL MPRD(SP,GF,C,R,SR,0,1)
CALL INIC(R,EPS7,0,C,11,0,LOG)
IF(LOG.EQ.0) GO TO 350
CALL DUAL(R,SPSR,0,0,SI,0)
350 CALL MPRD(ST,SI,ST,SR,0,0,1)
CALL MSUB(GF,PR,1,GRST,0,1,0,0)
37 CALL MPRD(P,GRST,SR,0,0,0,1)
S*GRST=ALPHA*(R,0)
GR*PR=ALPHA*(R,GF)
T=1+3.000000/(1.0+S*GR)
IF(PTR.LT.0.01) GO TO 200
DO 200 I=1,2
IF(SI(I).EQ.1.0.AND.S(I).GT.0.) S(I)=0.
IF(SI(I).EQ.-1.0.AND.S(I).LT.0.) S(I)=0.
S(I)=S(I)/SUMP
200 CONTINUE
ALPHA=X-1.0/30
DO 150 I=1,2
IF(S(I).EQ.0.) GO TO 150
IF(S(I).GT.0.) ALPHA=(X(I)-XS(I))/S(I)
IF(S(I).LT.0.) ALPHA=(X(I)-XI(I))/S(I)
IF(ALPHA.LT.ALFA) ALFA=ALPHA
150 CONTINUE
CALL HOLD(C,0.,ALFA,X,0)
XI=XI+1
IF(SI.GE.100) S=0
PI=F(X)

```

```

WRITE(1W,351) A1,R,L,F1
351 FORMAT(/22X,13,*X,13,9X,13,9X,F9.4)
GO TO 1000
2000 CALL NPRO(VA,GFATM1,LAMBDA,M,N,0,0,1)
CALL NPRO(A,X,IX,0,0,0,1)
CALL NSUB(AZ,SE,SPSUDN,1,1,0,0)
CALL TPRD(LAMBDA,EPSELN,FIN,M,1,0,0,1)
F1=F(X)
DO 333 I=1,M
333 LAMBDA(I)=-LAMBDA(I)
      ERRO=XNORMA(M,SPSUDN)
      ERROP=ABS(FI(I)/F1)*100.
      WRITE(1W,340)
      WRITE(1W,340)
      WRITE(1W,340)
      DO 339 I=1,2
      I1=(I-1)*12+1
      I2=I1+12
      WRITE(1W,340) (X(J),J=I1,I2)
339 WRITE(1W,341) (FI(J),J=I1,I2)
      WRITE(1W,345) F1
      IF(X.NE.0) STOP
      WRITE(1W,340)
      WRITE(1W,341) (LAMBDA(J),J=1,M)
      WRITE(1W,347)
      WRITE(1W,347) (SPSUDN(J),J=1,N)
      WRITE(1W,347) LAMBDA,ERROP
340 FORMAT(/10X,12(' '))
341 FORMAT(/22X,13*12(1X*))
345 FORMAT(/22X,13X,F1)
347 FORMAT(/22X,12(' ') =1,F15.4)
349 FORMAT(/10X,12(' '+,1X))
351 FORMAT(11X,12(F1.1,5X))
353 FORMAT(/22X,12 * LAMBDA *')
355 FORMAT(/22X,12 * EPSUDN *')
357 FORMAT(/22X,100*1A 00 ERRO = ',F7.4,
+ //10X,12ERRO PORCENTUAL = ',F7.4,' POR CENTO')
CALL ROUTH(ITERPO)
CPU=FLOAT(ITS*PP)/100.0
WRITE(1W,1983) CPU
1983 FORMAT(/22X,12 * CPU DE CPU = ',F7.2,' SECS')
STOP
END

-----
FUNCAO LAMBDA(M,X)
-----

OBJETIVO : CALCULAR A NORMA EUCLIDIANA DE UM VETOR DE N
----- COMPONENTES.

COMPOSICAO X(1)
SUM=0.
DO I=1,
SUM=SUM+X(I)**2.
CONTI=0.
LAMBDA=SQRT(SUM)
* LAMBDA

```


END

```
-----
SUBROUTINE INIC(R,EPS2,Q,C,MI,L,LOG)
-----
```

```
OBJETIVO : DETERMINAR A SOLUCAO OTIMA IRRESTRITA
----- DA FUNCAO DUAL, FORNECENDO-A COMO CONDI-
          CIONIO INICIAL PARA A SUBROUTINA 'DUAL'.
```

```
METODO : FLETCHER-REEVES.
-----
```

```
SUBPROGRAMAS UTILIZADOS : MPRD / TADD / XFORMA / TPRD
-----
```

```
Obs. : OBTIEM A SOLUCAO OTIMA EM B ITERACOES, A CADA
--- COMPOZICAO NEGATIVA DE 'MI' SERA ATRIBUINDO
O VALOR C.
```

```

DIMENSION Q(1),C(1),QMI(50),GRDFI(50),D(50)
DIMENSION QD(50),OTGRDFI(1),OTD(1),OTC(1)
REAL MI(1)
INTEGER R
DO 1 I=1,R
MI(I)=C.
CALL MPRD(Q,MI,QMI,R,R,0,0,1)
CALL TADD(QMI,C,GRDFI,R,1,0,0)
ENDPRD=XFORMA(Q,GRDFI)
IF (ENDPRD.LT.EPS2) RETURN
DO 10 I=1,R
D(1)=GRDFI(1)
DO 40 D=1,R
CALL TPRD(D,GRDFI,OTGRDFI,R,1,0,0,1)
CALL MPRD(Q,D,QD,R,R,0,0,1)
CALL TPRD(QD,D,OTD,R,1,0,0,1)
ALFA=-OTGRDFI(1)/OTD(1)
DO 20 I=1,R
MI(I)=MI(I)+ALFA*D(I)
IF (D.DD.R) GO TO 30
CALL MPRD(Q,MI,QMI,R,R,0,0,1)
CALL TADD(QMI,C,GRDFI,R,1,0,0)
ENDPRD=XFORMA(Q,GRDFI)
IF (ENDPRD.LT.EPS2) GO TO 50
CALL TPRD(GRDFI,D,OTD,R,1,0,0,1)
BETA=OTD(1)/OTD(1)
DO 30 I=1,R
D(I)=GRDFI(I)-BETA*D(I)
40 CONTINUE
LOG=0
DO 50 I=1,R
IF (MI(I).GE.C.) GO TO 60
MI(I)=C
LOG=1
CONTINUE
50 RETURN
END
```

```
-----
SUBROUTINE DUAL(B,EPSZ,Q,C,M1,L)
-----
```

```
OBJETIVO : RESOLVER O SUBPROBLEMA DUAL, DETERMINANDO
----- O VALOR OTIMO DA VARIAVEL DUAL 'M1'.
```

```
METODO : GRADIENTE PROJETADO.
-----
```

```
SUBPROGRAMAS UTILIZADOS : INIC / MPRO / MADD / XNORMA / TPRO
-----
```

```
Obs. : O NO. TOTAL DE ITERACOES DO SUBPROBLEMA DUAL SERA
----- PORTANTO IGUAL A 'Q', NO. DE ITERACOES DA SUBROUTINA
INIC', MAIS O NO. DE ITERACOES DESTA SUBROUTINA.
```

```

DIMENSIAO N(1),C(1),M1(50),GRDPI(50),D(50)
DIMENSIAO DTGDDPI(1),GD(50),DIRD(1)
REAL M1(1),XNORMA
INTEGER K
CALL TPRO(Q,Q,E,GRD,0,0,0,1)
CALL MADD(C,GRDPI,0,1,0,0)
DO 70 I=1,K
M1(M1(1),50,GRDPI) D(1)=GRDPI(1)
M1(M1(1),L,E,GRD,GRDPI(1),50,0) D(1)=GRDPI(1)
M1(M1(1),L,E,GRD,GRDPI(1),50,0) D(1)=0.
CONTINUE
DIRD=INORMA(5,5)
IF(DIRD.NE.EPSZ) RETURN
DO 10 I=1,K
D(1)=D(1)/DIRD
SIGMAX=1.0E30
DO 20 I=1,K
IF(D(1).GE.0.0) 50 TO 40
SIGMA=-D(1)/D(1)
IF(SIGMA.LE.SIGMAX) SIGMAX=SIGMA
CONTINUE
CALL TPRO(Q,GRDPI,DTGDDPI,K,1,0,0,1)
CALL MPRO(Q,GRD,K,0,1,0,1)
CALL TPRO(Q,GRD,DTGDD,K,1,0,1)
SIGMA=-DTGDD(1)/DTGDD(1)
IF(SIGMA.LE.SIGMAX) SIGMA=SIGMAX
DO 30 I=1,K
M1(1)=M1(1)+SIGMA*D(1)
D=L+1
GO TO 10
NEXT M1
END

```

```
-----
SUBROUTINE DUALD(Q,GRD1,GRD2,K,S)
-----
```

```
OBJETIVO : REALIZAR BASES UNIDIMENSIONAL PARA O PROBLEMA
----- O VALOR OTIMO DO PASSO DO PROBLEMA PRINCIPAL.
```

ESTA BUSCA E' DESENVOLVIDA NO INTERVALO DE 0
ATE' O MAXIMO VALOR DE ALFA, ALFAX.

METODO : FALSA POSICAO.

SUBPROGRAMAS UTILIZADOS : TPRO / GRDF

Obs. : NA SUA ATUAL CONFIGURACAO ESTA SUBROTINA REALIZA
--- APENAS UMA ITERACAO RETORNANDO AO P.P. COM O VE/

TOR 'X' ATUALIZADO. DESEJANDO-SE REALIZAR BUSCA

ATE' CONVERGENCIA, INCLUIR UM 'C' AO 'RETORN',

LOGO APÓS '20 X(I)=X(I),...'.
C
C

DIMENSIOX X(I),X1(50),X2(50),S(I),GF(50),GF1(50),GF2(50)

DIMENSIOX GF3(1),GF13(1),GF23(1)

DO 1 I=1,3

X1(I)=X(I)-ALFA1*S(I)

X2(I)=X(I)-ALFA2*S(I)

CONTINUE

CALL GRDF(X1,GF1)

CALL TPRO(GF1,S,GF13,N,1,0,0,1)

GF13(1)=-GF13(1)

IF(GF13(1).GE.0.0) GO TO 50

CALL GRDF(X2,GF2)

CALL TPRO(GF2,S,GF23,N,1,0,0,1)

GF23(1)=-GF23(1)

IF(GF23(1).LE.0.0) GO TO 50

K=V

ALFA=ALFA1-GF13(1)*((ALFA2-ALFA1)/(GF23(1)-GF13(1)))

DO 20 I=1,3

X(I)=X(I)-ALFA*S(I)

K=1000

CALL GRDF(X,GF)

CALL TPRO(GF,S,GF3,N,1,0,0,1)

GF3(1)=-GF3(1)

IF(ABS(GF3(1)).LE.0.0) RETURN

IF(GF13(1).LE.0.0.AND.GF3(1).GE.0.0) GO TO 25

ALFA1=ALFA

GF13(1)=-GF3(1)

K=K+1

GO TO 10

25 ALFA2=-ALFA

GF23(1)=-GF3(1)

K=K+1

GO TO 10

30 DO 40 I=1,3

X(I)=X1(I)

K=1000

DO 50 I=1,3

X(I)=X2(I)

K=1000

END

SUBROUTINE PRINT(I0,I,0,0,SP01,SP02,X,X1,X2,S,GF,IO04)

```

-----
OBJETIVO : IMPRIMIR TITULOS, PARAMETROS E CONDICoes
----- INICIAIS DO PROBLEMA.

```

```

SUBPROGRAMAS UTILIZADOS : ARRAY / F
-----

```

```

DIMENSION X(1), XI(1), XS(1), AA(50,50), BE(1), IDEM(1)
DIMENSION A(1)
WRITE(10,1) (IDEM(I), I=1, 30)
WRITE(10,2) A,B
IF(= .EQ. 0) GO TO 32
CALL ARRAY(1,4,8,50,50,A,AA)
WRITE(10,3)
DO 10 I=1, 4
WRITE(10,39) (AA(I,J), J=1, 8)
CONTINUE
WRITE(10,5)
WRITE(10,3) (BE(J), J=1, 8)
32 WRITE(10,7)
DO 20 I=1, 2
I1=(I-1)*12+1
I2=I+12
40 WRITE(10,3) (X(J), J=I1, I2)
WRITE(10,12)
DO 30 I=1, 2
I1=(I-1)*12+1
I2=I+12
30 WRITE(10,3) (XI(J), J=I1, I2)
WRITE(10,12)
DO 40 I=1, 2
I1=(I-1)*12+1
I2=I+12
40 WRITE(10,3) (XS(J), J=I1, I2)
WRITE(10,3)
WRITE(10,14) EPS1, EPS2
WRITE(10,15)
FORMAT(//10X, 'PROBLEMA : ', /25X, 'MIN F(X) = ', 2X, 30A1, //27X, 'S.O.',
+ //33X, 'A = ', 5X, 'B = R(30) ; A = R(0)',
+ //30X, 'RESTRIÇÕES : A ∈ R(0)')
FORMAT(//10X, 'DADOS : (M = ', 13, ' ; N = ', 13, ')')
FORMAT(//10X, 24(F1.1, 1X))
FORMAT(//10X, 11Z(' '))
FORMAT(//10X, 'E VETOR B =')
FORMAT(//10X, 'A =')
FORMAT(//10X, 'E MATRIZ A =')
FORMAT(//10X, 'E A1 =')
FORMAT(//10X, 'E A2 =')
FORMAT(//10X, 'SOLUCAO : (EPS1 = ', 8X, 1, ' ; EPS2 = ', 8X, 4, ')')
FORMAT(//20X, 'IMPRESSAO RESTRICOES ITERAcoes FIMCAO',
+ //21X, 'PROBLEMA ATIVAS 0040 OBJETIVO')
FORMAT(//10X, 'F(X) = ', F12.4)
FORMAT(//10X, 12(F3.1, 1X))
CONTINUE
END

```

```
-----
SUBROUTINE INV(N,I,DET)
-----
```

OBJETIVO : INVERTIR UMA MATRIZ SIMETRICA DEFINIDA
 ----- POSITIVA 'A' DE ORDEN 'N'.

METODO : DECOMPOSICAO LU ($U = L^T$).

SUBPROGRAMAS UTILIZADOS : (NENHUM)

ONS. : A MATRIZ 'A' E' DESTRUIDA. EM 'A' ESTARA'
 --- SUA INVERSA E EM 'DET', O SEU DETERMINANTE.
 TO. O VALOR 'N' DEVE SER MAIOR OU IGUAL A 1.

```

DIMENSION A(50,50),AL(50,50),F(50,50),R1(50,50)
IF(N.EQ.1) GO TO 15
A(1,1)=SQR(4*(1,1))
AL(1,1)=A(1,1)
F(1,1)=1./AL(1,1)
DO 1 I=2,N
  AL(1,I)=A(1,I)/AL(1,1)
  DO 2 J=2,N
    DO 3 I=J,N
      IF(I.EQ.J) GO TO 4
      AL(I,I)=A(I,I)
      DO 5 K=1,I-1
        AL(I,I)=AL(I,I)-AL(I,K)*AL(I,K)**2.
      AL(I,I)=SQRT(AL(I,I))
      D=1/DET*A(I,I)
      R(1,I)=1./AL(I,I)
      GO TO 2
    AL(I,J)=A(I,J)
    DO 6 K=1,J-1
      AL(I,J)=A(I,J)-AL(I,K)*AL(I,K)**L(J,K)
    AL(I,J)=AL(I,J)/AL(I,J)
  CONTINUE
  DO 7 I=2,N
    R(1,I)=0.
  DO 39 L=1,I-1
    R(1,I)=R(1,I)-AL(I,L)*R(L,I)
  R(1,I)=R(1,I)/AL(I,I)
  CONTINUE
DO 50 J=2,N
DO 50 I=J,N
IF(I.EQ.J) GO TO 50
R(1,J)=0.
DO 60 K=1,I-1
  R(1,J)=R(1,J)+AL(I,K)*R(K,J)
R(1,J)=R(1,J)/AL(I,I)
CONTINUE
DO 9 I=1,N
DO 4 J=1,N

```

```
9      S(I,J)=R(J,I)
      DO 10 I=1,N
      DO 10 J=1,N
      R(1,0)=0.
      DO 10 K=1,N
11      S(1,J)=R(1,J)+RT(L,E)*R(S,J)
      RETURN
15      S(1,1)=1.0/A(1,1)
      RETURN
      END
```

```

1: C
2: C
3: C
4: DIMENSION XI(4),XS(4),XY(12),UI(4),US(4),TOLX(4),TOLU(4)
5: DIMENSION R(48),G(48),DAUX(48),UAUX(4,12),MAUX(48)
6: INTEGER T,Q,B
7: REAL MI
8: COMMON /PAR/N,T,X0(4),A(4,5)
9: COMMON /C1/X(4,12),U(4,12),B(4,4),Y(4,12),DJ(4,12)
10: COMMON /C2/NX,ISX(48),ICX(30),IVX(30)
11: COMMON /C3/NU,ISU(48),ICU(30),IVU(30)
12: COMMON /C4/NRA,EPS2,L,D(1600),C(60),MI(60)
13: COMMON /C5/PI(4),RO(4),GFX(4,12),GFU(4,12),P(4,12)
14: COMMON /C8/ENTOT(12),ENFUS(4,12)
15: C
16: C
17: C
18: C
19: C
20: C
21: C
22: C
23: C
24: C
25: C
26: C
27: C
28: C
29: C
30: C
31: C
32: C
33: C
34: C
35: C
36: C
37: C
38: C
39: C
40: C
41: C
42: C
43: C
44: C
45: C
46: C
47: C
48: C
49: C
50: C
51: C
52: C
53: C
54: C
55: C
56: C
57: C
58: C
59: C
60: C

```

```

*****
*
* P P P P R R R R I I I M M A A L 2 2 2 *
* P P R R I M M A A L 2 2 *
* P P P P R R R R I M M M A A A A L 2 *
* P R R I M M M A A L 2 *
* P R R I I M M A A L L L L 2 2 2 *
*
*****

```

```

OBJETIVO :
-----

RESOLVER PROBLEMAS DE CONTROLE OTIMO DISCRETO
COM HORIZONTE DE TEMPO FINITO NA FORMA :

MIN J(X(K),U(K))

X(K+1) = X(K)+BU(K)+Y(K); X(0) = X0.

X(K) E X=[X] XI<=X<[XS]

U(K) E U=[U] UI<=U<[US] - K E [0,T-1]

X(K) E R(N)
U(K) E R(N)
Y(K) E R(N)
B E R(NXN)

```

```

OBSERVACOES :
-----

1) O PROGRAMA ESTA INTEIRAMENTE ORIENTADO NO SENTIDO
DE RESOLVER PROBLEMAS DE GERACAO HIDROELETRICA DE
ENERGIA.

```

```

61: C          2) POR RESTRICÖES DE CAPACIDADE DO MICRO, O N° DE USI/
62: C          NAS ESTA' LIMITADO EM 4 E O INSTANTE FINAL DE TEM/
63: C          PO EM 12.
64: C
65: C          METODO:
66: C          -----
67: C
68: C
69: C          FERREIRA, P.A.V., UM METODO PRIMAL PARA PROBLEMAS
70: C          DE CONTROLE OTIMO DE SISTEMAS DINAMICOS DE GRAN/
71: C          DE PORTE, TESE DE MESTRADO, DEE - FEC - UNICAMP -
72: C          NOVEMBRO, 1983.
73: C
74: C
75: C          DADOS DE ENTRADA:
76: C          -----
77: C
78: C          - MATRIZ DE INTERCONECCAO: B E R(NXN)
79: C          - POLINOMIOS COTA-VOLUME
80: C          - LIMITANTES DE ESTADO E CONTROLE: XI, XS, UI, US
81: C          - RENDIMENTOS DAS USINAS: RO(I), I=1, N
82: C          - APORTES INDEPENDENTES: Y(K), K E IO, T-1]
83: C          - CONDICÖES INICIAL XO E FINAL P(T-1)
84: C          - LEI DE CONTROLE INICIAL: UO(K), K E IO, T-1]
85: C
86: C          PARAMETROS DE ENTRADA:
87: C          -----
88: C
89: C          TAU - TOLERANCIA PARA IDENTIFICACAO DE RESTRICÖES
90: C               ATIVAS.
91: C          EPS1 - CRITERIO DE PARADA DO PROBLEMA PRIMAL.
92: C          EPS2 - CRITERIO DE PARADA DO PROBLEMA DUAL.
93: C
94: C          SUBPROGRAMAS UTILIZADOS:
95: C          -----
96: C
97: C          AXBU / FGER / VJOTA / DHOU / ATIVA / PASSO / M /
98: C
99: C          DUAL / ENERG / ESCALA / XNORMA (FONTE)
100: C
101: C          GMADD / GMPRD / GTPRD (GSP)
102: C
103: C          HIST / VPLDT (VPLDT)
104: C
105: C
106: C
107: C          WRITE(1,1)
108: C          1      FORMAT(1H1,///5X,26('*'),/5X,'*',24X,'*',/5X,'*',7X,
109: C          +      'OTIMIZACAO',7X,'*',/5X,'*',24X,'*',/5X,'*',11X,'DE',
110: C          +      11X,'*',/5X,'*',24X,'*',/5X,'*',2X,'RECURSOS HIDRAU',
111: C          +      'LICOS',2X,'*',/5X,'*',24X,'*',/5X,26('*'))
112: C          WRITE(1,2)
113: C          2      FORMAT(/5X,'* ARQUIVO DE DADOS = ')
114: C          READ(1,3) IR
115: C          3      FORMAT(I2)
116: C          WRITE(1,4)
117: C          4      FORMAT(/5X,'* ARQUIVO DE SAIDA = ')
118: C          READ(1,3) IW
119: C          READ(IR,5) N,T,IH,ID
120: C          5      FORMAT(4I3)

```



```

121:      DO 6 I=1,N
122: 6      READ(IR,7) (B(I,J),J=1,N)
123: 7      FORMAT(5I3)
124:      DO 77 I=1,N
125: 77     READ(IR,78) (A(I,J),J=1,5)
126: 78     FORMAT(5F9.3)
127:      READ(IR,8) (XI(I),I=1,N)
128: 8      FORMAT(4F6.2)
129:      READ(IR,8) (XS(I),I=1,N)
130:      READ(IR,8) (UI(I),I=1,N)
131:      READ(IR,8) (US(I),I=1,N)
132:      READ(IR,8) (X0(I),I=1,N)
133:      READ(IR,9) (RO(I),I=1,N)
134: 9      FORMAT(4F7.4)
135:      DO 10 I=1,N
136:      READ(IR,11) (U(I,J),J=1,T)
137: 10     READ(IR,11) (Y(I,J),J=1,T)
138: 11     FORMAT(12F6.2)
139:      READ(IR,12) (P(I,T),I=1,N)
140: 12     FORMAT(4F6.2)
141:      READ(IR,13) TAU,EP1,EP2
142: 13     FORMAT(3F6.4)
143: C
144: C
145: C
146:      DO 401 I=1,N
147: 401     PI(I)=RO(I)/259.2
148:      DO 14 I=1,N
149:      TOLX(I)=(XS(I)-XI(I))*TAU
150: 14     TOLU(I)=(US(I)-UI(I))*TAU
151:      CALL AXEU
152:      VJOT2=VJOTA(Z)
153:      VJOT4=-VJOT2
154:      ITER=0
155:      NT=N*T
156:      NTV=2*NT
157:      NTR=5*NT
158:      WRITE(IW,1)
159:      WRITE(IW,15)
160: 15     FORMAT(//10X,'* DADOS GERAIS *')
161:      WRITE(IW,16) N,IH,ID,NTV,NTR
162: 16     FORMAT(//5X,'* NO. DE USINAS = ',I2,//5X,'* HORIZONTE DE ',
163: + 'TEMPO = ',I3,' MESES',//5X,'* INTERVALO DE DISCRETIZACAO = ',
164: + I3,' DIAS',//5X,'* NO. TOTAL DE VARIAVEIS = ',I3,//5X,
165: + '* NO. TOTAL DE RESTRICOES = ',I3)
166:      WRITE(IW,178)
167: 178    FORMAT(///10X,'* UNIDADES *',///5X,'* COTA : M',//5X,
168: + '* VOLUME : 10**9 M3',//5X,'* TURBINAGEM : 10**9 M3',
169: + //5X,'* POTENCIA GERADA : 10**9 W')
170:      IF(IW.EQ.1) READ(1,3) NUM
171:      WRITE(IW,17)
172: 17     FORMAT(1H1,////27X,'* DADOS ESPECIFICOS *',///5X,'USINA',
173: + ' VOLUME VOLUME TURBINAGEM TURBINAGEM RENDIMENTO',/12X,
174: + 'MINIMO MAXIMO MINIMA MAXIMA',/)
175:      DO 18 I=1,N
176: 18     WRITE(IW,19) I,XI(I),XS(I),UI(I),US(I),RO(I)
177: 19     FORMAT(/7X,I1,2X,F7.2,1X,F8.2,2X,F7.2,5X,F7.2,7X,F4.2)
178:      IF(IW.EQ.1) READ(1,3) NUM
179:      WRITE(IW,20)
180: 20     FORMAT(1H1,////20X,'* POLINOMIOS COTA(C)-VOLUME(V) *',//18X

```

```

181:      + 'C = C0+C1*V+C2*V**2+C3*V**3+C4*V**4')
182:      WRITE(IW,21)
183: 21    FORMAT(///5X,'USINA',6X,'C0',8X,'C1',8X,'C2',8X,'C3',8X,
184:      + 'C4',/)
185:      DO 121 I=1,N
186: 121    WRITE(IW,122) I,(A(I,J),J=1,5)
187: 122    FORMAT(/6X,I2,3X,5(F9.5,1X))
188:      IF(IW.EQ.1) READ(1,3) NVM
189:      WRITE(IW,24)
190: 24    FORMAT(1H1,///5X,'* MATRIZ DE INTERCONECCAO *',/)
191:      DO 25 I=1,N
192: 25    WRITE(IW,26) (B(I,J),J=1,N)
193: 26    FORMAT(/13X,4(I2,1X))
194:      WRITE(IW,27)
195: 27    FORMAT(///5X,'* CONDIC0ES DE CONTORNO *',///7X,'USINA',4X,
196:      + 'X(0)',3X,'P(T-1)',/)
197:      DO 28 I=1,N
198: 28    WRITE(IW,29) I,X0(I),P(1,T)
199: 29    FORMAT(/9X,I1,4X,F6.2,3X,F6.2)
200:      WRITE(IW,331)
201: 331    FORMAT(1H1,///13X,'* APORTES INDEPENDENTES *',//5X,
202:      + 'TEMPO',4X,'Y1',6X,'Y2',6X,'Y3',6X,'Y4',/)
203:      DO 332 I=1,T
204:      II=I-1
205: 332    WRITE(IW,32) II,(Y(J,I),J=1,N)
206:      WRITE(IW,30)
207: 30    FORMAT(///13X,'* LEI DE CONTROLE INICIAL *',//6X,'TEMPO',
208:      + 3X,'U1',6X,'U2',6X,'U3',6X,'U4',/)
209:      DO 31 I=1,T
210:      II=I-1
211: 31    WRITE(IW,32) II,(U(J,I),J=1,N)
212: 32    FORMAT(/7X,I2,2X,4(F6.2,2X))
213:      WRITE(IW,33) VJOT4
214: 33    FORMAT(//6X,'* POTENCIA INICIAL GERADA = ',F10.2,' GW')
215:      IF(IW.EQ.1) READ(1,3) NVM
216:      WRITE(IW,1)
217:      WRITE(IW,34) EPS1,EP52
218: 34    FORMAT(///25X,'* EVOLUCAO DO ALGORITMO : (EPS1 = ',F6.4,
219:      + ' ; EPS2 = ',F6.4,' )')
220:      WRITE(IW,35)
221: 35    FORMAT(/25X,'ITERACAO',2X,'RESTRIC0ES',2X,'ITERAC0ES',8X,
222:      + 'POTENCIA',5X,'TEMPO',/26X,'PRIMAL',5X,'ATIVAS',6X,'DUAL',
223:      + 12X,' (GW)',5X,'(MIN:SEG)')
224:      CPU=0.
225: 1000  CALL TMP(0,ITMIN,TSEG,TEMPO)
226:      CALL DHDU
227:      VJOT1=VJOT2
228:      DO 36 J=1,T
229:      DO 36 I=1,N
230:      IJ=(J-1)*N+I
231:      ISX(IJ)=0
232:      ISU(IJ)=0
233:      IF((X(I,J)-XI(I)).LT.TOLX(I)) ISX(IJ)=1
234:      IF((XS(I)-X(I,J)).LT.TOLX(I)) ISX(IJ)=-1
235:      IF((U(I,J)-UI(I)).LT.TOLU(I)) ISU(IJ)=1
236:      IF((US(I)-U(I,J)).LT.TOLU(I)) ISU(IJ)=-1
237: 36    CONTINUE
238:      CALL ATIVA(NT,ISX,ICX,IVX,NX)
239:      CALL ATIVA(NT,ISU,ICU,IVU,NU)
240:      NRA=NX+NU

```

```

241:      DO 38 J=1,T
242:      DO 38 I=1,N
243:      IJ=(J-1)*N+I
244:      DAUX(IJ)=DJ(I,J)
245: 38    R(IJ)=DAUX(IJ)
246:      L=0
247:      IF(NRA.EQ.0) GO TO 42
248:      NRA2=NRA**2
249:      DO 37 I=1,NRA2
250: 37    Q(I)=0
251:      DO 40 I=1,NRA
252:      C(I)=0.
253:      DO 39 K=1,NT
254:      MAUX(K)=M(I,K)
255:      C(I)=C(I)+MAUX(K)*DAUX(K)
256: 39    CONTINUE
257:      DO 40 J=I,NRA
258:      IJ=(J-1)*NRA+I
259:      JI=(I-1)*NRA+J
260:      Q(IJ)=0
261:      Q(JI)=0
262:      DO 65 K=1,NT
263:      MJK=M(J,K)
264: 65    Q(IJ)=Q(IJ)-MAUX(K)*MJK
265:      Q(JI)=Q(JI)
266: 40    CONTINUE
267:      CALL REEVES(LOG)
268:      IF(LOG.EQ.0) GO TO 350
269:      CALL DUAL
270: 350   DO 41 I=1,NT
271:      DO 41 K=1,NRA
272:      MKI=M(K,I)
273: 41    R(I)=R(I)-MKI*MI(K)
274: 42    RNORMA=XNORMA(NT,R)
275:      DNORMA=XNORMA(NT,DAUX)
276:      TETA=RNORMA/(1.0+DNORMA)
277:      IF(TETA.LT.EPS1) GO TO 140
278:      DO 43 I=1,NT
279:      IF(ISU(I).EQ.1.AND.R(I).GT.0.) R(I)=0.
280:      IF(ISU(I).EQ.-1.AND.R(I).LT.0.) R(I)=0.
281:      R(I)=R(I)/RNORMA
282: 43    CONTINUE
283:      DO 44 I=1,NT
284:      II=I
285:      IF(I.GT.N) II=MOD(I,N)
286:      IF(II.EQ.0) II=N
287:      G(I)=0.
288:      DO 44 J=1,I
289:      JJ=J
290:      IF(J.GT.N) JJ=MOD(J,N)
291:      IF(JJ.EQ.0) JJ=N
292:      G(I)=G(I)+E(II,JJ)*R(J)
293: 44    CONTINUE
294:      DO 444 I=1,NT
295:      IF(ISX(I).EQ.1.AND.G(I).GT.0.) G(I)=0.
296:      IF(ISX(I).EQ.-1.AND.G(I).LT.0.) G(I)=0.
297: 444   CONTINUE
298:      CALL PASSO(X,XI,XS,G,ALFA1)
299:      CALL PASSO(U,UI,US,R,ALFA2)
300:      ALFAX=AMIN1(ALFA1,ALFA2)

```

```

301:      DO 45 I=1,N
302:      DO 45 J=1,T
303: 45      UAUX(I,J)=U(I,J)
304: 450      DO 46 I=1,N
305:      DO 46 J=1,T
306:      IJ=(J-1)*N+I
307: 46      U(I,J)=UAUX(I,J)-ALFAX*R(IJ)
308:      CALL AXBU
309:      VJOT2=VJOTA(Z)
310:      IF(VJOT2.LT.VJOT1) GO TO 100
311:      ALFAX=ALFAX/2.
312:      GO TO 450
313: 100      VJOT3=-VJOT2
314:      ITER=ITER+1
315:      IF(ITER.GT.50) GO TO 140
316:      CALL TMP(1,ITMIN,TSEG,TEMPO)
317:      CPU=CPU+TEMPO
318:      WRITE(IW,120) ITER,NRA,L,VJOT3,ITMIN,TSEG
319: 120      FORMAT(/26X,I3,8X,I3,9X,I3,7X,F11.2,4X,I3,':',F5.2)
320:      GO TO 1000
321: 140      CALL TMP(1,ITMIN,TSEG,TEMPO)
322:      CPU=CPU+TEMPO
323:      ITMIN=CPU/60.
324:      TSEG=CPU-ITMIN*60.
325:      WRITE(IW,145) ITMIN,TSEG
326: 145      FORMAT(//40X,'* TEMPO DE CPU = ',I3,':',F5.2)
327:      WRITE(IW,147)
328: 147      FORMAT(1H1,///12X,'* TRAJETORIAS OTIMAS DE ESTADO *',
329:      + ///6X,'TEMPO',4X,'X1',6X,'X2',6X,'X3',6X,'X4',/)
330:      DO 148 I=1,T
331: 148      WRITE(IW,149) I,(X(J,I),J=1,N)
332: 149      FORMAT(/7X,I2,3X,4(F6.2,2X))
333:      IF(IW.EQ.1) READ(1,3) NVM
334:      WRITE(IW,150)
335: 150      FORMAT(1H1,///10X,'* TRAJETORIAS OTIMAS DE CONTROLE *',
336:      + ///6X,'TEMPO',4X,'U1',6X,'U2',6X,'U3',6X,'U4',/)
337:      DO 151 I=1,T
338:      II=I-1
339: 151      WRITE(IW,149) II,(U(J,I),J=1,N)
340:      IF(IW.EQ.1) READ(1,3) NVM
341:      DO 521 I=1,N
342:      DO 521 J=1,T
343: 521      P(I,J)=-P(I,J)
344:      WRITE(IW,152)
345: 152      FORMAT(1H1,///22X,'* CD - ESTADO *',///6X,'TEMPO',
346:      + 4X,'P1',6X,'P2',6X,'P3',6X,'P4',/)
347:      DO 153 I=1,T
348:      II=I-1
349: 153      WRITE(IW,149) II,(P(J,I),J=1,N)
350:      IF(IW.EQ.1) READ(1,3) NVM
351: 2000     WRITE(1,154)
352: 154      FORMAT(1H1,///18X,'* GRAFICOS *',///14X,'VARIABLE',
353:      + 10X,'TIPO',///14X,'ESTADO',14(' '),1',/14X,'CONTROLE',
354:      + 12(' '),2',/14X,'CD-ESTADO',11(' '),3',/14X,'POTENCIA',
355:      + ' TOTAL',6(' '),4',/14X,'POTENCIA P/ USINA',3(' '),5',
356:      + /14X,'FIM DE EXECUCAO',5(' '),6')
357:      WRITE(1,155)
358: 155      FORMAT(//18X,'* SELECIONE : ',//31X,'=> VARIABLE = ')
359:      READ(1,3) NV
360:      IF(NV.EQ.6) GO TO 300

```

```

361:      IF(NV.LT.0.OR.NV.GT.6) GO TO 2000
362:      CALL ENERG
363:      IF(NV.EQ.4) GO TO 166
364:      WRITE(1,156)
365: 156    FORMAT(31X,'=> USINA = ')
366:      READ(1,3) NU
367:      IF(NV.LT.0.OR.NV.GT.4) GO TO 2000
368:      WRITE(1,1983)
369: 1983   FORMAT(/31X,'=> DISPOSITIVO DE SAIDA = ')
370:      READ(1,3) IW
371:      IF(NV.NE.1) GO TO 160
372:      DO 157 I=1,T
373: 157    XY(I)=X(NU,I)
374:      WRITE(IW,158) NU
375: 158    FORMAT(1H1,/55X,'USINA ',I2,' - ESTADO',/)
376:      TT=FLOAT(T)+1.0
377:      CALL HIST(T,1.,TT,XI(NU),XS(NU),XY,IW)
378:      GO TO 2000
379: 160    IF(NV.NE.2) GO TO 163
380:      DO 161 I=1,T
381: 161    XY(I)=U(NU,I)
382:      WRITE(IW,162) NU
383: 162    FORMAT(1H1,/55X,'USINA ',I2,' - CONTROLE',/)
384:      TT=FLOAT(T)
385:      CALL HIST(T,0.,TT,UI(NU),US(NU),XY,IW)
386:      GO TO 2000
387: 163    IF(NV.NE.3) GO TO 168
388:      DO 164 I=1,T
389: 164    XY(I)=F(NU,I)
390:      CALL ESCALA(T,XY,XMAX,XMIN)
391:      WRITE(IW,165) NU
392: 165    FORMAT(1H1,/54X,'USINA ',I2,' - CO-ESTADO',/)
393:      TT=FLOAT(T)
394:      CALL HIST(T,0.,TT,XMIN,XMAX,XY,IW)
395:      GO TO 2000
396: 166    CALL ESCALA(T,ENTOT,XMAX,XMIN)
397:      WRITE(IW,167)
398: 167    FORMAT(1H1,55X,'POTENCIA TOTAL',/)
399:      TT=FLOAT(T)
400:      CALL HIST(T,0.,TT,XMIN,XMAX,ENTOT,IW)
401:      GO TO 2000
402: 168    IF(NV.NE.5) GO TO 2000
403:      DO 169 I=1,T
404: 169    XY(I)=ENFUS(NU,I)
405:      CALL ESCALA(T,XY,XMAX,XMIN)
406:      WRITE(IW,170) NU
407: 170    FORMAT(1H1,/55X,'USINA ',I2,' - POTENCIA',/)
408:      TT=FLOAT(T)
409:      CALL HIST(T,0.,TT,XMIN,XMAX,XY,IW)
410:      GO TO 2000
411: 300    STOP
412:      END
413: C
414: C
415: C
416: C
417: SUBROUTINE AXBU
418: C
419: C
420: C

```

```

421: C      OBJETIVO :
422: C      -----
423: C      DADOS U(K),Y(K),K E [0,T-1] E B, A SUBROTINA
424: C      'AXBU' DETERMINA A CORRESPONDENTE TRAJETORIA
425: C      DE ESTADO. ESTA SUBROTINA ASSUME QUE B E'
426: C      TRIANGULAR INFERIOR.
427: C
428: C      SUBPROGRAMAS UTILIZADOS :
429: C      -----
430: C      NENHUM.
431: C
432: C      INTEGER T,T1,B
433: C      COMMON /PAR/N,T,X0(4)
434: C      COMMON /C1/X(4,12),U(4,12),B(4,4),Y(4,12),C(4,12)
435: C      DO 20 I=1,N
436: C      C(I,1)=0.
437: C      DO 10 L=1,I
438: C      C(I,1)=C(I,1)+B(I,L)*U(L,1)
439: C      X(I,1)=X0(I)+C(I,1)+Y(I,1)
440: C      T1=T-1
441: C      DO 40 K=1,T1
442: C      KK=K+1
443: C      DO 40 I=1,N
444: C      C(I,KK)=0.
445: C      DO 30 L=1,I
446: C      C(I,KK)=C(I,KK)+B(I,L)*U(L,KK)
447: C      X(I,KK)=X(I,K)+C(I,KK)+Y(I,KK)
448: C      RETURN
449: C      END
450: C
451: C
452: C
453: C      -----
454: C      SUBROUTINE FGER
455: C      -----
456: C      INTEGER T.
457: C
458: C
459: C      OBJETIVO :
460: C      -----
461: C      DADOS X(K+1),U(K),K E [0,T-1], A SUBROTINA 'FGER
462: C      DETERMINA AS DERIVADAS PARCIAS DA FUNCAO DE GERA
463: C      CAO EM RELACAO AO ESTADO E AO CONTROLE.
464: C
465: C      SUBPROGRAMAS UTILIZADOS :
466: C      -----
467: C      NENHUM.
468: C
469: C
470: C      COMMON /PAR/N,T,X0(4),A(4,5)
471: C      COMMON /C1/X(4,12),U(4,12)
472: C      COMMON /C5/PI(4),RO(4),GFX(4,12),GFU(4,12)
473: C      DO 10 I=1,N
474: C      DO 10 J=1,T
475: C      GFX(I,J)=- (PI(I)*U(I,J))*(A(I,2)+2.*A(I,3)*X(I,J)+
476: C      + 3.*A(I,4)*X(I,J)**2.+4.*A(I,5)*X(I,J)**3.)
477: C      GFU(I,J)=-PI(I)*(A(I,1)+A(I,2)*X(I,J)+A(I,3)*X(I,J)**2.+
478: C      + A(I,4)*X(I,J)**3.+A(I,5)*X(I,J)**4.)
479: C      RETURN
480: C      END

```

```

481:
482: C
483: C
484: C
485: FUNCTION VJOTA(Z)
486: C
487: C
488: C
489: C
490: C
491: C
492: C
493: C
494: C
495: C
496: C
497: C
498: C
499: C
500: C
501: C
502: C
503: C
504: C
505: C
506: C
507: C
508: C
509: C
510: C
511: C
512: C
513: C
514: C
515: C
516: C
517: C
518: C
519: C
520: C
521: C
522: C
523: C
524: C
525: C
526: C
527: C
528: C
529: C
530: C
531: C
532: C
533: C
534: C
535: C
536: C
537: C
538: C
539: C
540: C

```

FUNCTION VJOTA(Z)

OBJETIVO :

DETERMINAR O VALOR DO CRITERIO PARA $X(0)$, $X(K+1)$
E $U(K)$, K E $[0, T-1]$ DADOS.

SUBPROGRAMAS UTILIZADOS :

NENHUM.

```

INTEGERS T, T1
COMMON /PAR/N, T, X(4), A(4, 5)
COMMON /C1/X(4, 12), U(4, 12)
COMMON /C5/PI(4), RO(4)
VJOTA=0.
DO 10 I=1, N
VJOTA=VJOTA-(PI(I)*U(I, 1))*(A(I, 1)+A(I, 2)*X(I)+
+ A(I, 3)*X(I)**2.+A(I, 4)*X(I)**3.+A(I, 5)*X(I)**4.)
T1=T-1
DO 20 K=1, T1
KK=K+1
DO 20 I=1, N
VJOTA=VJOTA-(PI(I)*U(I, KK))*(A(I, 1)+A(I, 2)*X(I, K)+
+ A(I, 3)*X(I, K)**2.+A(I, 4)*X(I, K)**3.+A(I, 5)*X(I, K)**4.)
RETURN
END

```

SUBROUTINE DHDU

OBJETIVO :

OBTER O GRADIENTE REDUZIDO DO FUNCIONAL PARA $X(K)$
 $U(K)$, K E $[0, T-1]$ E $P(T-1)$ DADOS. ESTA SUBROTINA A
SOME QUE B E' TRIANGULAR INFERIOR.

SUBPROGRAMAS UTILIZADOS :

FGER.

```

INTEGERS T, T1, B
COMMON /PAR/N, T
COMMON /C1/X(4, 12), U(4, 12), B(4, 4), Y(4, 12), DJ(4, 12)
COMMON /C5/PI(4), RO(4), GFX(4, 12), GFU(4, 12), P(4, 12)
CALL FGER
DO 20 I=1, N
DJ(I, T)=0.

```

```

541:      DO 10 J=1,I
542: 10    DJ(I,T)=DJ(I,T)+B(J,I)*P(J,T)
543: 20    DJ(I,T)=DJ(I,T)+GFU(I,T)
544:      T1=T-1
545:      DO 40 K=1,T1
546:      KT=T-K
547:      KF=KT+1
548:      DO 40 I=1,N
549:      P(I,KT)=P(I,KF)+GFX(I,KF)
550:      DJ(I,KT)=0.
551:      DO 30 J=1,I
552: 30    DJ(I,KT)=DJ(I,KT)+B(J,I)*P(J,KT)
553: 40    DJ(I,KT)=DJ(I,KT)+GFU(I,KT)
554:      RETURN
555:      END

```

```

556: C
557: C
558: C
559: C

```

```

-----
SUBROUTINE ATIVA(N,IS,IC,IV,L)
-----

```

```

561: C
562: C
563: C

```

```

OBJETIVO :

```

```

000000000

```

```

566: C
567: C
568: C
569: C
570: C

```

DADOS O VETOR DE SATURACAO DE ESTADO (ISX) OU CC
 TROLE (ISU), A SUBROTINA 'ATIVA' DETERMINA O NO.
 DE VARIAVEIS DE ESTADO (NX) OU CONTROLE (NU) AT
 VAS E ARMAZENA A MATRIZ D(X) OU D(U) CORRESPONDE
 TE NA FORMA ESPARSA.

```

571: C
572: C
573: C

```

```

EXEMPLO : MATRIZ D(X)
-----

```

```

574: C
575: C
576: C
577: C
578: C
579: C
580: C

```

ICX(I), I=1, NX : INDICA A COLUNA DE "D" EM C
 EXISTE ELEMENTO DIFERENTE
 ZERO.

```

581: C
582: C
583: C
584: C
585: C

```

IVX(I), I=1, NX : INDICA O VALOR DO ELEMENTO
 "D" CORRESPONDENTE.

```

SUBPROGRAMAS UTILIZADOS :
-----

```

```

NENHUM.

```

```

586: C
587: C
588: C
589: C
590: 10
591: C
592: C
593: 20
594: C
595: 30
596: C
597: C
598: C
599: C
600: 40

```

```

DIMENSION IS(1),IC(1),IV(1)
L=0
DO 10 I=1,N
IF(IS(I).NE.0) L=L+1
CONTINUE
I=0
J=0
I=I+1
IF(I.GT.L) GO TO 40
J=J+1
IF(IS(J).EQ.0) GO TO 30
IC(I)=J
IV(I)=IS(J)
GO TO 20
CONTINUE

```



```

601:      RETURN
602:      END
603: C
604: C
605: C
606: C
607:      SUBROUTINE PASSO(V,VI,VS,S,ALFAX)
608: C
609: C
610: C
611: C      OBJETIVO :
612: C      -----
613: C              DETERMINAR O VALOR DO PASSO DE BLOQUEIO, ALFAX.
614: C
615: C      SUBPROGRAMAS UTILIZADOS :
616: C      -----
617: C              NENHUM.
618: C
619: C
620:      DIMENSION V(4,12),VI(1),VS(1),S(1)
621:      COMMON /PAR/N,M
622:      ALFAX=1.0E30
623:      DO 10 J=1,M
624:      DO 10 I=1,N
625:      IJ=(J-1)*N+I
626:      IF(S(IJ).EQ.0.) GO TO 10
627:      IF(S(IJ).LT.0.) ALFA=(V(I,J)-VS(I))/S(IJ)
628:      IF(S(IJ).GT.0.) ALFA=(V(I,J)-VI(I))/S(IJ)
629:      ALFAX=AMIN1(ALFAX,ALFA)
630: 10    CONTINUE
631:      RETURN
632:      END
633: C
634: C
635: C
636: C
637:      FUNCTION M(I,J)
638: C
639: C
640: C
641: C
642: C      OBJETIVO :
643: C      -----
644: C              DADOS B,NX,NU,ICX,ICU,IVX E IVU, A FUNCTION M(I,J)
645: C              DETERMINA QUALQUER ELEMENTO DE "M" DESEJADO. ESTA
646: C              FUNCTION E' APENAS UM SUBPROGRAMA DE OPERACOES LOGI-
647: C              GICAS.
648: C
649: C
650:      INTEGER B
651:      COMMON /PAR/N
652:      COMMON /C1/X(4,12),U(4,12),B(4,4)
653:      COMMON /C2/NX,ISX(48),ICX(30),IVX(30)
654:      COMMON /C3/NU,ISU(48),ICU(30),IVU(30)
655:      IF(1.GT.NX) GO TO 40
656:      K=ICX(I)
657:      IF(K.GE.J) GO TO 10
658:      M=0
659:      RETURN
660: 10    IF(K.GT.N) GO TO 20

```

```

661:      M=IVX(I)*E(K,J)
662:      RETURN
663: 20    J1=J
664:      K=MDD(K,N)
665:      IF(K.EQ.0) K=N
666:      IF(J1.LE.N) GO TO 30
667:      J1=MOD(J1,N)
668:      IF(J1.EQ.0) J1=N
669: 30    M=IVX(I)*E(K,J1)
670:      RETURN
671: 40    INX=I-NX
672:      IF(ICU(INX).EQ.J) GO TO 50
673:      M=0
674:      RETURN
675: 50    M=IVU(INX)
676:      RETURN
677:      END
678: C
679: C
680: C
681: C
682:      SUBROUTINE REEVES(LOG)
683: C
684: C
685: C
686: C      OBJETIVO :
687: C      -----
688: C      DETERMINAR A SOLUCAO OTIMA IRRESTRITA DA FUNCAO
689: C      DUAL, FORNECENDO-A COMO CONDICAO INICIAL PARA A
690: C      SUBROTINA 'DUAL'.
691: C
692: C      METODO :
693: C      -----
694: C      FLETCHER-REEVES.
695: C
696: C      SUBPROGRAMAS UTILIZADOS :
697: C      -----
698: C      GMPRD / GMADD / GTPRD / XNORMA
699: C
700: C      OBSERVACOES :
701: C      -----
702: C      OBTIDA A SOLUCAO OTIMA NO MAXIMO EM 'NRA = NX+
703: C      NU' ITERACOES, 'A CADA COMPONENTE NEGATIVA DE
704: C      'MI' SERA' ATRIBUIDO O VALOR 0.
705: C
706: C
707: C
708:      DIMENSION DTGDFI(1),DTGD(1),GTGD(1)
709:      INTEGER Q
710:      REAL MI
711:      COMMON /C4/NRA, EPS2, L, Q(1600), C(60), MI(60)
712:      COMMON /C6/QMI(60), D(60), QD(60)
713:      DO 1 I=1, NRA
714: 1      MI(I)=0.
715:      CALL GMPRD(Q, MI, QMI, NRA, NRA, 1)
716:      CALL GMADD(QMI, C, QMI, NRA, 1)
717:      FNORMA=XNORMA(NRA, QMI)
718:      IF(FNORMA.LT.EPS2) RETURN
719:      DO 10 I=1, NRA
720: 10     D(I)=QMI(I)

```

```

721: DO 40 L=1,NRA
722: CALL GTPRD(D,QMI,DTGDFI,NRA,1,1)
723: CALL GMPRD(Q,D,QD,NRA,NRA,1)
724: CALL GTPRD(D,QD,DTQD,NRA,1,1)
725: ALFA=-DTGDFI(1)/DTQD(1)
726: DO 20 I=1,NRA
727: 20 MI(I)=MI(I)+ALFA*D(I)
728: IF(L.EQ.NRA) GO TO 50
729: CALL GMPRD(Q,MI,QMI,NRA,NRA,1)
730: CALL GMADD(QMI,C,QMI,NRA,1)
731: FNORMA=XNORMA(NRA,QMI)
732: IF(FNORMA.LT.EPS2) GO TO 50
733: CALL GTPRD(QMI,QD,GTQD,NRA,1,1)
734: BETA=GTQD(1)/DTQD(1)
735: DO 30 I=1,NRA
736: 30 D(I)=QMI(I)-BETA*D(I)
737: 40 CONTINUE
738: 50 LOG=0
739: DO 60 I=1,NRA
740: IF(MI(I).GE.0.) GO TO 60
741: MI(I)=0.
742: LOG=1
743: 60 CONTINUE
744: RETURN
745: END
746: C
747: C
748: C
749: C
750: SUBROUTINE DUAL
751: C
752: C
753: C
754: C OBJETIVO :
755: C
756: C RESOLVER O SUBPROBLEMA DUAL, DETERMINANDO O VALOR
757: C OTIMO DA VARIÁVEL DUAL 'MI'.
758: C
759: C METODO :
760: C
761: C GRADIENTE PROJETADO.
762: C
763: C SUBPROGRAMAS UTILIZADOS :
764: C
765: C GMPRD / GMADD / GTPRD / XNORMA
766: C
767: C OBSERVAÇÕES :
768: C
769: C O NO. DE ITERAÇÕES DO SUBPROBLEMA DUAL SERÁ
770: C PORTANTO IGUAL AO NO. DE ITERAÇÕES DA SUBRO-
771: C TINA 'REEVES', MAIS O NO. DE ITERAÇÕES DESTA
772: C SUBROTINA CASO A SOLUÇÃO NÃO ESTEJA IRRESTRI-
773: C TA.
774: C
775: C
776: C DIMENSION DTGDFI(1),DTQD(1)
777: C INTEGER Q
778: C REAL MI
779: C COMMON /C4/NRA, EPS2, L, D(1600), C(60), MI(60)
780: C COMMON /C6/QMI(60), D(60), QD(60)

```

```

781: 10    CALL GMFRD(Q,MI,QMI,NRA,NRA,1)
782:      CALL GMADD(QMI,C,QMI,NRA,1)
783:      DO 20 I=1,NRA
784:        IF(MI(I).GT.EPS2) D(I)=QMI(I)
785:        IF(MI(I).LE.EPS2.AND.QMI(I).GT.0.) D(I)=QMI(I)
786:        IF(MI(I).LE.EPS2.AND.QMI(I).LE.0.) D(I)=0.
787: 20     CONTINUE
788:      DNORMA=XNORMA(NRA,D)
789:      IF(DNORMA.LT.EPS2) RETURN
790:      DO 30 I=1,NRA
791: 30     D(I)=D(I)/DNORMA
792:      SIGMAX=1.0E30
793:      DO 40 I=1,NRA
794:        IF(D(I).GE.0.) GO TO 40
795:        SIGMA=-MI(I)/D(I)
796:        SIGMAX=AMINI(SIGMAX,SIGMA)
797: 40     CONTINUE
798:      CALL GTPRD(D,QMI,DTGDFI,NRA,1,1)
799:      CALL GMFRD(Q,D,QD,NRA,NRA,1)
800:      CALL GTPRD(D,QD,DTQD,NRA,1,1)
801:      SIGMA=-DTGDFI(1)/DTQD(1)
802:      SIGMA=AMINI(SIGMAX,SIGMA)
803:      DO 50 I=1,NRA
804: 50     MI(I)=MI(I)+SIGMA*D(I)
805:      L=L+1
806:      GO TO 10
807:      END
808: C
809: C
810: C
811: C
812: C      SUBROUTINE ENERG
813: C      -----
814: C
815: C
816: C      OBJETIVO :
817: C      -----
818: C          DETERMINADA A SOLUCAO OTIMA DO PROBLEMA, A SUB/
819: C          ROTINA 'ENERG' DETERMINA AS TRAJETORIAS OTIMAS
820: C          DE POTENCIA DE CADA USINA ISOLADAMENTE E A PO/
821: C          TENCIA TOTAL GERADA PELO SISTEMA NO TEMPO.
822: C
823: C      SUBPROGRAMAS UTILIZADOS :
824: C      -----
825: C          NENHUM.
826: C
827: C
828: C      INTEGER T,T1
829: C      COMMON /PAR/N,T,X0(4),A(4,5)
830: C      COMMON /C1/X(4,12),U(4,12)
831: C      COMMON /C5/PI(4),RO(4)
832: C      COMMON /C8/ENTOT(12),ENPUS(4,12)
833: C      DO 10 I=1,N
834: C      DO 10 J=1,T
835: C      ENTOT(J)=0.
836: 10     ENPUS(I,J)=0.
837: C      DO 20 I=1,N
838: C      TERMO=(PI(I)*U(I,1))*(A(I,1)+A(I,2)*X0(I)+A(I,3)*X0(I)**2.+
839: +      A(I,4)*X0(I)**3.+A(I,5)*X0(I)**4.)
840: C      ENTOT(1)=ENTOT(1)+TERMO

```

```

841: 20      ENFUS(I,1)=ENFUS(I,1)+TERMO
842:         T1=T-1
843:         DO 30 K=1,T1
844:         KK=K+1
845:         DO 30 I=1,N
846:         TERMO=(PI(I)*U(I,KK))*(A(I,1)+A(I,2)*X(I,K)+A(I,3)*X(I,K)**2.+
847: + A(I,4)*X(I,K)**3.+A(I,5)*X(I,K)**4.)
848:         ENTOT(KK)=ENTOT(KK)+TERMO
849: 30      ENFUS(I,KK)=ENFUS(I,KK)+TERMO
850:         RETURN
851:         END
852: C
853: C
854: C
855: C
856: C      -----
857: C      SUBROUTINE ESCALA(N,X,XMAX,XMIN)
858: C      -----
859: C
860: C      OBJETIVO :
861: C      -----
862: C          DETERMINAR OS VALORES MINIMO E MAXIMO DE UM CONJUNTO
863: C          DE NUMEROS.
864: C
865: C      SUBPROGRAMAS UTILIZADOS :
866: C      -----
867: C          NENHUM.
868: C
869: C
870: C      DIMENSION X(1)
871: C      XMAX=X(1)
872: C      XMIN=X(1)
873: C      DO 10 I=2,N
874: C      XAUX=X(I)
875: C      XMAX=AMAX1(XMAX,XAUX)
876: C      XMIN=AMIN1(XMIN,XAUX)
877: 10      CONTINUE
878:         RETURN
879:         END
880: C
881: C
882: C
883: C
884: C      -----
885: C      FUNCTION XNORMA(N,X)
886: C      -----
887: C
888: C      OBJETIVO :
889: C      -----
890: C          DETERMINAR A NORMA EUCLIDIANA DE UM VETOR DE N
891: C          COMPONENTES.
892: C
893: C      SUBPROGRAMAS UTILIZADOS :
894: C      -----
895: C          NENHUM.
896: C
897: C
898: C      DIMENSION X(1)
899: C      SOMA=0.
900: C      DO 1 I=1,N

```

```
901:      SOMA=SOMA+X(I)**2.  
902: 1     CONTINUE  
903:      XNORMA=SQRT(SOMA)  
904:      RETURN  
905:      END
```

```

1:      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
2:      DIMENSION A(5),X(100),ALFA(100),ALFAC(100),H(5,100)
3:      DIMENSION Q(5,5),Y(5),AC(5),ERRO(100),XY(300),JXY(4)
4:      C
5:      C
6:      C
7:      C
8:      C
9:      C
10:     C
11:     C
12:     C
13:     C
14:     C
15:     C
16:     C
17:     C
18:     C
19:     C
20:     C
21:     C
22:     C
23:     C
24:     C
25:     C
26:     C
27:     C
28:     C
29:     C
30:     C
31:     C
32:     C
33:     C
34:     C
35:     C
36:     C
37:     C
38:     C
39:     C
40:     C
41:     C
42:     C
43:     C
44:     C
45:     C
46:     C
47:     C
48:     C
49:     C
50:     CC
51:     WRITE(1,3)
52:     3      FORMAT(1H1,///20X,'*** FITAGEM DOS POLINOMIOS COTA-VOLUME ***')
53:     WRITE(1,200)
54:     200    FORMAT(//5X,'* COEFICIENTES DO POLINOMIO DE ORDEM 4 *',
55:     +      //5X,'=> ')
56:     READ(1,210)(A(I),I=1,5)
57:     210    FORMAT(5F8.5)
58:     WRITE(1,215)(A(I),I=1,5)
59:     215    FORMAT(1X,5F8.5)
60:     WRITE(1,220)

```

```

*
* FFFFF III TTTT AAA GGG EEEEE M M *
* F I T A A G G E MM MM *
* FFF I T AAAAA G EEE M MM M *
* F I T A A G GG E M M *
* F III T A A GGG EEEEE M M *
*
*****

```

OBJETIVO :

ESTE PROGRAMA REALIZA O AJUSTE OTIMO DE PARAMETROS PARA UM DADO POLINOMIO COTA - VOLUME ESPECIFICADO. FORNECE COMO SAIDA OS POLINOMIOS DE ORDENS 0, 1, 2, 3, 4 E O ERRO PERCENTUAL MAXIMO COMETIDO PARA CADA ORDEM CONSIDERADA. ESTE PROGRAMA INCLUE SAIDAS EM FORMA DE GRAFICOS DOS POLINOMIOS E HISTOGRAMAS DO ERRO PERCENTUAL.

METODO :

AJUSTE OTIMO DE PARAMETROS. (APENDICE A)

SUBPROGRAMAS UTILIZADOS :

HX / XMAX / SISLIN (FONTE)

HIST / VPLOT (VPLOT)

DADOS DE ENTRADA :

- COEFICIENTES DO POLINOMIO COTA-VOLUME DE ORDEM 4.
- LIMITANTES INFERIOR E SUPERIOR DO VOLUME ARMAZE/
NADO FELD RESERVATORIO.

```

61: 220   FORMAT(/5X,'* LIMITANTES DO RESERVATORIO *',/5X,'=> ')
62:      READ(1,230) XI,XS
63: 230   FORMAT(2F5.2)
64:      WRITE(1,230) XI,XS
65:      M=50
66:      WRITE(1,240)
67: 240   FORMAT(/5X,'* NO. DO DISPOSITIVO DE SAIDA = ')
68:      READ(1,5) NW
69: 5      FORMAT(I2)
70:      WRITE(NW,600)
71: 600   FORMAT(1H1,///44X,'COEFICIENTES',//20X,'EPM*',7X,'A0',7X,'A1',
72:      + 7X,'A2',7X,'A3',7X,'A4')
73:      N1=0
74:      KEY=0
75: 3000  N2=N1+1
76:      DIFER=XS-XI
77:      DELT=DIFER/FLD0AT(M)
78:      DO 10 I=1,M
79:      X(I)=XI+FLD0AT(I)*DELT
80:      W=X(I)
81:      ALFA(I)=HX(A,W)
82: 10     CONTINUE
83:      DO 20 I=1,M
84:      H(1,I)=1./ALFA(I)
85: 20     CONTINUE
86:      IF(N1.EQ.0) GO TO 40
87:      DO 30 J=2,N2
88:      J1=J-1
89:      DO 30 I=1,M
90:      H(J,I)=(X(I)**J1)/ALFA(I)
91: 30     CONTINUE
92: 40     DO 35 I=1,5
93: 35     AC(I)=0.
94:      DO 50 I=1,N2
95:      DO 50 J=1,N2
96:      Q(I,J)=0.
97:      Y(I)=0.
98:      DO 50 K=1,M
99:      Q(I,J)=Q(I,J)+H(I,K)*H(J,K)
100:     Y(I)=Y(I)+H(I,K)
101: 50     CONTINUE
102:     CALL SISLIN(N2,Q,Y,AC)
103:     DO 60 I=1,M
104:     W=X(I)
105:     ALFAC(I)=HX(AC,W)
106:     ERRO(I)=(DABS(ALFA(I)-ALFAC(I))/ALFA(I))*100.
107: 60     CONTINUE
108:     ERROM=XMAX(M,ERRO)
109:     IF(KEY.EQ.1) GO TO 5000
110:     WRITE(NW,700) N1,ERROM,(AC(I),I=1,N2)
111: 700   FORMAT(/5X,'* ORDEM',I2,' : ',F6.3,' %',1X,5(F8.5,1X),/)
112:     N1=N2
113:     IF(N2.LT.5) GO TO 3000
114:     IF(NW.EQ.1) READ(1,5) KK
115:     KEY=1
116: 4000  WRITE(1,800)
117: 800   FORMAT(1H1,///25X,'SAIDA GRAFICA',///10X,'* ORDEM DO',
118:      + ' POLINOMIO COTA - VOLUME (<=4) = ')
119:     READ(1,5) N1
120:     WRITE(1,900)

```



```

121: 900      FORMAT(/10X,'* DISPOSITIVO DE SAIDA = ')
122:         READ(1,5) NW
123:         IF(NW.GT.2) GO TO 80
124:         GO TO 3000
125: 5000     DO 100 I=1,M
126:         XY(I)=X(I)
127:         II=M+I
128:         XY(II)=ALFA(I)
129:         III=2*M+I
130:         XY(III)=ALFAC(I)
131: 100      CONTINUE
132:         JXY(1)=1
133:         JXY(2)=2
134:         JXY(3)=1
135:         JXY(4)=3
136:         WRITE(NW,110)
137: 110      FORMAT(1H1,/40X,'HISTOGRAMA DO ERRO PERCENTUAL')
138:         CALL HIST(M,NW,ERROM,ERRO)
139:         IF(NW.EQ.1) READ(1,5) KK
140:         WRITE(NW,120)
141: 120      FORMAT(1H1,/30X,'POLINOMIOS COTA-VOLUME')
142:         CALL VPLOT(XY,JXY,M,M,2,0,XI,XS,ALFA(1),ALFA(M),0,NW)
143:         WRITE(NW,130) N1
144: 130      FORMAT(/17X,'- POLINOMIO DE ORDEN 4 ; + POLINOMIO DE ORDEN',I2
145:         IF(NW.EQ.1) READ(1,5) KK
146:         GO TO 4000
147: 80        STOP
148:         END
149: C
150: C
151: C
152: C         -----
153: C         FUNCTION HX(A,X)
154: C         -----
155: C
156: C         OBJETIVO :
157: C         -----
158: C
159: C         ESTE SUBPROGRAMA DETERMINA O VALOR DO POLI/
160: C         NOMIO COTA-VOLUME DE ORDEN 4 ESPECIFICADO
161: C         ATRAVES DE A0,A1,A2,A3,A4, PARA UM DADO VA/
162: C         DE X ENTRE XI E XS.
163: C
164: C         SUBPROGRAMAS UTILIZADOS :
165: C         -----
166: C
167: C
168: C
169: C         IMPLICIT DOUBLE PRECISION(A-H,O-Z)
170: C         DIMENSION A(1)
171: C         HX=A(1)+A(2)*X+A(3)*X**2.+A(4)*X**3.+A(5)*X**4.
172: C         RETURN
173: C         END
174: C
175: C
176: C         -----
177: C         FUNCTION XMAX(N,X)
178: C         -----
179: C
180: C         OBJETIVO :

```

```

181: C -----
182: C DETERMINAR O MAXIMO VALOR ENTRE UM CONJUNTO DE
183: C N NUMEROS.
184: C
185: C SUBPROGRAMAS UTILIZADOS :
186: C -----
187: C NENHUM.
188: C
189: C
190: C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
191: C DIMENSION X(1)
192: C XMAX=X(1)
193: C DO 10 I=2,N
194: C XAUX=X(I)
195: C XMAX=DMAX1(XMAX,XAUX)
196: 10 CONTINUE
197: C RETURN
198: C END
199: C
200: C
201: C -----
202: C SUBROUTINE SISLIN(N,A,B,Y)
203: C -----
204: C
205: C
206: C OBJETIVO :
207: C -----
208: C RESOLVER UM SISTEMA DE N EQUACOES LINEARES. A E'
209: C A MATRIZ DOS COEFICIENTES. B E' O VETOR DE CONS/
210: C TANTES E Y E' A SOLUCAO DO SISTEMA.
211: C
212: C METODO :
213: C -----
214: C DECOMPOSICAO LU (U=L').
215: C
216: C SUBPROGRAMAS UTILIZADOS :
217: C -----
218: C NENHUM.
219: C
220: C
221: C
222: C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
223: C DIMENSION A(5,5),B(1),Y(1),AL(5,5),Z(5)
224: C Y(1)=B(1)/A(1,1)
225: C IF(N.EQ.1) GO TO 10
226: C AL(1,1)=DSQRT(A(1,1))
227: C DO 1 I=2,N
228: 1 AL(I,1)=A(I,1)/AL(1,1)
229: C DO 2 J=2,N
230: C DO 2 I=J,N
231: C IF(I.NE.J) GO TO 4
232: C AL(I,I)=A(I,I)
233: C II=I-1
234: C DO 5 K=1,II
235: 5 AL(I,I)=AL(I,I)-AL(I,K)**2.
236: C AL(I,I)=DSQRT(AL(I,I))
237: C GO TO 2
238: 4 AL(I,J)=A(I,J)
239: C JJ=J-1
240: C DO 6 K=1,JJ

```

```
241: 6      AL(I,J)=AL(I,J)-AL(I,K)*AL(J,K)
242:      AL(I,J)=AL(I,J)/AL(J,J)
243: 2      CONTINUE
244:      Z(1)=B(1)/AL(1,1)
245:      DO 3 I=2,N
246:      Z(I)=B(I)
247:      II=I-1
248:      DO 9 K=1,II
249: 9      Z(I)=Z(I)-AL(I,K)*Z(K)
250:      Z(I)=Z(I)/AL(I,I)
251: 3      CONTINUE
252:      Y(N)=Z(N)/AL(N,N)
253:      NN=N-1
254:      DO 7 L=1,NN
255:      I=N-L
256:      Y(I)=Z(I)
257:      II=I+1
258:      DO 8 K=II,N
259: 8      Y(I)=Y(I)-AL(K,I)*Y(K)
260:      Y(I)=Y(I)/AL(I,I)
261: 7      CONTINUE
262: 10     RETURN
263:      END
```