

**CÁLCULO DE EQUILÍBRIOS ECONÔMICOS
POR MÉTODOS HOMOTÓPICOS**

JOSÉ RAMÓN ARICA CHÁVEZ



UNICAMP

UNIVERSIDADE ESTADUAL DE CAMPINAS

INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E CIÊNCIA DA COMPUTAÇÃO

Ar41c

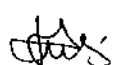
7806/BC

**CAMPINAS - SÃO PAULO
BRASIL**

CÁLCULO DE EQUILÍBRIOS ECONÔMICOS POR MÉTODOS HOMOTÓPICOS

Este exemplar corresponde a redação final da tese devidamente corrigida e defendida pelo Sr. JOSÉ RAMÓN ARICA CHÁVEZ e aprovada pela Comissão Julgadora.

Campinas, 6 de abril de 1987.


Prof.Dr. JOSÉ ANTONIO SCARAMUCCI
Orientador

Dissertação apresentada ao Instituto de Matemática, Estatística e Ciência da Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Matemática Aplicada.

AGRADECIMENTOS

Ao Prof. Dr. José Antonio Scaramucci, pela valiosa orientação que me possibilitou o desenvolvimento deste trabalho.

À Gudélia, pelo estímulo constante que sempre soube me dar.

Aos colegas da Matemática Aplicada, amigos inesquecíveis.

À "Universidad Nacional de Ingeniería", à UNICAMP e ao CNPq, pelo apoio.

À Lourdes, pelo excelente trabalho de datilografia.

Às minhas meninas

Catalina e Natália

ÍNDICE

INTRODUÇÃO	i
CAPÍTULO I - EQUILÍBRIO EM UMA ECONOMIA DE TROCAS	1
1.1. O Teorema de Brouwer	1
1.2. Lema de Sperner	3
1.3. Aplicação do Teorema de Brouwer ao Problema de Equilíbrio Econômico	13
CAPÍTULO II - EQUILÍBRIO EM UMA ECONOMIA COM PRODUÇÃO	19
2.1. Semicontinuidade Superior	19
2.2. Teorema de Kakutani	21
2.3. Extensões do Teorema de Kakutani	23
2.4. Aplicação ao Teorema de Kakutani ao Problema de Equilíbrio Econômico	27
CAPÍTULO III - O PONTO DE VISTA HOMOTÓPICO NO CÁLCULO DE PONTOS FIXOS	34
3.1. O Princípio Homotópico e Aproximações Lineares por Partes	34
3.2. O Acompanhamento de Soluções Lineares por Partes .	35
3.3. Melhoramento da Solução Aproximada	41
3.4. Método Homotópico para Correspondências	42
CAPÍTULO VI - O ALGORITMO DE MERRILL	46
4.1. Triangulação de Kuhn	46
4.2. Hipóteses para a Formulação do Algoritmo	50
4.3. Convergência do Algoritmo de Merrill	53
4.4. O Algoritmo	56

CAPÍTULO	V - EXPERIÊNCIA COMPUTACIONAL	58
5.1.	Descrição do Programa	58
5.2.	Resultados Numéricos	64
5.3.	O Programa	
REFERÊNCIAS	73

INTRODUÇÃO

Os modelos de equilíbrio computáveis vêm se tornando cada vez mais importantes como instrumentos na avaliação de políticas econômicas envolvendo diversas áreas de interesse, como impostos, tarifas, comércio exterior, fluxos de capital e energia. Eles são necessários quando, além da eficiência, é preciso buscar-se aspectos igualmente relevantes de equidade através da descentralização de preços.

O problema numérico que está por trás desse problema econômico, segundo a formulação aqui estudada, tem a ver com a solução de sistemas de equações não-lineares, para o que existem métodos numéricos disponíveis. No entanto, estes são de caráter local, no sentido de precisarem uma boa aproximação inicial à verdadeira solução para garantirem convergência. Métodos globais foram desenvolvidos a partir do trabalho de Scarf, nos anos sessenta, o primeiro a fornecer uma solução numérica para o problema de equilíbrio econômico.

A formulação apresentada por Scarf foi extensivamente desenvolvida, permitindo sua aplicação, em geral, ao problema de achar um zero ou ponto fixo de uma correspondência (aplicação ponto-conjunto ou multiaplicação). A idéia básica dos métodos assim disponíveis, citados na literatura como simpliciais, homotópicos ou de acompanhamento de trajetórias é simples: do ponto do domínio no

que estamos, construir uma trajetória linear por partes até a solução.

Os teoremas de ponto fixo de Brouwer (1912) e Kakutani (1941) determinam as condições para existência de pontos fixos, se bem que por não serem construtivos, não permitem o cálculo destes. Scarf (1967), baseado no trabalho de Lemke e Howson (1964), é que apresenta uma prova construtiva dos teoremas de ponto fixo, permitindo, assim, uma formulação algorítmica para seu cálculo. Kuhn (1968) e o próprio Scarf e Hansen (1969) reformularam o trabalho anterior em termos de partições simplícias do simplexo unitário. Merrill (1971) melhorou a eficiência computacional deste algoritmo e Eaves (1972) faz a relação com o método homotópico e o acompanhamento de trajetórias lineares por partes. Aqui estudamos o algoritmo de Merrill, através de uma formulação homotópica. No entanto, deixamos de incluir aspectos sobre aceleração destes métodos, com uso de técnicas quase-Newton, como discutido por Saigal (1977), Saigal e Todd (1978) e Awoniyi e Todd (1983).

No primeiro capítulo apresentamos um modelo em uma economia de trocas e o uso do teorema de Brouwer na procura dos preços de equilíbrio. No segundo capítulo, agregamos ao modelo de trocas um sistema de produção por atividades lineares e apresentamos o uso do teorema de Kakutani para achar o equilíbrio desse modelo. No terceiro capítulo, introduzimos a formulação homotópica na busca de pontos fixos de uma correspondência. No quarto, apresentamos

o algoritmo de Merrill do ponto de vista homotópico e no último, algumas experiências computacionais do que fizemos.

CAPÍTULO I

EQUILÍBRIO EM UMA ECONOMIA DE TROCAS

Brouwer (1912) provou que uma função contínua de um conjunto convexo compacto em si mesmo tem no mínimo um ponto fixo. O principal propósito do nosso trabalho diz respeito ao cálculo de pontos fixos. Scarf (1967), baseado no lema combinatório de Sperner, estabeleceu uma prova construtiva do teorema de Brouwer, formulando um algoritmo para a aproximação de pontos fixos de funções contínuas e sua aplicação ao problema do cálculo de equilíbrios econômicos.

1.1. O TEOREMA DE BROUWER

TEOREMA 1.1. (Brouwer). Seja $C \subset \mathbb{R}^n$ um conjunto convexo compacto com $\text{int } C \neq \emptyset$ e seja $f: C \rightarrow C$ contínua. Então, f tem um ponto fixo, i.e., existe $x^* \in C$ tal que $f(x^*) = x^*$.

Mostraremos abaixo que é suficiente provar o teorema de Brouwer para o caso em que C é o simplexo unitário de \mathbb{R}^n .

DEFINIÇÃO 1.1. O simplexo unitário S^n é a envoltória convexa de e^0, e^1, \dots, e^n , onde e^{i-1} corresponde ao i -ésimo vetor unitário do \mathbb{R}^{n+1} , i.e.,

$$S^n = \{x \in \mathbb{R}_+^{n+1} / e^T x = 1\},$$

onde $e^T = (1, 1, \dots, 1)$.

Os pontos e^0, e^1, \dots, e^n são chamados vértices de S^n . Para $i = 0, 1, \dots, n$, S_i^n denota a face de S^n oposta ao vértice e^i , i.e.,

$$S_i^n = \{x \in S^n / x_i = 0\}.$$

A fronteira de S^n é $\partial S^n = \bigcup_{i=0}^n S_i^n$.

Se C é um conjunto convexo compacto, podemos embutir C num simplexo maior S (ver Figura 1) e definir uma função $g : S \rightarrow S$ da seguinte forma:

$$g(x) = \begin{cases} f(x), & x \in C \\ f(\text{Proj}_C x), & x \notin C \end{cases}$$

onde $\text{Proj}_C x$ é a projeção de x sobre C , na distância euclidiana.

Como a $\text{Proj}_C x$ é contínua, então g é contínua e, pelo teorema de Brouwer, existe $x^* \in S$ em $g(x^*) = x^*$. Como $g(x^*) \in C$, então $x^* \in C$ e, portanto, $x^* = f(x^*)$, $x^* \in C$.

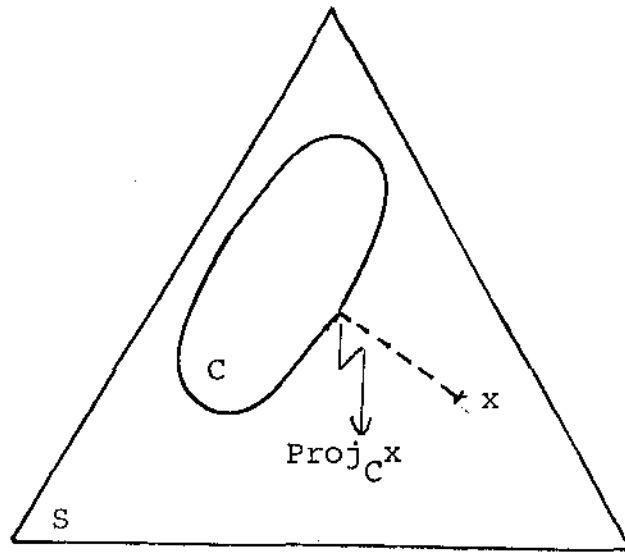


Figura 1 - Embutimento de um conjunto convexo compacto $C \subset \mathbb{R}^n$ em um simplexo S .

1.2. LEMA DE SPERNER

Para provar o teorema de Brouwer para S^n , o reduziremos ao lema de Sperner. Faremos isto primeiro estabelecendo o lema de KNASTER-KURATOWSKI-MAZURKIEWICZ (K-K-M).

LEMA 1.2. (K-K-M). Seja C_i , $i = 0, 1, \dots, n$ uma família de conjuntos fechados em S^n que satisfazem as seguintes condições:

$$(i) \quad S^n = \bigcup_{i=0}^n C_i$$

$$(ii) \quad \emptyset \neq I \subset \{0, \dots, n\} \text{ e } J = \{0, \dots, n\} \setminus I,$$

$$\Rightarrow \bigcap_{i \in I} S_i^n \subset \bigcup_{j \in J} C_j.$$

Então, $\bigcap_{i=0}^n C_i \neq \emptyset$.

Antes de provar o lema de K-K-M, veremos que este implica no teorema de Brouwer.

PROPOSIÇÃO 1.3. O lema de K-K-M implica no teorema de Brouwer para S^n .

PROVA. Seja $f: S^n \rightarrow S^n$ contínua. Para $i = 0, 1, \dots, n$ seja

$$C'_i = \{x \in S^n / f_i(x) \leq x_i, x_i > 0\} \text{ e } C_i = \overline{C'_i}.$$

Os C_i satisfazem as hipóteses do lema de K-K-M. Como mostrado a seguir.

Supomos que existe $x \in S^n$, tal que se $x \notin C_i, i = 0, 1, \dots, n$, então $f_i(x) > x_i$, para $i \in I = \{i \in \{0, 1, \dots, n\} \mid x_i > 0\}$.

Então,

$$1 = e^T x = \sum_{i \in I} x_i < \sum_{i \in I} f_i(x) \leq e^T f(x) = 1,$$

o que é uma contradição.

Portanto, dado que $C_i \subset S^n, i = 0, 1, \dots, n$, temos que $S^n = \bigcup_{i=0}^n C_i$ e a condição (i) é satisfeita.

Por outro lado, se $x \in \bigcap_{i \in I} S_i^n, x \notin C'_i, i \in I$, então, para

$J = \{0, 1, \dots, n\} \setminus I$, $x \in \bigcup_{j \in J} C_j^i \subset \bigcup_{j \in J} C_j$, e a condição (ii) também é satisfeita.

Como o lema de K-K-M é satisfeito, então existe $x^* \in \bigcap_{i=0}^n C_i$.

Dado que f é contínua e $x^* \in \overline{C_i^1}$, $i = 0, 1, \dots, n$, temos que $f_i(x^*) \leq x_i^*$, $i = 0, 1, \dots, n$. Como $1 = e^T x^* = e^T f(x^*)$, então $f_i(x^*) = x_i^*$ e, portanto, x^* é um ponto fixo de f .

Para provar o lema de K-K-M, precisamos introduzir o conceito de triangulação.

Um j -simplexo fechado é a envoltória convexa de $j+1$ pontos afim-independentes em \mathbb{R}^k . Os pontos são chamados vértices do j -simplexo. Um j -simplexo aberto é o interior relativo de um j -simplexo fechado (observar que S^n é um n -simplexo fechado).

O j -simplexo aberto (fechado) gerado pelos pontos v^0, v^1, \dots, v^j é denotado por:

$$\langle v^0, v^1, \dots, v^j \rangle \quad ([v^0, v^1, \dots, v^j]).$$

Uma face de um simplexo σ é um simplexo cujos vértices são vértices de σ (S_i^n é uma face fechada de S^n , para $i = 0, 1, \dots, n$).

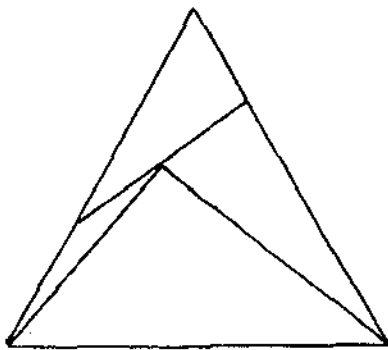
Dois simplexos são ditos incidentes se um deles é uma face do outro. Dois j -simplexos são ditos adjacentes se tem um $(j-1)$ -simplexo como face comum.

Uma triangulação T de S^n é uma coleção finita de

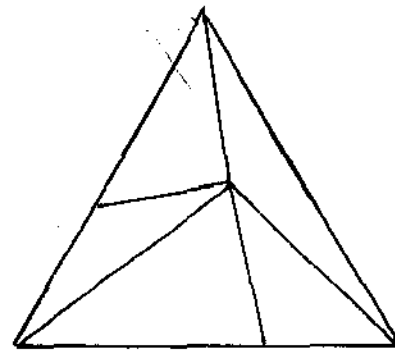
n -simplexos abertos, tal que os n -simplexos abertos, com todas suas faces abertas, formam uma partição de S^n , i.e., S^n é sua união disjunta. Esta definição em termos de n -simplexos fechado é:

- (i) os n -simplexos fechados cobrem S^n ; e
- (ii) a intercessão de dois n -simplexos fechados é vazia ou uma face comum.

EXEMPLOS ($n = 2$):



Não é triangulação



Triangulação

Chamaremos a um vértice de um simplexo de uma triangulação T , um vértice de T . Daqui em diante, um simplexo será considerado como um simplexo aberto.

Utilizaremos a notação T^j para nos referir a coleção de j -simplexos que são faces de simplexos de T (então $T^n = T$). Os vértices de T serão denotados por T^0 e $T^+ = \bigcup_{j=0}^n T^j$.

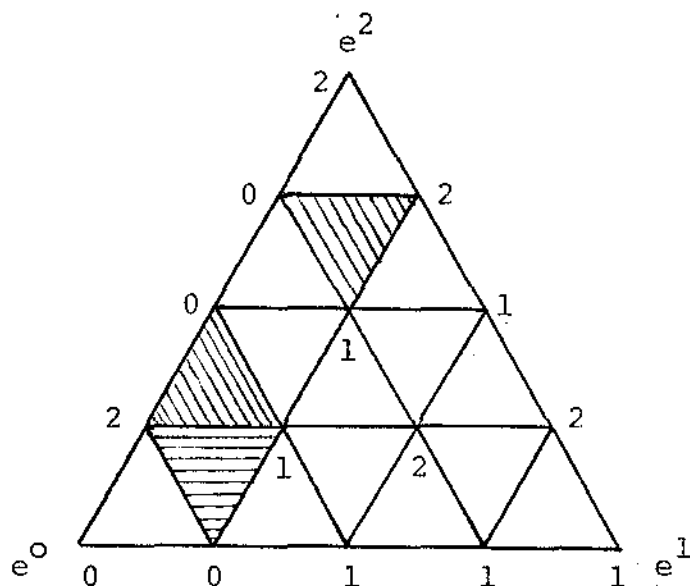
Algumas propriedades das triangulações S^n são:

- (a) Se T é uma triangulação de S^n e τ é um $(n-1)$ -simplexo que é uma face de um n -simplexo em T , então acontece (i) ou (ii):
- (i) $\tau \subset \partial S^n$ e τ é uma face de um único simplexo $\sigma \in T$.
 - (ii) $\tau \not\subset \partial S^n$ e τ é uma face comum a dois simplexos em T .
- (b) Existem triangulações de S^n de tamanho arbitrário (tamanho de $T = \text{tam } T = \sup_{\sigma \in T} \text{diam } \sigma$).
- (c) Seja T uma triangulação de S^n e $i \in \{0, 1, \dots, n\}$. Então, a coleção T' de $(n-1)$ -simplexos, que são faces de simplexos em T e situam-se em S_i^n , é uma triangulação de S_i^n .

Agora podemos estabelecer o lema de Sperner.

LEMA 1.4. (Sperner). Seja T uma triangulação de S^n com cada vértice de T rotulado com um inteiro em $\{0, 1, \dots, n\}$, tal que nenhum vértice em S_i^n é rotulado com i (tal rotulação é chamada admissível). Então existe um simplexo em T cujos vértices estão rotulados com todos os inteiros $0, 1, \dots, n$ (simplexo completamente rotulado).

Exemplo (ver figura a seguir):



O lema 1.4 é a forma fraca do lema de Sperner. A forma forte assegura que existe um número ímpar de simplexes completamente rotulado.

PROPOSIÇÃO 1.5. O lema de Sperner (forma fraca) implica o lema de K-K-M.

PROVA. Sejam C_i , $i = 0, 1, \dots, n$, conjuntos fechados que satisfazem as hipóteses do lema de K-K-M e $\{T_k\}_{k \in \mathbb{N}}$ uma sequência de triangulações de S^n com $\text{tam } T_k \rightarrow 0$. Para cada k , rotular os vértices de T_k com $i = \min\{j/y \in C_j, y \notin S_j^n\}$. Esta é uma rotulação admissível. Se o lema de Sperner é verdadeiro, então existe um

simplexo completamente rotulado $\sigma_k \in T_k$. Seja $\sigma_k = \langle v^{k0}, v^{k1}, \dots, v^{kn} \rangle$ com rótulo v^{ki} igual a i , então $v^{ki} \in C_i$, $i = 0, 1, \dots, n$. Como $\{v^{k0}\}_{k \in \mathbb{N}} \subset S^n$, existe uma subsequência convergente. Vamos supor que seja a mesma $\{v^{k0}\}_{k \in \mathbb{N}}$, então $v^{k0} \rightarrow x^* \in S^n$. Dado que $\text{tam } T_k \rightarrow 0$, temos que $\lim_{k \rightarrow \infty} v^{ki} = x^*$, para $i = 0, 1, \dots, n$. Como C_i fechado, $x^* \in C_i$, $i = 0, 1, \dots, n$, então $\bigcap_{i=0}^n C_i \neq \emptyset$. ■

O seguinte lema estabelece que os pontos do simplexo completamente rotulado, determinado pelo lema de Sperner, se constituem de aproximações de um ponto fixo.

LEMA 1.6. Seja T uma triangulação de S^n com $\text{tam}_\infty T \leq \delta$. Seja $f : S^n \rightarrow S^n$ tal que $\|x - z\|_\infty < \delta$ implica $\|f(x) - f(z)\|_\infty \leq \varepsilon$. Rotular cada vértice v de T por $i = \min\{j/f_j(v) \leq v_j, v_j > 0\}$. Então se σ é um simplexo completamente rotulado de T e $x^* \in \sigma$, $\|f(x^*) - x^*\|_\infty \leq n(\varepsilon + \delta)$.

PROVA. Seja $\sigma = \langle v^0, v^1, \dots, v^n \rangle$, onde rótulo de v^i é i . Então, para cada $i = 0, 1, \dots, n$, temos:

$$f_i(x^*) - x_i^* = (f_i(x^*) - f_i(v^i)) + (f_i(v^i) - v_i^i) + (v_i^i - x_i^*) .$$

Donde, pelas hipóteses,

$$f_i(x^*) - x_i^* \leq \varepsilon + \delta .$$

Também, como $e^T f(x^*) = e^T x^* = 1$, temos

$$f_i(x^*) - x_i^* = - \sum_{j \neq i} (f_j(x^*) - x_j^*) \geq -n(\varepsilon + \delta).$$

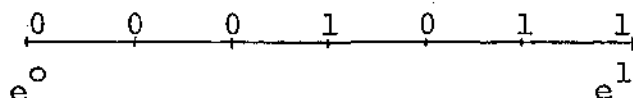
Então,

$$\|f_i(x^*) - x_i^*\|_\infty \leq n(\varepsilon + \delta). \quad \blacksquare$$

Agora vamos apresentar a prova do Lema de Sperner, que finalmente nos permitirá estabelecer o teorema de Brouwer.

PROVA DO LEMA DE SPERNER. Procederemos por indução sobre n para provar a forma forte.

$n = 1$: Aqui já formulamos o processo indutivo ao apresentar para o caso geral. Temos o seguinte desenho



Consideremos as incidências de vértices $((n-1)$ -simplexos) rotulados com 0, com os segmentos $(n$ -simplexos) da triangulação. Estas incidências podem-se contar de duas formas:

- (i) Contemos as incidências para cada segmento com no mínimo um vértice rotulado com 0. Estes segmentos dividem-se em dois conjuntos:

A - conjunto de segmentos com um vértice 0 e outro 1 ; e

B - conjunto de segmentos com ambos os vértices 0.

Cada segmento em A contribui uma vez na contagem das incidências e cada segmento em B contribui duas. Então a contagem total é $|A| + 2|B|$.

(ii) Contemos as incidências para cada vértice rotulado com 0. Tais vértices também dividem-se em dois conjuntos:

C - o conjunto que consiste só de e^0 , o único vértice de ∂S^1 rotulado com 0 ; e

D - o conjunto de vértices internos ($\notin \partial S^1$) rotulados com 0.

Vemos que o vértice em C contribui uma vez na contagem e os vértices em D duas. Então a contagem é $1 + 2|D|$.

Como $|A| + 2|B| = 1 + 2|D| \Rightarrow |A|$ é ímpar.

Passo indutivo: Supor que a forma forte do lema de Sperner é válida para dimensão $n-1$, e seja T uma triangulação de S^n com os seus vértices rotulados admissivelmente. Seja H a coleção de $(n-1)$ -simplexos que são faces de simplexos em T e cujos vértices tem os rótulos $0, 1, \dots, n-1$. Novamente contaremos as incidências de T e H .

(i) Contemos as incidências para cada $\sigma \in T$ cujos vértices estão rotulados com $0, 1, \dots, n-1$. Tais simplexos caem em dois

conjuntos:

A - o conjunto de simplexos de T completamente rotulados.

B - o conjunto de simplexos de T cujos v rtices tem todos os r tulos $0, 1, \dots, n-1$, mas n o n .

Cada simplexo em A tem s o uma face em H , enquanto que cada simplexo em B tem duas. Assim, a contagem total de incid ncias   $|A| + 2|B|$.

(ii) Contemos as incid ncias para cada $\tau \in H$. H   particionado em dois conjuntos:

C - o conjunto de $\tau \in H$ com $\tau \subset \partial S^n$.

D - o conjunto de $\tau \in H$ com $\tau \not\subset \partial S^n$.

Pelas regras de rotula  o admiss vel, cada $\tau \in C$ cai em S_n^n e contribui uma vez, enquanto que $\tau \in D$ contribui duas vezes nas incid ncias de T e H . Ent o a contagem total   $|C| + 2|D|$ e temos que $|A| + 2|B| = |C| + 2|D|$.

Para provar que $|A|$    mpar, devemos provar que $|C|$    mpar. Considerando a cole  o T' de $(n-1)$ -simplexos que s o faces de T e caem em S_n^n , T' triangulariza S_n^n . Como S_n^n   homeom rfico a S^{n-1} (  s o eliminar ou aumentar a  ltima coordenada zero) e T'   rotulado admissivelmente, ent o, pela hip tese indutiva, existe um n mero  mpar de simplexos em T' completamente rotulados, mas esses simplexos s o aqueles de C. Ent o, $|A|$    mpar. ■

1.3. APLICAÇÃO DO TEOREMA DE BROUWER AO PROBLEMA DE EQUILÍBRIO ECONÔMICO

Scarf (1967), baseado no lema de Sperner, construiu um algoritmo para achar um simplexo completamente rotulado para o caso do problema de equilíbrio econômico.

Aqui apresentamos a idéia de Scarf de como utilizar o teorema de Brouwer no caso de uma economia de trocas.

1.3.1. O PROBLEMA DE EQUILÍBRIO EM UMA ECONOMIA DE TROCAS

Consideramos aqui um modelo bastante simplificado. As suposições feitas nesta seção serão em parte relaxadas quando o modelo para uma economia com produção for discutido no Capítulo II. Para a discussão econômica das hipóteses feitas, ver Varian (1978).

Supomos que existem m consumidores, cada um dos quais enfrenta o problema de escolher certas cestas de consumo determinadas por vetores de diferentes bens $x \in \mathbb{R}_+^{n+1}$ ($n+1$ bens). Cada consumidor i possui certa quantidade de bens inicialmente (dotes), formando um vetor $w^i \in \mathbb{R}_+^{n+1}$, $i = 1, \dots, m$, que está disposto a trocar na procura de uma cesta ótima. Para isto, supomos também a existência de uma relação de preferência \succeq_i , que ordena as cestas $(x \succeq_i y, \text{ se o consumidor } i \text{ prefere a cesta } x \text{ quando comparada com } y_k \text{ ou é indiferente entre elas})$. Sobre a relação de preferência é suposto que para toda $x, y, z \in \mathbb{R}_+^{n+1}$:

$$(a) \quad x \succeq_i y \text{ ou } y \succeq_i x ;$$

$$(b) \quad x \succeq_i y/x$$

$$(c) \quad x \succeq_i y, y \succeq_i z \Rightarrow x \succeq_i z .$$

É claro que podemos definir a relação de preferência estrita \succ_i e a de indiferença \sim_i .

$$(d) \quad \text{Os conjuntos } U_x = \{y \in \mathbb{R}_+^{n+1} / y \succeq_i x\}, \quad Lx = \{y \in \mathbb{R}_+^{n+1} / x \succeq_i y\} \text{ são fechados e } U_x \text{ convexo.}$$

Segue-se que $\{y \in \mathbb{R}_+^{n+1} / x \succ_i y\}$ e $\{y \in \mathbb{R}_+^{n+1} / y \succ_i x\}$ são abertos.

Na teoria econômica clássica, com frequência o comportamento do consumidor i é resumido através da função utilidade, que é uma função contínua $u_i : \mathbb{R}_+^{n+1} \rightarrow \mathbb{R}$, tal que $x \succ_i y$ se e somente se $u_i(x) > u_i(y)$. Pode-se provar que sob as hipóteses (a) - (d) para \succeq_i , existe uma função utilidade que a representa (ver Debreu (1964)). Então o problema do consumidor consiste em, dado um vetor de preços não-nulo $p \in \mathbb{R}_+^{n+1}$, determinar a cesta que maximiza a sua utilidade, levando em conta que seu conjunto de cestas disponíveis é

$$B^i(p) = \{x \in \mathbb{R}_+^{n+1} / p^T x \leq p^T w^i\}.$$

Formalmente o problema é

$$\max u_i(x)$$

$$x \in B^i(p),$$

para $p \in \mathbb{R}_+^{n+1}$, $p \neq 0$, dado.

Se considerarmos que $p > 0$ e $p^T w^i > 0$, o conjunto $B^i(p)$ é compacto e, portanto, o consumidor terá uma cesta ótima $d^i(p)$, que é a demanda do consumidor i sob o vetor de preços p . Faremos também a hipótese simplificadora de que $d^i(p)$ é única para todo $p \geq 0$. É claro que $d^i(\lambda p) = d^i(p)$, para $\lambda > 0$. Portanto, podemos normalizar os preços para que $\sum_{j=0}^n p_j = 1$, i.e., $p \in S^n$. Então, $d^i : S^n \rightarrow \mathbb{R}_+^{n+1}$, e supomos que é contínua. Além disso, se supomos não-saciedade local sobre \mathbb{R}_+^{n+1} , temos que o consumidor i gasta toda sua receita na procura da cesta ótima, i.e., $p^T d^i(p) = p^T w^i$, para todo $p \in S^n$ (esta propriedade é conhecida como a lei de Walras). Então, se $d(p) = \sum_{i=1}^m d^i(p)$ é a demanda agregada e $w = \sum_{i=1}^m w^i$ é o vetor de recursos iniciais totais (dote inicial do mercado), obtemos que $p^T d(p) = p^T w$. Seja $e(p) = d(p) \cdot -w$ o vetor demanda em excesso, então $p^T e(p) = 0$, i.e., o valor da demanda em excesso é zero para todo $p \in S^n$. Como $d(p)$ continua, $e : S^n \rightarrow \mathbb{R}_+^{n+1}$ é contínua.

Um vetor de preços não seria de grande utilidade se não transmitisse os interesses de todos os consumidores. Para que todas as trocas sejam factíveis, é necessário e suficiente que

$e(p)$ seja não-positivo.

DEFINIÇÃO 3.1. $p^* \in S^n$ é um preço de equilíbrio para uma economia de trocas se $e(p^*) \leq 0$.

Observar que se a lei de Walras é satisfeita e p^* é preço de equilíbrio, então $e_i(p^*) \leq 0$, $i = 0, 1, \dots, n$, com igualdade se $p_i^* > 0$. Então, todos os mercados estão balanceados, exceto, possivelmente, certos bens, chamados livres, que podem estar com oferta em excesso.

Agora provaremos que sob as hipóteses feitas, um preço de equilíbrio p^* sempre existe.

TEOREMA 1.7. Seja $e : S^n \rightarrow \mathbb{R}^{n+1}$ contínua, tal que $p^T e(p) = 0$, para todo $p \in S^n$. Então existe $p^* \in S^n$, com $e(p^*) \leq 0$.

PROVA. Definiremos uma função contínua cujos pontos fixos são preços de equilíbrio. Para construir essa função, vamos aplicar a seguinte idéia para os vetores de preços que não representam equilíbrio: aumentar p_i , se $e_i(p) > 0$ (bem i com demanda em excesso) e diminuir p_i se $e_i(p) < 0$ (bem i com oferta em excesso). Esse método de "tâtonnement", sugerido por Walras, poderia ser bastante instável, como mostrado por Scarf (1967), mas o usaremos como base da nossa função.

Consideremos h levando p em $p + \lambda e(p)$, $\lambda > 0$. Como h não leva S^n em S^n , ela deve ser modificada. Seja $h^+ : S^n \rightarrow \mathbb{R}_+^{n+1}$,

onde $h^+(p) = (h_0^+(p), \dots, h_n^+(p))^T$, com $h_i^+(p) = \max\{0, h_i(p)\}$. Para $p \in S^n$, $h^+(p) \geq 0$, mas as suas coordenadas poderiam não somar 1. Então, definimos $f: S^n \rightarrow S^n$, como $f(p) = h^+(p)/e^T h^+(p)$. A função f está bem definida, pois

$$\begin{aligned} e^T h^+(p) = 0 &\Rightarrow h^+(p) = 0 \Rightarrow h(p) \leq 0 \Rightarrow p^T h(p) \leq 0 \\ &\Rightarrow p^T p + \lambda p^T e(p) = p^T p \leq 0, \end{aligned}$$

o que contradiz o fato de

$$p \in S^n.$$

Como $e(p)$ é contínua, h, h^+ e $e^T h^+$ são contínuas. E a partir de $e^T h^+$ ser contínua e positiva no conjunto compacto S^n , é limitada por um escalar positivo; então, f é bem definida.

O teorema de Brouwer garante um ponto fixo p^* de f . Assim, $h^+(p^*) = \mu p^*$, $\mu > 0$. Seja $I = \{i/p_i^* > 0\}$. Então, temos

$$(a) \forall i \in I, \mu p_i^* = h_i^+(p^*) = h_i(p^*) = p_i^* + \lambda e_i(p^*); \text{ e}$$

$$(b) \forall j \notin I, 0 = \mu p_j^* = h_j^+(p^*) \geq h_j(p^*) = \lambda e_j(p^*).$$

Se $i \in I$, $e_i(p^*) = \rho p_i^*$, com $\rho = (\mu - 1)/\lambda$. Então a lei de Walras fornece

$$0 = p^{*T} e(p^*) = \sum_I p_i^* e_i(p^*) = \rho \sum_I p_i^* p_i^* = \rho \|p^*\|_2^2$$

$$p^* \in S^n \Rightarrow \rho = 0 \Rightarrow e_i(p^*) = 0, i \in I \quad (\alpha)$$

Portanto, (a), (b) e (α) implicam que $e(p^*) \leq 0$. ■

CAPÍTULO II

EQUILÍBRIO EM UMA ECONOMIA COM PRODUÇÃO

2.1. SEMICONTINUIDADE SUPERIOR

O estudo do equilíbrio econômico, quando a produção é agregada ao modelo, requer a extensão do teorema de Brouwer para se determinar aproximações ao ponto fixo resultante. É necessário relaxar o conceito de função (trabalhar com o conceito de correspondência) e o conceito de continuidade de uma função (definir o conceito de semicontinuidade superior de uma correspondência).

DEFINIÇÃO 2.1. Seja $P(\mathbb{R}^P)$ o conjunto potência de \mathbb{R}^P , $C \subset \mathbb{R}^m$ e $F : C \rightarrow P(\mathbb{R}^P)$ uma correspondência (aplicação ponto-conjunto). Dizemos que F é semicontínua superior (s.c.s) se:

- (i) para todo $x \in C$, $F(x)$ é compacto; e
- (ii) para todo $x \in C$, dado $\varepsilon > 0$, arbitrário, existe $\delta > 0 / z \in B(x, \varepsilon) \cap C$, então $F(z) \subset B(F(x), \varepsilon)$.

A propriedade mais importante de uma correspondência s.c.s. é a seguinte:

LEMA 2.2. Se $F : C \rightarrow P(\mathbb{R}^P)$ é s.c.s., $\{x^k\} \subset C$ é tal que $x^k \rightarrow x^*$,

e $\{y^k\}$, com $y^k \in F(x^k)$, converge a y^* , então $y^* \in F(x^*)$.

PROVA. Dado qualquer um $\varepsilon > 0$, temos k tal que $\|y^k - y^*\| \leq \varepsilon/2$ e $F(x^k) \subset B(F(x^*), \varepsilon/2)$. Então $y^* \in B(F(x^*), \varepsilon)$, para todo $\varepsilon > 0$. Como $F(x^*)$ compacto e $y^k \rightarrow y^*$, temos que $y^* \in F(x^*)$. ■

Algumas propriedades importantes a respeito das correspondências s.c.s. são dadas no seguinte teorema.

TEOREMA 2.3. Aqui consideramos, a menos que seja especificado em contrário, que as correspondências são de $C \subset \mathbb{R}^m$ a $P(\mathbb{R}^p)$.

- (a) Se F é s.c.s. e $D \subset C$ é compacto, então $F(D)$ é compacto.
- (b) Seja F s.c.s. e $G : D \subset \mathbb{R}^p \rightarrow P(\mathbb{R}^l)$ s.c.s. com $F(C) \subset D$. Então $H = GF : C \rightarrow P(\mathbb{R}^l)$ é s.c.s., onde $H(x) = G(F(x))$.
- (c) Seja F_i s.c.s., $i = 1, 2, \dots, k$ e $F : C \rightarrow P(\mathbb{R}^p)$ definida por $F(x) = \cup F_i(x)$. Então F é s.c.s.
- (e) Seja F s.c.s. e $C \subset D$ fechado. Então $G : D \rightarrow P(\mathbb{R}^p)$, onde $G(x) = F(x)$ se $x \in C$, e \emptyset se $x \in D \setminus C$, é s.c.s.
- (f) Se F é s.c.s., então $\text{conv } F : C \rightarrow P(\mathbb{R}^p)$, onde $(\text{conv } F)(x) = \text{conv}(F(x))$, é s.c.s.

PROVA. ver Tood (1976).

2.2. TEOREMA DE KAKUTANI.

DEFINIÇÃO 2.4. Seja $C \subset \mathbb{R}^m$ convexo, com $\dim C = n$, e T uma triangulação de C . Seja $F : C \rightarrow P(\mathbb{R}^P)$ uma correspondência cujos valores são não-vazios. Para cada vértice v de T , escolher $f(v) \in F(v)$. Cada $x \in C$ pertence a um único simplexo $\langle v^0, \dots, v^j \rangle \in T^+$, então

$$x = \sum_{i=0}^n \lambda_i v^i, \quad \lambda_i \geq 0, \quad \sum_{i=0}^n \lambda_i = 1, \quad \text{onde os pares } \lambda_i, v^i \text{ são únicos para } \lambda_i > 0.$$

Seja $f(x) = \sum_{i=0}^n \lambda_i f(v^i)$. Então a função $f : C \rightarrow \mathbb{R}^P$ está bem definida. Esta função f é chamada de *aproximação linear por partes* (ALP) de F com respeito a T . (Pode-se provar que f é contínua).

Denotaremos C^* como a coleção de todos os conjuntos convexos não vazios de C .

TEOREMA 2.5 (KAKUTANI). Seja $C \subset \mathbb{R}^m$ convexo e compacto e $F : C^* \rightarrow C^*$ s.c.s. Então F tem um ponto fixo x^* , i.e., $x^* \in F(x^*)$.

PROVA. A prova apresentada encontra-se em Eaves (1971). Se $\dim C = n < m$, podemos aplicar homeomorficamente a envoltória afim de C sobre \mathbb{R}^n . Podemos, portanto, supor sem perda de generalidade que $m = n$.

Seja $c \in \text{int } C$. Como C compacto, podemos embuti-lo em um n -simplexo fechado S . Extendemos F a S da seguinte forma:

$$F_1(x) = \begin{cases} F(x) & , x \in C \\ \emptyset & , x \in S \setminus C \end{cases}$$

Seja

$$F_2(x) = \begin{cases} \{c\} & , x \in S \setminus \text{int } C \\ \emptyset & , x \in \text{int } C \end{cases}$$

Definimos agora

$$F_0(x) = \text{conv}(F_1(x) \cup F_2(x)).$$

Pelo teorema 2.3, (e), (c) e (f), $F_0 : S \rightarrow S^*$ é s.c.s. Observar que se x^* é ponto fixo de F_0 , então $x^* \in C$ (do contrário $x^* \in F_0(x^*) = \{c\} \subset C$, uma contradição).

Também acontece que x^* é ponto fixo de F :

Se $x^* \in \text{int } C$, então $x^* \in F_0(x^*) = \text{conv}(F_1(x^*) \cup \emptyset) = F(x^*)$.

Se $x^* \in \partial C$, então $x^* \in F_0(x^*) = \text{conv}(F_1(x^*) \cup \{c\}) = \text{conv}(F(x^*) \cup \{c\})$.

Então, $\exists f \in F(x^*)$ e $\lambda \in [0,1]/x^* = \lambda c + (1 - \lambda)f$.

Se $\lambda > 0$, como $c \in \text{int } C$ e $f \in C$, então $x^* \in \text{int } C$ (contradição!), portanto $\lambda = 0$ e $x^* = f$, onde $x^* \in F(x^*)$.

Provaremos agora que F_0 tem um ponto fixo. Seja $\{T_k\}_{k \in \mathbb{N}}$

uma sequência de triangulações de S , tal que $\text{tam } T_k \rightarrow 0$. Para cada k , seja f_k uma ALP a F_0 com respeito a T_k . Como f_k é contínua, pelo teorema de Brouwer, cada f_k tem um ponto fixo x^k .

$$\text{Então } x^k = \sum_{i=0}^n \lambda_{k,i} v^{k,i} = \sum_i \lambda_{k,i} f^{k,i}, \quad \lambda_{k,i} \geq 0,$$

$$\sum_i \lambda_{k,i} = 1 \quad (\alpha)$$

onde $\langle v^{k,0}, \dots, v^{k,n} \rangle \in T_k$, $f^{k,i} \in F_0(v^{k,i})$, $i = 0, 1, \dots, n$. Quando $k \rightarrow \infty$, x^k , todos os $\lambda_{k,i}$ e os $f^{k,i}$ ficam nos conjuntos compactos C , $[0, 1]$ e C , respectivamente.

Então, existe uma subsequência (que suporemos a mesma por simplicidade de notação) tal que $x^k \rightarrow x^*$, $\lambda_{k,i} \rightarrow \lambda_i$ e $f^{k,i} \rightarrow f^i$. Dado que $\text{tam } T_k \rightarrow 0$, temos que $v^{k,i} \rightarrow x^*$, $i = 0, 1, \dots, n$. Como F_0 é s.c.s., então $f^i \in F_0(x^*)$, $i = 0, 1, \dots, n$. Tomando limites em (α) , temos

$$x^* = \sum_{i=0}^n \lambda_i f^i, \quad \lambda_i \geq 0, \quad \sum_{i=0}^n \lambda_i = 1.$$

Como $F_0(x^*)$ é convexo, $x^* \in F_0(x^*)$. ■

2.3. EXTENSÕES DO TEOREMA DE KAKUTANI

COROLÁRIO 2.6. Seja $C \subset \mathbb{R}^n$ compacto e convexo com $c \in \text{int } C$.

Seja $F : C \rightarrow \mathbb{R}^n$ s.c.s., tal que $c \in F(x)$ se $x \in \partial C$. Então,

existe $x^* \in C / x^* \in F(x^*)$.

PROVA. Como F é s.c.s., $F(C)$ é compacto e, portanto, $C' = \text{conv}(C \cup F(C))$ também é compacto. Extendemos F a $F_0 : C' \rightarrow C'^*$ usando o fato de que $c \in \text{int } C$, como no teorema de Kakutani, vemos que F_0 tem um ponto fixo x^* (teorema de Kakutani). Se $x^* \notin C$, $x^* \in F_0(x^*) = \{c\} \subset C$, contradição!. Então, $x^* \in C$ e $x^* \in F_0(x^*) = F(x^*)$. ■

COROLÁRIO 2.7. Seja $T^n = \{x \in \mathbb{R}^{n+1} / e^T x = 1\}$ e $F : S^n \rightarrow T^{n*}$ s.c.s. Supor que existe $f^i \in T^n - T^n$, $i = 0, 1, \dots, n$, tal que

$$(i) \quad x \in S_i^n \Rightarrow x + f^i \in F(x) ; \text{ e}$$

$$(ii) \quad f_j^i < 0, \text{ para } j \neq i.$$

Então existe $x^* \in S^n / x^* \in F(x^*)$.

PROVA. Seja $\tilde{S}^n = \{x \in T^n / x \geq -e\}$. Definir as correspondências F_i , $i = 1, 2, 3$, como:

$$F_1(x) = \begin{cases} F(x) & , \quad x \in S^n \\ \emptyset & , \quad x \in \tilde{S}^n \setminus S^n \end{cases}$$

$$F_2(x) = \begin{cases} \emptyset, & x \in \text{int rel } S^n \\ \{x\} + \text{conv}\{f^i/i \in I\}, & x \in \tilde{S}^n \setminus \text{int rel } S^n \\ & I = \{i/x_i = \min_j x_j\} \end{cases}$$

$$F_3(x) = \begin{cases} \emptyset, & x \in \text{int rel } \tilde{S}^n \\ \{(n+1)^{-1}e\}, & x \in \partial \tilde{S}^n. \end{cases}$$

Seja $F_0 = \text{conv}(F_1 \cup F_2 \cup F_3) : \tilde{S}^n \rightarrow T^{n*}$. Então F_0 é s.c.s. e $(n+1)^{-1}e \in F_0(x)$, para $x \in \partial \tilde{S}^n$. Pelo corolário 2.6, F_0 tem um ponto fixo $x^* \in \tilde{S}^n$.

Se $x^* \notin S^n$, seja $I = \{i/x_i^* = \min_k x_k^*\}$. Seja $z \in F_0(x^*) \subset \text{conv}\{(n+1)^{-1}e, \{x^* + f^i/i \in I\}\}$, qualquer um. Provaremos que $\sum_{j \in I} z_j > \sum_{j \in I} x_j^*$ e, portanto, $x^* \neq z$. (contradição com $x^* \in F_0(x^*)$).

Se $z = (n+1)^{-1}e$, $\sum_{j \in I} z_j > 0 > \sum_{j \in I} x_j^*$.

Se $z = x^* + f^i$, $i \in I$, então $\sum_{j \in I} z_j = \sum_{j \in I} (x_j^* + f_j^i) =$
 $= \sum_{j \in I} x_j^* - \sum_{k \in \{0,1,\dots,n\} \setminus I} f_k^i > \sum_{j \in I} x_j^*$,

por (ii) (observar que $I \neq \{0,1,\dots,n\}$, dado que $x^* \neq (n+1)^{-1}e$).

Como a desigualdade é válida para $z = (n+1)^{-1}e$ e $z = x^* + f^i$, $i \in I$, então é válida para qualquer z na sua envoltória convexa $F_0(x^*)$. Portanto, $x^* \in S^n$ e $x^* \in F_0(x^*) = F(x^*)$. ■

COROLÁRIO 2.8. Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n^*}$ s.c.s. Supor que existe $x^0 \in \mathbb{R}^n$ e $\mu > 0$, tal que se $x \notin B(x^0, \mu)$ existe $\omega \in \mathbb{R}^n$, com $\omega^T(x^0 - x) > 0$ e $\omega^T(f - x) > 0$, $\forall f \in F(x)$. Então, existe $x^* \in B(x^0, \mu)$, tal que $x^* \in F(x^*)$.

PROVA. Seja C o conjunto convexo e compacto $B(x^0, 2\mu)$, F_1 a restrição de F a C ,

$$F_2(x) = \begin{cases} \emptyset, & x \in \text{int } C \\ \{x^0\}, & x \in \partial C, \end{cases}$$

e $F_0 = \text{conv}(F_1 \cup F_2)$. Então, $F_0 : C \rightarrow \mathbb{R}^{n^*}$ é s.c.s. com $x^0 \in F_0(x)$, se $x \in \partial C$. Pelo corolário 2.6, F_0 tem um ponto fixo x^* . Supor que $x^* \notin B(x^0, \mu)$. Então, $x^* \in F_0(x^*) \subset \text{conv}\{x^0, F(x^*)\}$. Portanto, existe $f \in F(x^*)$ e $\lambda \in [0, 1]$ / $x^* = \lambda x^0 + (1 - \lambda)f$. Mas então, existe $\omega \in \mathbb{R}^n$ com $\omega^T(x^0 - x) > 0$ e $\omega^T(f - x^*) > 0$, o que implica que $\omega^T(x^* - x^*) > 0$, uma contradição. Então, $x^* \in B(x^0, \mu)$. ■

Para cálculos práticos, precisamos de uma leve modificação do corolário 2.8. O seguinte resultado é devido a Merrill.

COROLÁRIO 2.9. Seja $F : \mathbb{R}^n \rightarrow \mathbb{R}^{n*}$ s.c.s.. Supor que existe $x^0 \in \mathbb{R}^n$, $\mu > 0$ e $\delta > 0$, tal que se $x \notin B(x^0, \mu)$, $f \in F(x)$ e $z \in B(x, \delta)$, então $(f - x)^T (x^0 - z) > 0$. Então, existe $x^* \in B(x^0, \mu)$, tal que $x^* \in F(x^*)$.

PROVA. A condição implica que $w = x^0 - z$ satisfaz a hipótese do corolário 2.8. ■

2.4. APLICAÇÃO DO TEOREMA DE KAKUTANI AO PROBLEMA DE EQUILÍBRIO ECONÔMICO.

Aqui mostramos como o problema de equilíbrio geral, em uma economia walrasiana com produção, pode-se formular em termos de achar um ponto fixo de uma certa correspondência para, então, aplicando o teorema de Kakutani, resolver o problema.

2.4.1. O PROBLEMA DE EQUILÍBRIO EM UMA ECONOMIA COM PRODUÇÃO

Consideremos o modelo de economia de trocas apresentado em 1.3.1. Com respeito à demanda consideremos a seguinte relaxação: para qualquer $p \in S^n$, o i -ésimo consumidor tem um conjunto $D^i(p)$ de demandas alternativas. É fácil provar que se a relação de preferência \succeq satisfaz as hipóteses (a)-(d), dadas em 1.3.1, então dado $w \in \mathbb{R}_+^{n+1}$ fixo e $p \in \text{int rel } S^n$, sendo que $B(p) = \{x \in \mathbb{R}_+^{n+1} / p^T x \leq p^T w\}$ e $D(p) = \{x \in B(p) / x \sim y, \forall y \in B(p)\}$, acontece

que $D(p) \neq \emptyset$, convexo e compacto.

Além disso, $D: \text{int rel } S^n \rightarrow (\mathbb{R}_+^{n+1})^*$ é s.c.s. Portanto, vamos supor que $D^i: S^n \rightarrow (\mathbb{R}_+^{n+1})^*$ é s.c.s. Definindo $D(p) = \sum_{i=1}^m D^i(p)$, a correspondência demanda agregada $D: S^n \rightarrow (\mathbb{R}_+^{n+1})^*$ é s.c.s.

Supomos, também, que D satisfaz a lei de Walras: $\forall p \in S^n$, $d \in D(p)$, $p^T d = p^T \omega$.

A respeito da produção, vamos supor um modelo de atividades lineares. Seja $A = [-I, C]$ uma matriz $(n+1) \times m$. Para $1 \leq j \leq m$, a j -ésima coluna a^j de A representa um modo técnico de produção (ou atividade). Se a atividade j é operada ao nível 1, $|a_{ij}|$ unidades do bem i são fornecidos como produto (se $a_{ij} \geq 0$) ou solicitados como insumo (se $a_{ij} < 0$). As $n+1$ primeiras colunas de A correspondem a atividades de desperdício. O conjunto de todos os vetores de produção é $\{Az/z \geq 0\}$. Vamos supor que $\{Az/z \geq 0\} \cap \mathbb{R}_+^{n+1} = \{0\}$ (a produção é limitada). Equivalentemente, assumimos que existe $q > 0$ com $q^T A < 0$.

Agora podemos definir um equilíbrio como um vetor de preços e um vetor de nível de atividades, tal que a soma dos recursos iniciais e o vetor de produção (o total de oferta da economia) seja um membro do conjunto demanda destes preços (como estamos incluindo atividades de desperdício podemos deixar de considerar a oferta em excesso). Se, além disso, são só permitidas atividades

que maximizam lucros, os agentes terão incentivo para usá-las.

DEFINIÇÃO 2.10. Um equilíbrio é um par $(p^*, y^*) \in S^n \times \mathbb{R}_+^{m+1}$, tal que

$$(i) \quad \omega + Ay^* = d(p^*) \in D(p^*)$$

$$(ii) \quad p^{*T}A \leq 0.$$

Se usamos a lei de Walras, temos que $p^{*T}Ay^* = 0$, com $p^{*T}A \leq 0$. Quer dizer, nenhuma atividade faz lucro positivo e aquelas que são usadas ($y_j^* > 0$) fazem lucro zero.

TEOREMA 2.11. Se $\omega > 0$ existe um equilíbrio.

PROVA. Seja $p \in S^n$. Se p não é preço de equilíbrio, o modificaremos adequadamente:

Se $p_i = 0$, aumentaremos p_i para que não saia de S^n .

Se $p > 0$, mas $\pi^* = \max_j p^T a^j > 0$, ajustaremos p para diminuir o lucro das atividades mais rentáveis.

Se $p > 0$ e $\pi^* < 0$, ajustaremos p para diminuir a demanda.

Definimos as seguintes correspondências:

$$E_I(p) = \begin{cases} \text{conv} \{e^i / p_i = 0\}, & p \in \partial S^n \\ \emptyset, & \text{c.c.} \end{cases}$$

$$E_2(p) = \begin{cases} \text{conv}\{-a^j / p^T a^j = \pi^* = \max_k p^T a^k\} , & \pi^* \geq 0 \\ \emptyset , & \text{c.c.} \end{cases}$$

$$E_3(p) = \begin{cases} D(p) , & \pi^* < 0 \\ \emptyset , & \text{c.c.} \end{cases}$$

É claro que E_1, E_2, E_3 são s.c.s.

Seja $E = \text{conv}(E_1 \cup E_2 \cup E_3)$, então $E : S^n \rightarrow (\mathbb{R}^{n+1})^*$ é s.c.s. . Gostaríamos de aplicar p em $\{p\} + E(p)$, mas este conjunto poderia não estar em T^n . Então, devemos modificar $E(p)$ tal que se $f \in E'(p) \Rightarrow f \in T^n - T^n$.

Para qualquer $f \in E(p)$, fazemos o seguinte:

Seja $f'(f) = q_i f_i (q^T \omega / q^T f) - q_i \omega_i$, $f'(f)$ está bem definida e é contínua, pois:

Para $f = e^i$, $q^T f = q_i > 0$.

Para $f = -a^j$, $q^T f = -q^T a^j \geq \min_k (-q^T a^k) > 0$.

Para $f = d \in D(p)$, $q^T f \geq (\min_i q_i) e^T f \geq (\min_i q_i) p^T f$

$$= (\min_i q_i) p^T \omega \geq (\min_i q_i) (\min_j \omega_j) > 0.$$

Como f é combinação convexa de e^i , $-a^j$ e d , então

$$q^T f \geq \min_i \{ \min_k q_i, \min_k (-q^T a^k), (\min_k q_i) (\min_j \omega_j) \} > 0.$$

Assim, podemos definir

$$E'(p) = \{f'(f) / f \in E(p)\}.$$

Pelo teorema 2.3 (b), E' é s.c.s. Observar que $f'(f) \in T^n - T^n$ para qualquer f com $q^T f \neq 0$ e $E'(p)$ convexo. Então, $E' : S^n \rightarrow (T^n - T^n)^*$ e $F : S^n \rightarrow T^{n*}$, com $F(p) = \{p\} + E'(p)$, é s.c.s.

Agora, se $p \in S_i^n = \{x \in S^n / x_i = 0\}$, $e^i \in E(p)$, então $f'(e^i) \in E'(p)$.

Para cada $i = 0, 1, \dots, n$, definir

$$f^i = f'(e^i) = (-q_0 \omega_0, \dots, -q_i \omega_i + q^T \omega, \dots, -q_n \omega_n)^T.$$

Assim, para $p \in S_i^n$, $p + f^i \in F(p)$. Então as hipóteses do corolário 2.7 são satisfeitas e F tem um ponto fixo $p^* \in S^n$.

Então $0 \in E'(p^*)$, i.e., existe $f \in E(p^*)$ com $f = \lambda \omega$, para algum λ escalar positivo. Da definição de E , temos que

$$\lambda \omega = \sum_{i=0}^n \mu_i e^i + \sum_{j=1}^m v_j (-a^j) + \rho d \quad (\alpha)$$

$$\mu_i, v_j, \rho \geq 0, \sum \mu_i + \sum v_j + \rho = 1, d \in D(p^*).$$

Além disso, $\mu_i = 0$ a menos que $p_i^* = 0$; $v_j = 0$ a menos que $p^{*T} a^j$
 $= \max_k p^{*T} a^k = \pi^* \geq 0$; $\rho = 0$ a menos que $\pi^* \leq 0$.

Então, de (α)

$$\lambda p^{*T} \omega = 0 + (\Sigma - v_j) \pi^* + \rho p^{*T} d \quad (\beta)$$

Se $\pi^* > 0 \Rightarrow \rho = 0$ e $\lambda p^{*T} \omega < 0$, uma contradição!.

Se $\pi^* < 0 \Rightarrow v_j = 0$, $\forall j$, e

$$(\beta) \Rightarrow \lambda p^{*T} \omega = \rho p^{*T} d,$$

usando a lei de Walras, temos que $\lambda = \rho$, então

$$(\alpha) \Rightarrow \omega + \sum_{i=0}^n (\mu_i / \lambda) (-e^i) = d.$$

Portanto, $(p^*, (\mu_0 / \lambda, \dots, \mu_n / \lambda), 0, \dots, 0)$ é um equilíbrio.

Se $\pi^* = 0$, $(\alpha) \Rightarrow \lambda p^{*T} \omega = \rho p^{*T} d \Rightarrow \lambda = \rho$.

Dado que $e^i = -a^i$, $i = 0, 1, \dots, n$ e $p_i^* = 0$ implica que a^i faz máximo lucro zero ($p^{*T} a^i = 0$), podemos incorporar todos os μ_i nos v_j . Então, de (α), temos

$$\omega + \sum_{j=1}^m (v_j / \lambda) a^j = d.$$

Então, $(p^*, (v_1 / \lambda, \dots, v_m / \lambda))$ é um equilíbrio. ■

CAPÍTULO III

O PONTO DE VISTA HOMOTÓPICO NO CÁLCULO DE PONTOS FIXOS

Baseados no trabalho de Scarf, foram desenvolvidos algoritmos mais abrangentes para o cálculo de zeros de uma correspondência, levando em consideração uma homotopia convenientemente escolhida, que deforme uma função conhecida, com um único zero, para a correspondência em consideração. Neste capítulo fazemos uma discussão dos princípios básicos do ponto de vista homotópico. Para uma discussão ampla ver Broadie (1984), Eaves (1972), Garcia e Zangwill (1981), Scaramucci e Arica (1986) e Todd (1984) .

3.1. O PRINCÍPIO HOMOTÓPICO E APROXIMAÇÕES LINEARES POR PARTES

Supomos que estamos procurando um ponto fixo de uma função contínua $f: S^n \rightarrow S^n$. Escolhemos uma função linear $q: S^n \rightarrow S^n$, com um único ponto fixo x^0 , conhecido (por exemplo, $q(x) = x^0$, $\forall x \in S^n$). O algoritmo de Merrill, que discutimos no capítulo seguinte, pode ser visto como que aproxima os pontos fixos de uma função que é deformada de q a f . É conveniente, frequentemente, considerar o problema como o de achar zeros de funções. Assim, seja $g(x) = f(x) - x$, $r(x) = q(x) - x$. A função de homotopia $h: S^n \times [0,1] \rightarrow \mathbb{R}^{n+1}$, definida por

$$h(x,t) = tr(x) + (1-t)g(x)$$

transforma a função r na função g , na medida que t varia de 1 a 0. Observar que se $x^* \in S^n$ é tal que $h(x^*,0) = 0$, então $g(x^*) = 0$ e $f(x^*) = x^*$.

Sob certas condições, podemos assegurar que

$$P = \{(x,t) \in S^n \times [0,1] / h(x,t) = 0\}$$

contém uma trajetória iniciando em $(x^0,1)$ em P . A idéia do algoritmo é aproximar essa trajetória.

Para poder aproximar a trajetória de zeros precisamos do conceito de triangulação.

DEFINIÇÃO 3.1. Seja $C \subset \mathbb{R}^m$ um conjunto convexo, com $\dim(Af(C)) = n \leq m$. T é uma triangulação de C se:

- (i) T é uma coleção de n -simplexos ;
- (ii) As faces dos simplexos em T particionam o conjunto C ;
- (iii) Cada $x \in C$ tem uma vizinhança que encontra só um número finito de simplexos em T .

Dada uma triangulação T de $S^n \times [0,1]$, construímos uma aproximação linear por partes h_T a h . Isto é, para os vértices v de

T fazemos $h_T(v) = h(v)$, e estendemos h_T linearmente em cada simplex em T , tal que se $\omega \in \langle v^0, \dots, v^{n+1} \rangle$, com $\omega = \sum \lambda_i v^i$, então

$$h_T(\omega) = \sum \lambda_i h_T(v^i) .$$

As diferenças entre os algoritmos que tratam o assunto estão na consideração de diferentes triangulações T (ver Saigal e Todd (1978)). A idéia básica é que uma trajetória de zeros de h_T , começando em $(x^0, 1)$ nos levará a um ponto $(x^1, 0)$, onde x^1 é um zero aproximado de g . Então recomeçamos o algoritmo com uma triangulação T_1 , mais fina que T , e uma nova função linear $r_1(\cdot)$ com um único zero em x^1 para obter uma melhor aproximação x^2 , e assim por diante. Cada aplicação do algoritmo para conseguir um novo x^k , é chamada um *ciclo maior*.

3.2. O ACOMPANHAMENTO DE SOLUÇÕES LINEARES POR PARTES.

Aqui vamos ilustrar com um exemplo como funcionam os algoritmos acima descritos, relativamente ao acompanhamento das trajetórias lineares por partes que aproximam a verdadeira trajetória.

EXEMPLO 3.2. ($n=1$). Vamos considerar o problema de achar um zero de $f(x) = x^3 - 12$. Pegaremos como primeira aproximação da solução $x^0 = 0$ e como função artificial $r(x) = x$. A homotopia é

$$h(x,t) = tx + (1-t)(x^3 - 12), \quad 0 \leq t \leq 1,$$

e h_T é a ALP de h com respeito à triangulação T mostrada na figura 2 (essa triangulação é a que usaremos no algoritmo de Merrill e é devida a Kuhn).

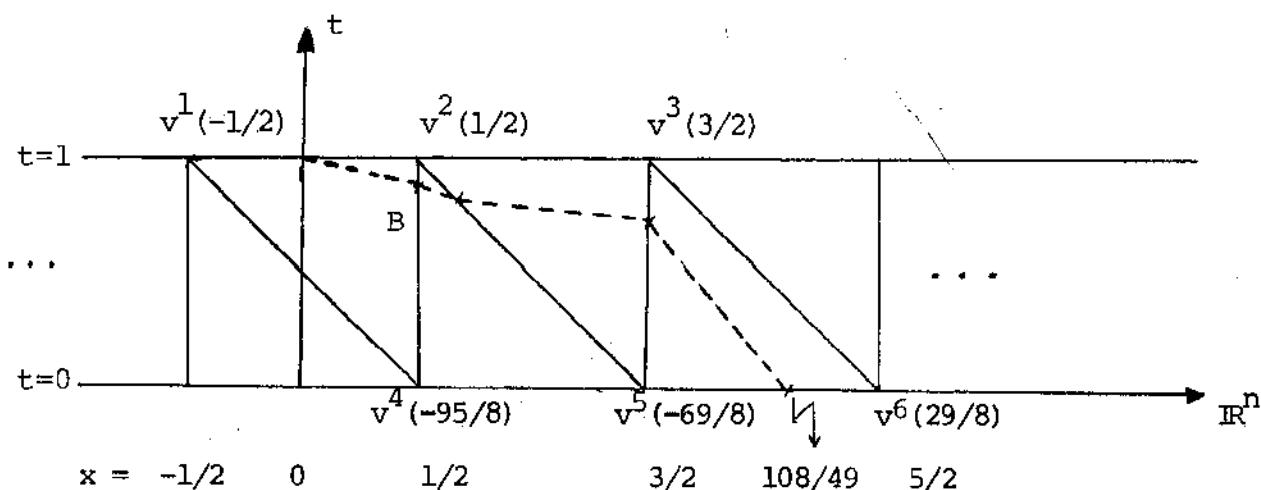


Figura 2 - Triangulação de Kuhn

Na figura 2, os valores de h_T nos vértices de T estão dados entre parênteses, a trajetória tracejada refere-se ao caminho linear por partes que aproxima P . Em um extremo, tem-se o zero conhecido desta aproximação $(x^0, t_0) = (0, 1)$; no outro, o ponto $(x^1, t_1) = (108/49, 0)$, e $x^1 = 108/49$ que é uma aproximação a solução de $f(x) = 0$. Observar que $f(x^1) = -1.2926$. O algoritmo que apresentamos consiste em seguir a trajetória linear por partes.

Dado o v rtice $v^1 = (-1/2, 1)$, o algoritmo constr i o simple_xo $\langle v^1, v^2, v^4 \rangle$ e determina os zeros de h_T nele, mediante um pivotamento similar ao M todo Simplex:

$$\text{Se } w \in \langle v^1, v^2, v^4 \rangle / h_T(w) = 0 \implies \sum \lambda_i h_T(v^i) = 0 ,$$

onde $w = \sum \lambda_i v^i$, $\sum \lambda_i = 1$, $\lambda_i \geq 0$, $i = 1, 2, 4$, que pode-se expressar como:

$$\begin{bmatrix} 1 & 1 & 1 \\ -1/2 & 1/2 & -95/8 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (3.1)$$

Expressando (3.1) como uma tabela do m todo Simplex, temos

λ_1	λ_2	λ_4	b
1	1	1	1
-1/2	1/2	-95/8	0

(3.2)

Se fazemos com que λ_1 e λ_2 estejam na base, obtemos de (3.2)

λ_1	λ_2	λ_4	b
1	0	99/8	1/2
0	1	-91/8	1/2

(3.3)

Observar que $(\lambda_1, \lambda_2, \lambda_4) = (1/2, 1/2, 0)$ corresponde à aproximação inicial de $x^0 = 0$, ponto meio do segmento $\langle v^1, v^2 \rangle$.

Fazendo com que λ_4 entre na base temos, pivotando, que λ_1 sai (i.e., a trajetória LP vai de $(0,1)$ ao simplexo $\langle v^2, v^4 \rangle$). Quer dizer que a trajetória LP vai através de $\langle v^1, v^2, v^4 \rangle$ para $\langle v^2, v^4, v^5 \rangle$, portanto, para continuar com o algoritmo devemos incluir λ_5 (que corresponde ao vértice v^5), substituindo λ_1 , e fazer com que λ_5 entre após na base.

Então, depois de atualizar $(h_T(v^5)) = (-69/8)$, temos

λ_5	λ_2	λ_4	b
73/99	0	1	4/99
26/99	1	0	95/99

(3.4)

Observar que $\lambda_5 = 0$, $\lambda_2 = 95/99$ e $\lambda_4 = 4/99$ corresponde às coordenadas baricêntricas de B em $\langle v^2, v^4, v^5 \rangle$ e $B = (x, t) = (1/2, 95/99)$. O algoritmo continua desta maneira, até que o ponto $(x^1, t_1) = (108/49, 0)$ é achado. Completando um primeiro ciclo maior do algoritmo.

A tabela 1 resume os cálculos deste ciclo maior do algoritmo. A segunda coluna, v^i , dá os vértices da base na iteração i, $(x^j, t_j)^T$, $j = 0, \dots, n$. A terceira coluna dá o vértice que entra $(\hat{x}, \hat{t})^T$. Juntas, estas colunas dão o simplex considerado na

iteração i . A base atual B^i é a matriz com colunas $(1, h(x^j, t^j))^T$, $j = 0, \dots, n$. As coordenadas baricêntricas, λ^i , estão definidas por

$$(x, t)^T = V^i \lambda^i.$$

(Ver a Tabela 1 na página seguinte).

TABELA 1 - Cálculo da Trajetória L.P. no Exemplo (3.2)

(Primeiro Ciclo Maior)

Iter	Vértices da Base V^i	Vértice que entra $(\hat{x}, \hat{t})^T$	Coluna que entra $(1, h(\hat{x}, \hat{t}))^T$	Base Atual B^i	Coord. Bari λ^i	Solução Atual $(x, t)^T$
-1	$\begin{bmatrix} -1/2 & 1/2 \\ 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1/2 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -95/8 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -1/2 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
0	$\begin{bmatrix} 1/2 & 1/2 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3/2 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 1 \\ -69/8 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -95/8 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 4/99 \\ 95/99 \end{bmatrix}$	$\begin{bmatrix} 1/2 \\ 95/99 \end{bmatrix}$
1	$\begin{bmatrix} 3/2 & 1/2 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 3/2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 3/2 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -69/8 & 1/2 \end{bmatrix}$	$\begin{bmatrix} 4/73 \\ 69/73 \end{bmatrix}$	$\begin{bmatrix} 81/146 \\ 69/73 \end{bmatrix}$
2	$\begin{bmatrix} 3/2 & 3/2 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 5/2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 29/8 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 \\ -69/8 & 3/2 \end{bmatrix}$	$\begin{bmatrix} 12/81 \\ 69/81 \end{bmatrix}$	$\begin{bmatrix} 3/2 \\ 69/81 \end{bmatrix}$
3	$\begin{bmatrix} 3/2 & 5/2 \\ 0 & 0 \end{bmatrix}$	(*)	(*)	$\begin{bmatrix} 1 & 1 \\ -69/8 & 29/8 \end{bmatrix}$	$\begin{bmatrix} 29/98 \\ 69/98 \end{bmatrix}$	$\begin{bmatrix} 108/49 \\ 0 \end{bmatrix}$

(*) não se calcula

A numeração das iterações $-1, 0, 1, 2, 3$ é devido ao fato de que na realidade o algoritmo começa trabalhar na iteração 1, pois o fato de que o primeiro simplexo é $\langle v^1, v^2, v^4 \rangle$, a primeira base é formada por v^1 e v^2 , assim como que v^4 entra na base são for necidas a priori.

3.3. MELHORAMENTO DA SOLUÇÃO APROXIMADA

Se a solução aproximada x^1 não é suficientemente boa, pode-se tentar aplicar um método quase-Newton. Pois, no final de cada ciclo maior, tem-se calculado os valores de f em cada vértice do simplex final e, portanto, uma aproximação em diferenças finitas à derivada de f é conhecida. Esta aproximação pode-se usar para um passo quase-Newton (ver Saigal (1977), Saigal e Todd (1978), Todd (1984)). Se não for convergente, então é usado novamente o método homotópico trabalhando com uma triangulação mais fina. Na figura 3 ilustramos uma tal triangulação. A trajetória LP tem vértice inicial no ponto $x^1 = 108/49 \approx 2.2040$. A nova homotopia é

$$h(x, t) = t(x - 108/49) + (1-t)(x^3 - 12)$$

O algoritmo nos fornecerá a solução aproximada $x^2 \approx 2.2622$ ($f(x^2) = -0.42162$)).

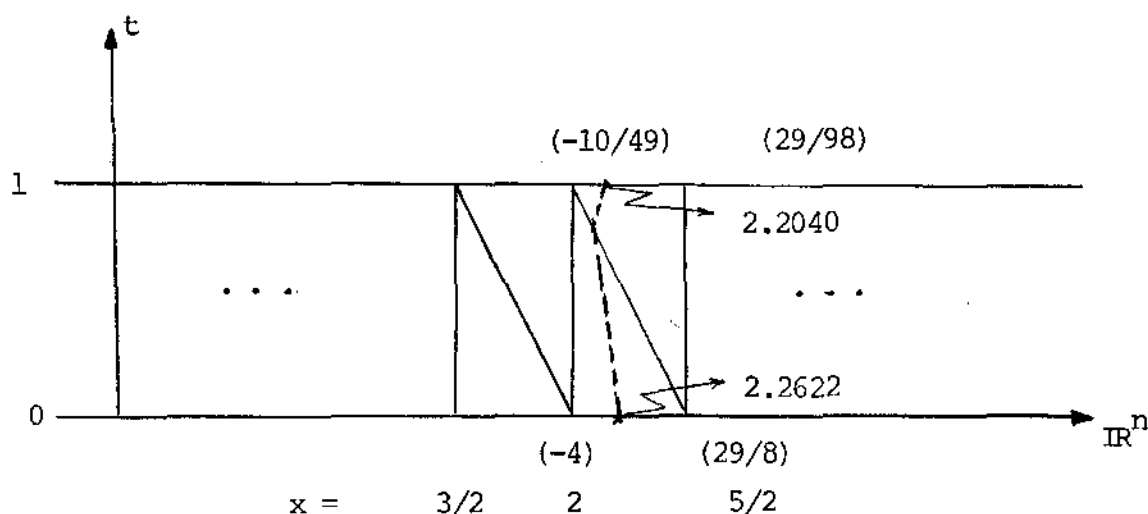


Figura 3 - Refinamento da Triangulação do Método Homotópico.

Para implementar o algoritmo, é preciso um procedimento para gerar um simplexo adjacente a outro quando eliminamos um vértice deste. O procedimento usado na triangulação de Kuhn é detalhado no capítulo seguinte.

3.4. MÉTODO HOMOTÓPICO PARA CORRESPONDÊNCIAS

Para aproximar um ponto fixo de uma correspondência s.c.s. $F : S^n \rightarrow S^{n*}$, procedemos como no caso de funções.

Definimos G por $G(x) = F(x) - \{x\} \approx \{f - x/f \in F(x)\}$ e a homotopia

$$H : S^n \times [0, 1] \rightarrow U^{n*}$$

(onde $U^n = \{x \in \mathbb{R}^{n+1} / \sum_j x_j = 0\}$).

como

$$H(x,t) = \{tg + (1-t)r(x)/g \in G(x)\}.$$

Definimos uma aproximação linear por partes a H com respeito à triangulação T , e, como antes, tracejamos uma trajetória de zeros desta aproximação.

3.4.1. REFORMULAÇÃO DO PROBLEMA DE EQUILÍBRIO ECONÔMICO (PARA UM MODELO COM PRODUÇÃO) EM TERMOS DO ZERO DE UMA CORRESPONDÊNCIA.

Esta formulação é devida a Todd (1984). Antes de apresentar a reformulação do nosso problema, vejamos como proceder para reformular o problema mais simples de uma economia de trocas. O objetivo é apresentar o problema como uma determinação de zeros em \mathbb{R}^n , na medida que muitos algoritmos são melhor formulados em \mathbb{R}^n .

Queremos achar o ponto fixo de uma função $f : S^n \rightarrow S^n$. Esse problema é equivalente a achar um zero da função $\bar{g} : S^n \rightarrow U^n$, onde $\bar{g}(x) = f(x) - x$. É simples estender \bar{g} a uma função $g : T^n \rightarrow U^n$, cujos zeros sejam exatamente os zeros de \bar{g} . Poderíamos, por exemplo, definir $g(x) = f(p(x)) - p(x)$, onde $p : T^n \rightarrow S^n$, e a projeção $p(x) = x^+ / e^T x^+$, com $x^+ = (x_0^+, \dots, x_n^+)$.

Seja, de outro lado, uma dada transformação linear não-singular $L : \mathbb{R}^n \rightarrow U^n$. Então, podemos formular o problema de achar um zero de $g : T^n \rightarrow U^n$ como o problema equivalente de achar um zero para $\tilde{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$, onde

$$g(y) = L^{-1}(g(x^1 + L(y))), \quad x^1 \in T^n.$$

Para que o condicionamento de \tilde{g} não seja pior que o de g , é conveniente escolher L como uma transformação ortogonal. Uma escolha conveniente, que requer só $O(n)$ operações para avaliar L e L^{-1} , é dada por $L(y) = Ny$, $L^{-1}(x) = N^T x$, onde N é a matriz de ordem $(n+1) \times n$ dada por as últimas n colunas da matriz ortogonal $I - a^T a / \alpha$, onde

$$\alpha = (n+1) + (n+1)^{1/2} \quad \text{e} \quad a = (1 + (n+1)^{1/2}, 1, \dots, 1) \in \mathbb{R}^{n+1}.$$

Para o caso do modelo walrasiano geral de equilíbrio com produção, temos uma matriz $A = [-I, C]$ de ordem $(n+1) \times m$ e procuramos um vetor $x^* \in S^n$, tal que o excesso de demanda nesse ponto, $e(x^*)$, satisfaça $e(x^*) = Ay$, $y \geq 0$ e $x^{*T} A \leq 0$. Sejam a^1, \dots, a^m as colunas de A e supomos que $\hat{x} > 0 / \hat{x}^T A < 0$.

Convertiremos este problema em um problema de achar um zero de uma correspondência $G: T^n \rightarrow U^{n^*}$; e equivalentemente, como antes, podemos achar o zero de uma outra correspondência $\tilde{G}: \mathbb{R}^n \rightarrow \mathbb{R}^{n^*}$.

A idéia para construir G é definir ajustes de preços adequados, quando estes não representam equilíbrio. Definimos uma subdivisão poliedral X_0, X_1, \dots, X_m de T^n da seguinte forma: $X_0 = \{x \in T^n / x^T A \leq 0\} \subset S^n$ e $X_j = \{x \in T^n / x^T a^j = \max\{x^T a^k / 1 \leq k \leq m\} \geq 0\}$, para $j = 1, 2, \dots, m$. Um preço de equilíbrio deve cair em X_0 ; X_j é o conjunto de "preços" (possivelmente sem respeitar a não-negatividade)

no que a atividade j faz o lucro máximo que é não negativo. Em X_j modificamos x para diminuir o lucro da atividade a^j ; em X_0 modificamos x para aumentar (diminuir) os preços dos bens com demanda (oferta) em excesso. Então, gostaríamos de por $-a^j$ em $G(x)$, para x em X_j , e $e(x)$ em $G(x)$, para x em X_0 . Para ficar em U^n devemos fazer algumas modificações. Definimos:

$$\hat{e}(x) = (I - \frac{e^T e}{n+1}) e(x) , \quad \hat{A} = (I - \frac{e^T e}{n+1}) A .$$

Então $\hat{e}(x)$ e \hat{a}^j são as projeções ortogonais de $e(x)$ e a^j sobre U^n .

Se definimos $g^0(x) = \hat{e}(x)$, $g^j(x) = -\hat{a}^j$ e

$$G(x) = \text{conv}\{g^j(x) / 0 \leq j \leq m \text{ e } x \in X_j\} ,$$

pode-se ver que se $o \in G(x^*)$, então $x^* \in X_0 \subset S^n$ e o peso de $g^0(x^*)$ é positivo. Assim, $\hat{e}(x^*) - \hat{A}y = 0$, para algum $y \geq 0$. E da qui, temos que:

$$e(x^*) = Ay \quad \text{com} \quad x^{*T}A \leq 0 .$$

CAPÍTULO IV

O ALGORÍTMO DE MERRILL

Os algoritmos desenvolvidos por Scarf para aproximação de pontos fixos tem a desvantagem computacional de que a triangulação usada é fixa. Portanto, para melhorar a aproximação obtida, tem que se mudar de triangulação (para uma mais fina) sem possibilidade de reaproveitar os cálculos anteriores. Merrill (1972) propõe um algoritmo que permite reaproveitar a aproximação feita com uma determinada triangulação, fazendo com que o ponto inicial para iniciar a busca do ponto fixo possa ser qualquer um (o algoritmo de Scarf precisa iniciar na fronteira do simplexo). Portanto, com triangulações T_k , com $\text{tam } T_k \rightarrow 0$, consegue-se boas aproximações dos pontos fixos.

Iniciamos a nossa discussão do algoritmo de Merrill apresentando a triangulação de Kuhn usada no algoritmo.

4.1. TRIANGULAÇÃO DE KUHN

Como já dissemos antes, a diferença entre os algoritmos simpliciais está na triangulação T usada no domínio da função. Aqui apresentamos a triangulação K , usada por Merrill. Esta triangulação é chamada com frequência "triangulação de Kuhn" (ver Todd (1976)).

Seja $K_0 = \mathbb{Z}^n = \{v \in \mathbb{R}^n / v_i \in \mathbb{Z}, i = 1, \dots, n\}$. Se $v^0 \in K_0$ e p é uma permutação de $\{1, \dots, n\}$, então denotamos por $\sigma(v^0, p)$ o n -simplexo $\langle v^0, \dots, v^n \rangle$, onde

$$v^i = v^{i+1} + e^{p_i}, \quad i=1, \dots, n.$$

LEMA 4.1. Se $K = \{\sigma(v^0, p) / v^0 \in K^0\}$ então K é uma triangulação de \mathbb{R}^n .

PROVA. Ver Todd (1976).

Já que temos triangulizado \mathbb{R}^n , vejamos como pode-se triangulizar S^n . Para qualquer um $C \subset \mathbb{R}^n$ e $\delta \in \mathbb{R}$, δC denotará $\{\delta c / c \in C\}$. Para qualquer família T de subconjuntos de \mathbb{R}^n e qualquer $\delta \in \mathbb{R}$, não nulo, δT denotará $\{\delta C / C \in T\}$. Se T é uma triangulação de D , δT será uma triangulação de δD . Em particular δK é uma triangulação de \mathbb{R}^n . Então, temos triangulações de \mathbb{R}^n de diâmetro arbitrário; estas geram triangulações de S^n como segue. Seja $C^n = \{x \in \mathbb{R}^n / 1 \geq x_1 \geq \dots \geq x_n \geq 0\}$, primeiro triangularemos C^n e depois obteremos triangulações de S^n .

Escolher $0 < m \in \mathbb{Z}$ e seja \tilde{K} a coleção de todos os n -simplexos de K que encontram $mC^n = \{x \in \mathbb{R}^n / m \geq x_1 \geq \dots \geq x_n \geq 0\}$. Seja $x \in mC^n / x \in \sigma \in K$. Como cada vértice de σ pode ser obtido de x , arredondando cada componente de x ao inteiro imediatamente superior ou inferior, então cada vértice de σ também está em mC^n . É simples de ser visto que \tilde{K}

trianguliza mC^n e $m^{-1}\tilde{K}$ trianguliza C^n . Agora usando uma transformação adequada obtemos uma triangulação de S^n . Vejamos, seja $h : C^n \rightarrow S^n$, onde $S^n = [e^0, e^1, \dots, e^n]$, tal que $h(x) = e^0 + Qc$, com Q como a matriz

$$\begin{bmatrix} -1 & & & & \\ & \cdot & & & 0 \\ -1 & & \cdot & & \\ & \cdot & & \cdot & \\ 0 & & \cdot & & -1 \\ & & & \cdot & \\ & & & & 1 \end{bmatrix}_{(n+1) \times n}$$

Então h é um homeomorfismo entre C^n e S^n .

Donde, se C^n está triangulizado por $m^{-1}\tilde{K}$, transformando C^n , por h , em S^n , obtemos uma triangulação de S^n que denotaremos $K_1(m)$.

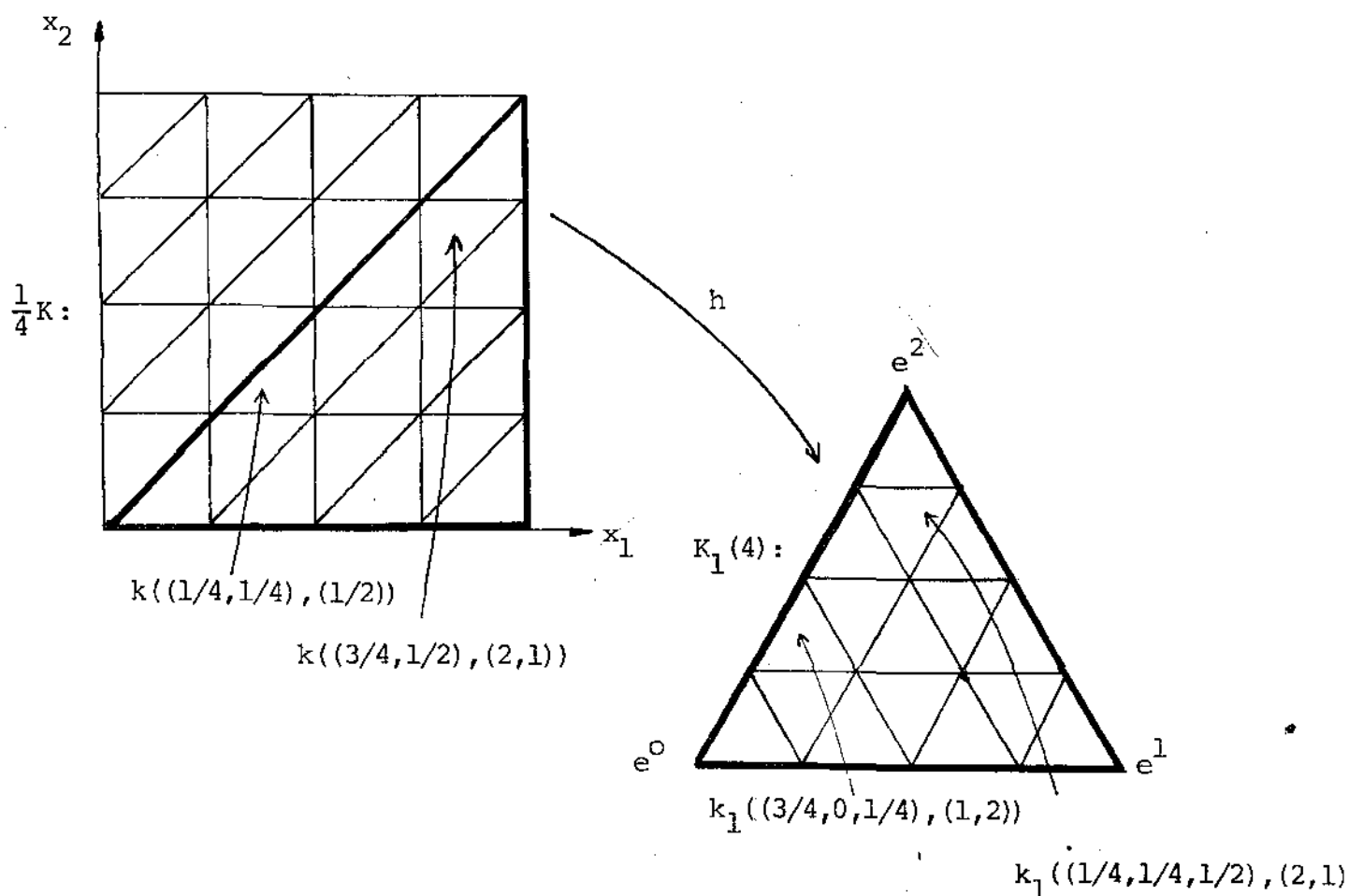
DEFINIÇÃO 4.2. Seja a matriz Q dada acima, q^j a sua j -ésima coluna, $j = 1, \dots, n$, e $K_1^0(m) = \{y \in S^n / my_i \in \mathbb{Z}, i = 0, \dots, n\}$. Se $y^0 \in K_1^0(m)$ e p é uma permutação de $\{1, \dots, n\}$, seja $\sigma = \langle y^0, \dots, y^n \rangle$ onde

$$y^i = y^{i-1} + m^{-1}q^{p_i}, \quad i = 1, \dots, n.$$

Se $\sigma \in S^n$, denotaremos $\sigma = k_1(y^0, p)$ (m é implícito). $K_1(m)$

é a coleção de todos esses $k_1(y^0, p)$.

EXEMPLO 1.3. ($n = 2$).



Os algoritmos homotópicos geram uma sequência de simplexes $\sigma_1, \sigma_2, \dots$, da triangulação T com σ_i e σ_{i+1} adjacentes. Obtemos σ_{i+1} de σ_i eliminando um vértice y^- de σ_i e aumentando um novo vértice y^+ . As regras para obter σ_{i+1} de σ_i e y^- são chamadas de regras de pivoteamento de T .

Dado que geralmente numerosos destes simplexos são gerados para obter uma aproximação suficientemente boa dum ponto fixo, é necessário uma regra de pivoteamento simples. Seja $\sigma = (y^0, \dots, y^n) = k(y^0, p^1)$ dado. Desejamos obter $\tau = (z^0, \dots, z^n) = k(z^0, p^2)$, com todos os vértices de σ exceto y^i . A tabela abaixo mostra como z^0 e p^2 dependem de y^0 , p^1 e i (desta tabela é simples obter os vértices de τ):

$T = K$	z^0	p^2
$i = 0$	$y^0 + e^{p_1^1}$	$(p_2^1, \dots, p_n^1, p_1^1)$
$0 < i < n$	y^0	$(p_1^1, \dots, p_{i+1}^1, p_i^1, \dots, p_n^1)$
$i = n$	$y^0 - e^{p_n^1}$	$(p_n^1, p_1^1, \dots, p_{n-1}^1)$

Para δK , substituir $e^{p_1^1}$ e $e^{p_n^1}$ por $\delta e^{p_1^1}$ e $\delta e^{p_n^1}$, respectivamente. Para $T = K_1(m)$, substituir $e^{p_1^1}$ e $e^{p_n^1}$ por $m^{-1} q^{p_1^1}$ e $m^{-1} q^{p_n^1}$.

4.2. HIPÓTESES PARA A FORMULAÇÃO DO ALGORÍTMO

Começaremos estabelecendo as hipóteses que faremos no

desenvolvimento do algoritmo.

Desejamos achar um zero aproximado de uma correspondência s. c.s. $\tilde{G} : \mathbb{R}^n \rightarrow \mathbb{R}^{n*}$. Supomos que \tilde{G} satisfaz as hipóteses do Corolário 2.9.

HIPÓTESE 4.3. Existe $x^0 \in \mathbb{R}^n$, $\delta > 0$ e $\mu > 0$ tal que se $x \notin B(x^0, \mu)$, $f \in \tilde{G}(x) + \{x\}$ e $z \in B(x, \delta)$, então $(f - x)^T(x^0 - z) > 0$. (com a hipótese feita em 3.4.1, $\{x > 0 / A^T x < 0\} \neq \emptyset$, é fácil ver que nosso problema de achar equilíbrio para uma economia com produção satisfaz a hipótese 4.3). Escolhamos $c \in \mathbb{R}^n$ arbitrariamente e seja M uma matriz real não singular ($n \times n$). Definimos a homotopia

$$H(x, t) = \{t\tilde{g} + (1-t)M(c-x)/\tilde{g} \in \tilde{G}(x)\}, \quad (x, t) \in \mathbb{R}^n \times [0, 1] \quad (4.1)$$

Observar que $(c^T, 0)^T$ é o único zero de H em $\mathbb{R}^n \times \{0\}$ e que qualquer zero de H em $\mathbb{R}^n \times \{1\}$ é zero em \tilde{G} .

Uma triangulação T de $\mathbb{R}^n \times [0, 1]$ com $T^0 \subset \mathbb{R}^n \times \{0, 1\}$ é chamada especial (a triangulação de Kuhn é especial). Se $\sigma \in T$, $\text{diam}_q^1 \sigma = \sup \{\|x - z\|_q / (x, t_1), (z, t_2) \in \sigma\}$ é o diâmetro da projeção de σ , e o $\text{tam}_q^1 T = \sup_{\sigma \in T} \text{diam}_q^1 \sigma$, para $q = 2, \infty$.

A triangulação de \mathbb{R}^n cujos simplexos são $\langle x^0, \dots, x^n \rangle$ para $\sigma = \langle (x^0, t_0), \dots, (x^n, t_n) \rangle \in T^n$, com $o \in \mathbb{R}^n \times \{1\}$, é denotada por T_i , $i = 0, 1$.

Observar que a restrição \tilde{K} de K (como triangulação de \mathbb{R}^{n+1}) a $\mathbb{R}^n \times [0,1]$ é especial. Seja E a matriz $(n+1) \times (n+1)$, $[e^0, \epsilon e^1, \dots, \epsilon e^n]$, então

$$\tilde{K}(\epsilon) = \{ \langle Ev^{-1}, \dots, Ev^n \rangle / \langle v^{-1}, \dots, v^n \rangle \in \tilde{K} \}$$

é especial. Temos que $\text{tam}_\infty^1(\tilde{K}(\epsilon)) = \epsilon$ e $\text{tam}_2^1(\tilde{K}(\epsilon)) = \epsilon\sqrt{n}$.

Vamos supor que c cai em um n -simplexo de T_0 ou, equivalentemente, que $(c^T, 0)^T$ cai em um n -simplexo τ_0 de T .

DEFINIÇÃO 4.4. Seja $H_T(v)$ uma aproximação LP de H com respeito a T . Se $\sigma = \langle v^0, \dots, v^{k+1} \rangle \in T^{k+1}$, definimos a matriz de rótulos de σ como

$$L_\sigma = \begin{bmatrix} 1 & \dots & 1 \\ H_T(v^0) & \dots & H_T(v^{k+1}) \end{bmatrix}$$

Dizemos que σ é completo se existe uma solução a

$$L_\sigma y = e^0, \quad y \geq 0.$$

LEMA 4.5. Sejam H, T e H_T como acima e $\sigma \in T^n \cup T^{n+1}$. Então existe um zero de H_T em $\bar{\sigma}$ se e somente se σ é completo. O único

zero de H_T em $\mathbb{R}^n \times \{0\}$ é $(c^T, 0)^T$, e se $H_T(x^*, 1) = 0$, então x^* é um zero aproximado de \tilde{G} respeito de T_1 .

PROVA. Ver Todd (1976).

4.3. CONVERGÊNCIA DO ALGORÍTMO DE MERRILL

Para estudar a convergência do algoritmo, vamos interpretar a trajetória de zeros determinada por H_T como um grafo Γ .

DEFINIÇÃO 4.6. Os nós de Γ são $(n+1)$ -simplexos completos de T e n -simplexos completos de T^n que caem em $\mathbb{R}^n \times \{0\}$ ou $\mathbb{R}^n \times \{1\}$. Dois nós são adjacentes se um deles é uma face do outro ou se eles compartilham uma face completa.

LEMA 4.7.

- (a) Γ tem só um nó em $\mathbb{R}^n \times \{0\}$, digamos, o n -simplexo τ_0 de T^n que contém $(c^T, 0)^T$.
- (b) Qualquer nó de Γ que cai em $\mathbb{R}^n \times \{1\}$ fornece um zero aproximado de \tilde{G} .
- (c) Se $\tan_2 T \leq \delta$ e M é definida positiva, e a hipótese 4.3 é satisfeita, Γ tem um número finito de nós.

PROVA. Os itens (a) e (b) são consequência imediata do lema 4.5.

(c) Provaremos que fora de uma região compacta não existem simplexos completos. Temos x^0 , μ e δ tal que se $x \notin B(x^0, \mu)$, $f \in \tilde{G}(x) + \{x\}$ e $z \in B(x, \delta)$, então $(f - x)^T(x^0 - z) > 0$. Seja $\|M\|_2 = \max\{\|Mx\|_2 / \|x\|_2 = 1\}$ e $\eta = \min\{x^T Mx / \|x\|_2 = 1\}$. Como $\{x \in \mathbb{R}^n / \|x\|_2 = 1\}$ é compacto, então $\|M\|_2 < \infty$, $\eta > 0$.

Seja $\mu' = \max\{\mu + \delta, \|M\|_2(\delta + \|c - x^0\|_2)/\eta\}$ e $C = B(x^0, \mu') \times [0, 1] \subset \mathbb{R}^{n+1}$. Então C é compacto. Provaremos que todos os simplexos completos estão em C .

Seja $\sigma \in T$ com um vértice $v^* = (x^*, t^*) \notin C$ e $s = x^0 - x^*$. Provaremos que $s^T H_T(v) > 0$, para todo vértice v de σ ; portanto, σ não pode ser completo (pois, σ completo implica $\exists \omega \in S^n / \sum \omega_i H_T(v^i) = 0 \Rightarrow 0 = \sum \omega_i s^T H_T(v^i) > 0$, uma contradição!).

Seja $v = (x, 1)$ vértice de σ , então $H_T(v) = \tilde{g} \in \tilde{G}(x)$, para algum \tilde{g} , então $f = \tilde{g} + x \in \tilde{G}(x) + \{x\}$. Como $x^* \notin B(x^0, \mu + \delta)$ e $x \in B(x^*, \delta)$, então $x \notin B(x^0, \mu)$. De onde pela hipótese 4.3, $s^T H_T(v) = (f - x)(x^0 - x^*) > 0$.

Seja $v = (x, 0)$ vértice de σ , então $H_T(v) = M(c - x)$ e

$$s^T H_T(v) = (x^0 - x^*)^T M(c - x) = (x^0 - x^*)^T M(x^0 - x^*) + \\ + (x^0 - x^*)^T M(c - x^0) + (x^0 - x^*)^T M(x^* - x).$$

O primeiro termo do segundo membro é positivo; vejamos que domina aos outros dois:

Se $\|x^0 - x^*\|_2 = v > \mu^*$,

$$(x^0 - x^*)^T M (x^0 - x^*) \geq v^2 \eta > \eta v \mu^* \geq v \|M\|_2 \delta + v \|M\|_2 \|c - x^0\|_2 \quad (1)$$

Como,

$$v \|M\|_2 \delta \geq \|x^0 - x^*\|_2 \|M(x^* - x)\|_2 \geq |(x^0 - x^*)^T M(x^* - x)|$$

$$v \|M\|_2 \|c - x^0\|_2 \geq \|x^0 - x^*\|_2 \|M(c - x^0)\|_2 \geq |(x^0 - x^*)^T M(c - x^0)|$$

Então, (1) implica que

$$(x^0 - x^*)^T M (x^0 - x^*) > |(x^0 - x^*)^T M(x^* - x) + (x^0 - x^*)^T M(c - x^0)|,$$

de onde $S_{H_T}^T(v) > 0$.

Portanto, todos os simplexes completos estão em C e o resultado está provado. ■

É simples ver que os nós de Γ tem grau 1 ou 2, segundo sejam n -simplexos ou $(n+1)$ -simplexos. Então, combinando os resultados anteriores, temos que:

TEOREMA 4.8. Cada componente conexa do grafo finito Γ é um circuito simples ou uma trajetória simples, onde cada um dos seus pontos terminais é τ_0 ou um n -simplexo em $\mathbb{R}^n \times \{1\}$.

4.4. O ALGORÍTIMO

Supor que temos \tilde{G}, δ, T, H e H_T como na Seção 2, com $\text{tam}'_2 \leq \delta$. Seja c uma aproximação inicial a um zero de \tilde{G} , τ_0 o simplexo em T^n que contém $(c^T, 0)^T$ e σ_1 o simplexo em T^{n+1} com τ_0 como uma face. O algoritmo constrói a trajetória simples em Γ com τ_0 como um ponto inicial.

4.4.1. ALGORÍTIMO BÁSICO

PASSO 0. Formar o sistema correspondente provando que τ_0 é completo. Seja v^+ o vértice de σ_1 que não está em τ_0 . Fazer $k \leftarrow 1$.

PASSO 1. Calcular $H_T(v^+)$ e introduzir a coluna $(H_T^1(v^+))$ na base atual, substituindo a coluna $(H_T^1(v^-))$, com v^- um vértice de σ_k (determina-se v^- com um pivotamento tipo método Simplex da programação linear).

PASSO 2. Se a face τ de σ_k , oposta a v^- , cai em $\mathbb{R}^n \times \{1\}$, PARAR (τ contém um zero aproximado de \tilde{G}). Caso contrário, seja σ_{k+1} o único simplexo de T que compartilha a face τ . Seja v^+ o novo vértice de σ_{k+1} , fazer $k \leftarrow k+1$ e ir para o PASSO 1.

4.4.2. ALGORÍTMO COM REFINAMENTO DE MALHA

PASSO 0. Escolher uma seqüência $\{\delta_i\}$ decrescente com $\delta_i \rightarrow 0$ e $\delta_0 \leq \delta$. Para cada i escolher uma triangulação $T_{(i)}$ com $\text{tam}_2 T_{(i)} = \delta_i$. Pegar $x^0 \in \mathbb{R}^n$ e fazer $p \leftarrow 0$.

PASSO 1. Aplicar o algoritmo 4.4.1 com $T_{(p)}$ e x^p substituindo T e c , respectivamente, e alguma matriz M_p , positiva definida, para obter um zero aproximado x^{p+1} . Fazer $p \leftarrow p+1$ e voltar ao PASSO 1.

Eventualmente, x^p deve-se perturbar senão cai em um n -simplexo de $(T_{(p)})_0$.

CAPÍTULO V

EXPERIÊNCIA COMPUTACIONAL

O algoritmo de Merrill foi programado em FORTRAN77, precisão dupla, e rodado em um computador DEC-VAX/VMS Version V4.3. Como a matriz M, em (4.1), foi usada a matriz identidade e para efeitos de gerar uma sequência de malhas de tamanho decrescente (4.4.2 - PASSO 0), usou-se o fator constante 1/3. Como critério de convergência, usou-se um tamanho de malha menor ou igual a 10^{-11} ou $\text{RESÍDUO} = \|\text{OFERTA-DEMANDA}\|_2 \leq 10^{-10}$. A cada ciclo maior permitimos um número máximo de 300 iterações (pivoteamentos tipo método Simplex da programação linear). De maneira que, se em um máximo de 300 iterações não é achada uma nova aproximação, é gerado um novo simplexo adjacente inicial para reiniciar o ciclo maior ou solicitado para fornecer uma nova tentativa inicial (se for o primeiro ciclo maior).

5.1. DESCRIÇÃO DO PROGRAMA

Aqui apresentamos as variáveis mais relevantes usadas no programa, o fornecimento dos dados e um diagrama de blocos básico :

N: número de bens (é solicitado na execução do programa).

M: número de atividades lineares, sem incluir as atividades de desperdício (é solicitado na execução do programa).

K1: número de consumidores (é solicitado na execução do programa).

O programa tem como alternativa apresentar os resultados a cada ciclo maior ou apresentar, apenas, os resultados do último ciclo maior (o de melhor aproximação conseguida). Isto é solicitado na execução do programa com a pergunta:

"QUER SAÍDA DOS RESULTADOS DE CADA CICLO MAIOR?(S/N)" .

ID: tamanho da malha em que é subdividido o simplexo unitário a cada iteração (o tamanho inicial é solicitado na execução do programa).

CHUTE: vetor de dimensão N que contém a tentativa inicial (a primeira tentativa inicial é solicitada na execução do programa).

SS: matriz de dimensão $(N+1) \times (N+1)$ cujas colunas contém os vêrtices do n-simplexo completo considerado em cada iteração.

O programa requer um arquivo de dados, FOR020.DAT, com cinco registros para fornecer o seguinte:

UTIL: tipo de função de utilidade, que pode ser COBB-DOUGLAS, LEONTIEFF ou ESC. É o primeiro registro do arquivo. UTIL está definida como CHARACTER UTIL * 15.

A: matriz de dimensão $N \times M$ cujas colunas representam as atividades lineares (não são consideradas as atividades de desperdício). É o segundo registro do arquivo e os dados são fornecidos por linhas da matriz.

WØ: matriz de dimensão $K1 \times N$, cujas linhas representam as cestas iniciais de cada consumidor. É o terceiro registro do arquivo e os dados são fornecidos por linhas da matriz.

PARUTI: matriz de dimensão $K1 \times N$, cujas linhas representam os parâmetros de utilidade de cada consumidor. É o quarto registro do arquivo e os dados são fornecidos por linhas da matriz.

B: vetor de dimensão $K1$ associado a função de utilidade. É o quinto registro do arquivo. Se a função de utilidade for Cobb-Douglas, B deve ser composto de uns, i.e., $B = (1, \dots, 1)$; e se for de Leontieff, $B = (0, \dots, 0)$.

Outras VARIÁVEIS usadas são:

ITER: número de iterações acumuladas (pivoteamento tipo método Simplex da programação linear).

ICICLO: número de ciclos maiores realizados.

NÍVEL: Níveis de operação das atividades lineares depois de cada ciclo maior.

PREÇO: Preços dos bens.

LUCRO: Lucros de cada atividade depois de cada ciclo maior.

DEMAN: Demandas dos bens.

OFERTA: Ofertas dos bens depois de cada ciclo maior.

RESTO(I): $|OFERTA(I) - DEMAN(I)|$, $I = 1, \dots, N$, depois de cada ciclo maior.

RES: Resíduo total, i.e., $\|RESTO\|_2$.

J1: Índice da coluna de SS que entra na base a cada iteração.

J2: Índice da coluna de SS a ser substituída para gerar o simple xo adjacente a cada iteração.

J3: Índice da coluna que sai da base a cada iteração.

BINV: Matriz de dimensão $N \times N$ que representa a inversa da base a cada iteração.

JCOLSS: Vetor de dimensão N que indica as colunas de SS na base a cada iteração.

ROT: Vetor de dimensão N que representa o rótulo do vetor que entra na base.

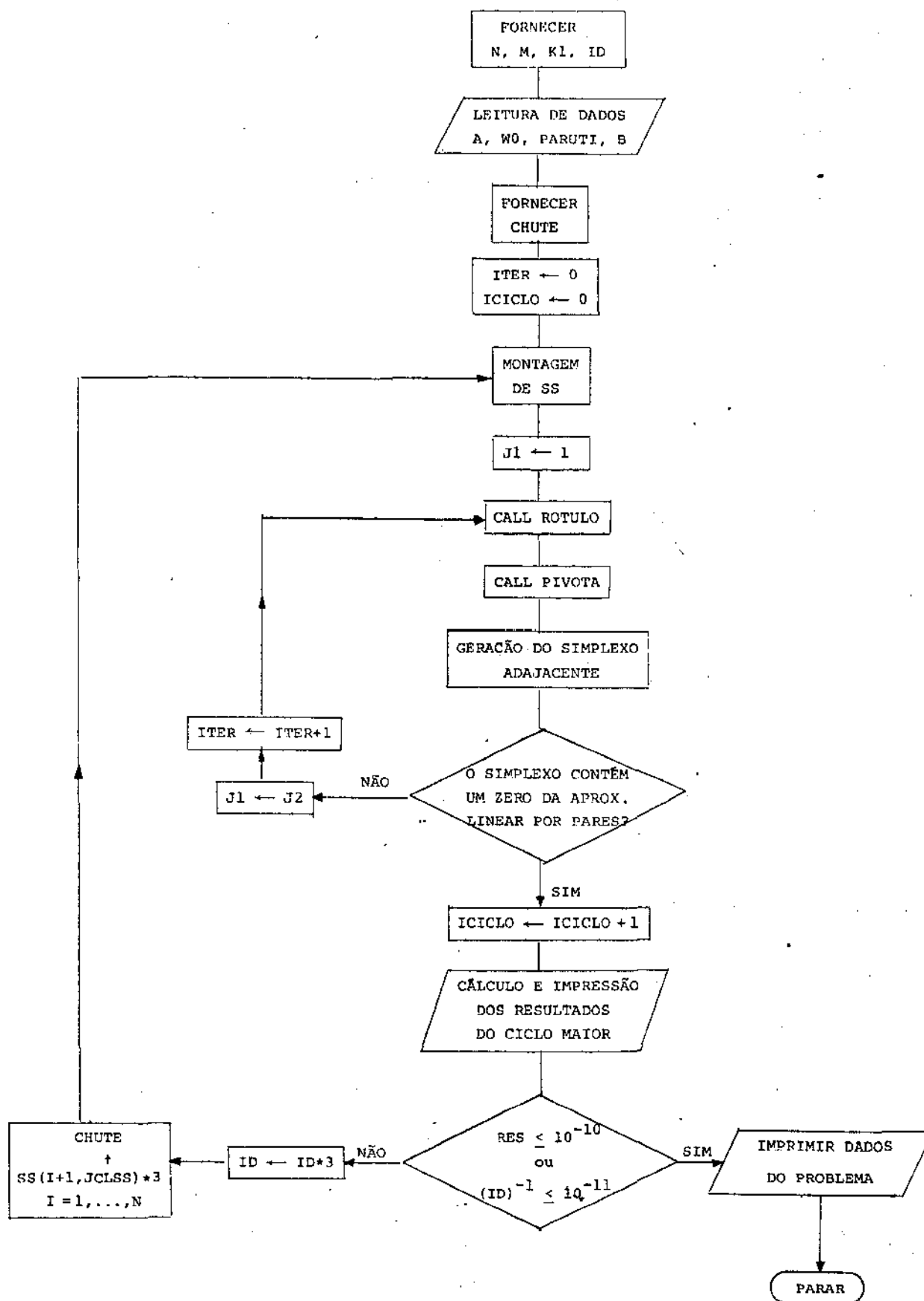
O programa consta do programa principal chamado de PROGRAMA MERRILL e as subrotinas ROTULO, PIVOTA e DEMAND.

ROTULO: Calcula o vetor ROT.

PIVOTA: Faz o pivoteamento tipo método Simplex para calcular J2 e J3; atualiza também os valores de NIVEL e BINV.

DEMAND: Calcula a demanda de cada bem.

Apresentamos a seguir o diagrama de blocos básico do PROGRAMA MERRILL (o programa está no apêndice deste capítulo).



5.2. RESULTADOS NUMÉRICOS

Com o PROGRAMA MERRILL foram rodados três exemplos que chamaremos de Problemas 1, 2 e 3, respectivamente. O Problema 1 é devido a Scarf (1967). O problema 3 é uma variação do Problema 2.

PROBLEMA 1.

O tempo de processamento foi de 10 segundos, com um total de 4047 iterações e 456 avaliações da função demanda. O tamanho inicial da malha foi 50 e o CHUTE = (9,9,8,8,8,8). A convergência foi atingida no ciclo maior 21. Os resultados são mostrados a continuação.

PROBLEMA 2.

O tempo de processamento foi de 5 segundos, com um total de 3853 iterações e 39 avaliações da função demanda. O tamanho inicial da malha foi 50 e o CHUTE = (15,5,12,8,10). A convergência foi atingida no ciclo maior 21. Os resultados são mostrados a continuação.



Income Statement for the Year 2019 (in million VND)

Income Statement for the Year 2019

Income Statement for the Year 2019

Revenue	200.00	100.00	50.00	25.00	12.50	6.25
Cost of Sales	- 50.00	- 25.00	- 12.50	- 6.25	- 3.125	- 1.5625
Gross Profit	150.00	75.00	37.50	18.75	9.375	4.6875
Operating Expenses	- 100.00	- 50.00	- 25.00	- 12.50	- 6.25	- 3.125
Operating Income	50.00	25.00	12.50	6.25	3.125	1.5625
Non-Operating Income	10.00	5.00	2.50	1.25	0.625	0.3125

Income Statement for the Year 2019

Revenue	100.00	50.00	25.00	12.50	6.25	3.125
Cost of Sales	- 25.00	- 12.50	- 6.25	- 3.125	- 1.5625	- 0.78125

Income Statement for the Year 2019

Revenue	50.00	25.00	12.50	6.25	3.125	1.5625
Cost of Sales	- 12.50	- 6.25	- 3.125	- 1.5625	- 0.78125	- 0.390625

Income Statement for the Year 2019

Income Statement for the Year 2019



გამოცემისათვის: მთავარი, ვიცე, დირექტორი, უფროსი
პო. 111 77 41 11, 111 77 41 11
სტ. 111 77 41 11, 111 77 41 11
სტ. 111 77 41 11, 111 77 41 11

გამოცემისათვის: მთავარი, ვიცე, დირექტორი, უფროსი

გამოცემისათვის:	პო. 111 77 41 11	სტ. 111 77 41 11
1	111 77 41 11	111 77 41 11
2	111 77 41 11	111 77 41 11
3	111 77 41 11	111 77 41 11
4	111 77 41 11	111 77 41 11
5	111 77 41 11	111 77 41 11
6	111 77 41 11	111 77 41 11

გამოცემისათვის:	პო. 111 77 41 11	სტ. 111 77 41 11	სტ. 111 77 41 11
1	111 77 41 11	111 77 41 11	111 77 41 11
2	111 77 41 11	111 77 41 11	111 77 41 11
3	111 77 41 11	111 77 41 11	111 77 41 11
4	111 77 41 11	111 77 41 11	111 77 41 11
5	111 77 41 11	111 77 41 11	111 77 41 11

გამოცემისათვის: მთავარი, ვიცე, დირექტორი, უფროსი

- 111 77 41 11
- 111 77 41 11
- 111 77 41 11
- 111 77 41 11
- 111 77 41 11

PROBLEMA 3.

O tempo de processamento foi de 4 segundos, com um total de 1884 iterações e 115 avaliações da função demanda. O tamanho inicial da malha foi 50 e o CHUTE = (12,8,15,5,10) A convergência foi atingida no ciclo maior 21. Os resultados são mostrados a continuação.





Journal of Management Inquiry, Vol. 19 No. 1, March 2010
DOI: 10.1177/1056492609358010
© The Author(s) 2010
Reprints and permissions: <http://www.sagepub.com/journalsPermissions.nav>

Downloaded from <http://ajphaphysocpharm.sagepub.com>

Variable	Value	Loglik
1	0.00000000	-0.00000000
2	0.00000000	-0.00000000
3	0.00000000	-0.00000000
4	0.00000000	-0.00000000
5	0.00000000	-0.00000000
6	0.00000000	-0.00000000
7	0.00000000	-0.00000000
8	0.00000000	-0.00000000
9	0.00000000	-0.00000000
10	0.00000000	-0.00000000
11	0.00000000	-0.00000000
12	0.00000000	-0.00000000
13	0.00000000	-0.00000000
14	0.00000000	-0.00000000
15	0.00000000	-0.00000000
16	0.00000000	-0.00000000
17	0.00000000	-0.00000000
18	0.00000000	-0.00000000
19	0.00000000	-0.00000000
20	0.00000000	-0.00000000
21	0.00000000	-0.00000000
22	0.00000000	-0.00000000
23	0.00000000	-0.00000000
24	0.00000000	-0.00000000
25	0.00000000	-0.00000000
26	0.00000000	-0.00000000
27	0.00000000	-0.00000000
28	0.00000000	-0.00000000
29	0.00000000	-0.00000000
30	0.00000000	-0.00000000
31	0.00000000	-0.00000000
32	0.00000000	-0.00000000
33	0.00000000	-0.00000000
34	0.00000000	-0.00000000
35	0.00000000	-0.00000000
36	0.00000000	-0.00000000
37	0.00000000	-0.00000000
38	0.00000000	-0.00000000
39	0.00000000	-0.00000000
40	0.00000000	-0.00000000
41	0.00000000	-0.00000000
42	0.00000000	-0.00000000
43	0.00000000	-0.00000000
44	0.00000000	-0.00000000
45	0.00000000	-0.00000000
46	0.00000000	-0.00000000
47	0.00000000	-0.00000000
48	0.00000000	-0.00000000
49	0.00000000	-0.00000000
50	0.00000000	-0.00000000
51	0.00000000	-0.00000000
52	0.00000000	-0.00000000
53	0.00000000	-0.00000000
54	0.00000000	-0.00000000
55	0.00000000	-0.00000000
56	0.00000000	-0.00000000
57	0.00000000	-0.00000000
58	0.00000000	-0.00000000
59	0.00000000	-0.00000000
60	0.00000000	-0.00000000
61	0.00000000	-0.00000000
62	0.00000000	-0.00000000
63	0.00000000	-0.00000000
64	0.00000000	-0.00000000
65	0.00000000	-0.00000000
66	0.00000000	-0.00000000
67	0.00000000	-0.00000000
68	0.00000000	-0.00000000
69	0.00000000	-0.00000000
70	0.00000000	-0.00000000
71	0.00000000	-0.00000000
72	0.00000000	-0.00000000
73	0.00000000	-0.00000000
74	0.00000000	-0.00000000
75	0.00000000	-0.00000000
76	0.00000000	-0.00000000
77	0.00000000	-0.00000000
78	0.00000000	-0.00000000
79	0.00000000	-0.00000000
80	0.00000000	-0.00000000
81	0.00000000	-0.00000000
82	0.00000000	-0.00000000
83	0.00000000	-0.00000000
84	0.00000000	-0.00000000
85	0.00000000	-0.00000000
86	0.00000000	-0.00000000
87	0.000	

	1974-75	1975-76	1976-77
1	11751.000000	11841.000000	1.1077127611-04
2	12151.000000	12241.000000	0.2010742-0-01
3	12551.000000	12641.000000	1.80717000-00
4	12951.000000	13041.000000	1.604707331-12
5	13351.000000	13441.000000	0.001051517-12

... ..

2. The \mathcal{H}_2 norm of the system is $\sqrt{2}$.
3. The \mathcal{H}_2 norm of the system is $\sqrt{2}$.
4. The \mathcal{H}_2 norm of the system is $\sqrt{2}$.
5. The \mathcal{H}_2 norm of the system is $\sqrt{2}$.



PROGRAM MERRILL

ALGORITMO DE MERRILL PARA O CALCULO DE PREÇOS DE EQUILIBRIO
NUMA ECONOMIA WALRASIANA COM PRODUÇÃO DETERMINADA POR ATIVI-
DADES LINEARES

33 É A MATRIZ DO A-SIMPLEX COMPLETO CONSIDERADO EM CADA ITE-
RAÇÃO. AS COLUNAS SÃO OS NUMERADORES DOS VERTICES. O NA PRI-
MEIRA LINHA INDICA QUE O RESTO DA COLUNA É UM VERTICE DO SIM-
PLEX ORIGINAL; 1 INDICA QUE O VERTICE É ARTIFICIAL.

AO CORRESPONDE A CESTA INICIAL, CHUTE AO CHUTE INICIAL, PARU-
TI A MATRIZ DE PARÂMETROS DE UTILIDADES, R AO VETOR DE PARÂME-
TROS E A MATRIZ A REPRESENTA AS ATIVIDADES LINEARES.

```
REAL*8 RESTO(20),RST(200),Z(20),EPS(20),LUCRO(20),CBERTA(20),  
1 NIVEL(20),PARUTI(20,20),DEMAN(20),A(20,20),PREC(20),BINV(20,  
2 20),PIV0(20,20),PIV1(20),X0(20,20),RET1(20),W1(20),W2(20),  
3 X0,X1,Q,S9,W(20)  
INTEGER JCOLSS(20),JCOLA(20),BASAC(20)  
REAL*8 SSAC(20,20),SS(20,20),CHUTE(20),ID,IS,RES,CIC,CNIVEL(20)  
CHARACTER YX,YX1,YX2,UTIL*16
```

```
PRINT 12  
FORMAT (////////)  
PRINT *,'  
PRINT *,'  
PRINT *,'  
PRINT *,'  
PRINT *,'  
PRINT 13  
FORMAT (//)
```

```
WRITE(21,*) '  
WRITE(21,*) '  
WRITE(21,*) '  
WRITE(21,*) '  
WRITE(21,*) '  
WRITE(21,13)
```

```
PRINT *,'OBSERVAÇÃO:'  
PRINT *,'  
PRINT *,' ESTE PROGRAMA PRECISA UM ARQUIVO DE DADOS,'  
PRINT *,'FDR020.DAT, COM CINCO REGISTROS. O PRIMEIRO CONTEM O'  
PRINT *,'TIPO DE FUNÇÃO DE UTILIDADE (COBB-DOUGLASS,LEONTIEFF,'  
PRINT *,'ESQ), O SEGUNDO A MATRIZ DE ATIVIDADES, O TERCEIRO A'  
PRINT *,'MATRIZ DE DOTAS (OU RECURSOS INICIAIS), O QUARTO A MA-'  
PRINT *,'TRIZ DE PARÂMETROS DE UTILIDADES E O QUINTO O VETOR B'  
PRINT *,'DE PARÂMETROS (ESTE VETOR DEVE SER B=(1,...,1), SE A U-'  
PRINT *,'TILIDADE FOR COBB-DOUGLASS, OU B=(0,...,0), SE FOR LE-'  
PRINT *,'ONTIEFF).'  
PRINT *,' AS MATRIZES ESTÃO DADAS POR FILAS, NOS RE-'  
PRINT *,'GISTROS, E OS NÚMEROS DEVEEM ESTAR SEPARADOS POR VIRGUL-
```



```
PRINT *, "LAS."
PRINT *, "OS RESULTADOS GERADOS SAO CARREGADOS NO "
PRINT *, "ARQUIVO FOR021.DAT."
PRINT 14
FORMAT (// " DAR <ENTER> PARA CONTINUAR ...")
READ 15, YX
FORMAT(A1)

PRINT *, "DETERMINAR O NUMERO DE BENS:          N="
READ *, N
PRINT *, "DETERMINAR O NUMERO DE ATIVIDADES: M="
READ *, M
PRINT *, "DETERMINAR O NUMERO DE CONSUMIDORES: K1="
READ *, K1

PRINT *, "QUER SABER DOS RESULTADOS DE CADA CICLO MAIOR? (S/V)"
READ 15, YX1
```

```
*****
C PRIMEIRO REGISTRO DO ARQUIVO FOR020.DAT E' O TIPO DE FUNCAC
DE UTILIZADA
*****
```

```
READ(20,16) UTIL
FORMAT( A15 )
```

```
*****
C SEGUNDO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ A
*****
```

```
READ(20,*) ((A(I,J),J=1,N),I=1,K)
```

```
*****
C TERCEIRO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ DE DGTES
K0
*****
```

```
READ (20,*) ((K0(I,J),J=1,N),I=1,K1)
DO 505 J=1,N
  K(J)=0.00
  DO 510 I=1,K1
    K(J)=K(J)+K0(I,J)
  CONTINUE
  K2(J)=K(J)
CONTINUE
```

```
*****
C QUARTO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ PARUTI
*****
```

```
READ (20,*) ((PARUTI (I,J),J=1,N),I=1,K1)
```

```
*****
```



O QUINTO REGISTRO DO ARQUIVO FORQ20.DAT É O VETOR B (SE A
DEMANDA É COBB-DOUGLASS O VETOR B É COMPOSTO DE K1 UNS)

PRINT *, 'SE A DEMANDA É COBB-DOUGLASS O VETOR B =(1.,...,1.)'
PRINT *, 'OU B=(0, ..., 0), SE FOR LEONTIEFF.'

PRINT 14
READ 10, YX

READ (20, *) (b(I), I=1, K1)

ITER=0
KO=0
KOC=0
IS=0.00
ICICLC=0
IT=0
NEVALP=0
NEVALI=0
NPIVOT=0
PRINT *, 'DETERMINAR O TAMANHO DA MALHA: ID='
READ *, ID
ICHAVE=0
ICHAV=1=0
DO 500 I=1, N
PRINT *, 'CHUTE(', I, ')='
READ *, CHUTE(I)
IS=1'+CHUTE(I)
CONTINUE

ALUI VERIFICAMOS SE O CHUTE ESTA BEM DADO: CHUTE(1)+...+CHUTE(N)=ID
IF (IS.AE.ID) THEN
PRINT *, 'ERRO NO FORNECIMENTO DO CHUTE. TENTAR NOVAMENTE'
PRINT *, 'OBSERVACAO: CHUTE(1) ...+CHUTE(N) = ID '
GO TO 20
END IF

DO 512 I=1, N
DO 514 J=1, N
BINV(I, J)=0.00
CONTINUE
BINV(1, I)=1.00
CONTINUE

ALUI É O GERACAO DO SIMPLEX, DA TRIANGULACAO, QUE CONTEM O CHU-
TE

SS(1,1)=0
DO 515 J=1, N
SS(1+1, J) = CHUTE(J)
SS(1, J+1) = 1



CONTINUE

DO 528 I=2,N+1

DO 529 J=2,N+1

SS(I,J) = CHUTE(I-1)

CONTINUE

SS(I,I) = CHUTE(I-1)-1

JCOLSS(I-1) = I

CONTINUE

J1 E' A COLUNA DE SS QUE ENTRA NA BASE

J1 = 1

AGUI CALCULAMOS O ROTULO, ROT, DO VETOR QUE ENTRA NA BASE:
SS(*,J1)

CALL ROTULO (SS,CHUTE,ROT,A,PREC0,DEMAN,N,M,J1,JMAX,IT20,ID,
1 X1,W0,PARUTI,X,J4,NEVALP,NEVALQ)

AGUI CALCULAMOS A COLUNA J3 QUE SAÍ DA BASE E A COLUNA J2 A
SER SUSTITUIDA NA MATRIZ SS PARA GERAR O SIMPLEX ADJACENTE

CALL PIVOTA (PINV,ROT,ROT1,BPS,W,SASAD,JCOLLA,JCOLSS,J1,PIV1
1 ,X1,PIVO,J2,J3,JMAX,IT2),N,J4,NPIVOT)

ITER = ITER+1

IF(ITER-IT.GT.300) GO TO 1600

IF(INT(ITER/10.)*10.EQ.ITER) PRINT *,'ITER=',ITER

IF (ITER.GE.N) GO TO 1640

AGUI CALCULAMOS O SIMPLEX ADJACENTE

JANT = J2 - 1

JPOST = J2 + 1

IF(J2.EQ.1) JANT=N+1

IF(J2.EQ.N+1) JPOST=1

DO 530 I=1,N+1

SS(I,J2)=SS(I,JANT)+SS(I,JPOST)-SS(I,J1)

CONTINUE

J1 = J2

GO TO 1430

IF(SS(1,J2).EQ.1.00) GO TO 1660

GO TO 1610

N9 = 0

DO 535 J=1,N+1

IF(SS(1,J).EQ.0.00) GO TO 539

N9=N9+1

CONTINUE

IF(N9.GT.1) GO TO 1610



LICAMP

AQUI CALCULAMOS O VETOR NIVEIS FINAIS

```
DO 540 J=1,N
  IF (HASAC(J).EQ.0) GO TO 540
  IF (JCOLA(J).EQ.0) GO TO 540
  NIVEL(JCOLA(J)) = W(J)
CONTINUE
```

AQUI CALCULAMOS O VETOR DE PRECOS FINAIS

```
DO 545 I=2,N+1
  SS = 0.00
  DO 550 J=1,N+1
    IF (SS(I,J).EQ.1.00) GO TO 550
    SS=SS+SS(I,J)
  CONTINUE
  PRECO(I-1)=SS/(ID*DELCAT(N))
CONTINUE
```

AQUI CALCULAMOS O VETOR LUCRO

```
DO 555 J=1,M
  Q=0.00
  DO 560 I=1,N
    Q=Q+PRECO(I)*A(I,J)
  CONTINUE
  LUCRO(J)=Q
CONTINUE
```

AQUI CALCULAMOS O VETOR DEMANDA FINAL

```
DO 565 I=1,N
  PRECO(I)=ID*PRECO(I)
CONTINUE

CALL DEMANDA(CUBANA,PRECO,X0,PRAUTI,ID,N,K1,R)
DO 570 I=1,N
  OFERTA(I)=0.00
  DO 575 J=1,M
    OFERTA(I)=OFERTA(I)+A(I,J)*NIVEL(J)
  CONTINUE
  OFERTA(I)=OFERTA(I)+W(I)
CONTINUE
```

AQUI IMPRIMIMOS OS RESULTADOS DO CICLO MAIOR CORRESPONDENTE

ICICLL=ICICLL+1



NICAMP

TEMOS A OPCAO DE NAO IMPRIMIR OS RESULTADOS DE CADA CICLO MAIOR
E IMPRIMIR SOMENTE O RESULTADO FINAL:

```
IF (ICHAVE.E1.1) THEN
  ICICLO=ICICLO-1
ELSE
  IF (YX1.E2.'N') GO TO 128
END IF
PRINT 99,ICICLO
WRITE(21,99) ICICLO
FORMAT(' ***** CICLO MAIOR: ',12,' *****'//)
PRINT 100
FORMAT(' ATIVIDADE',15X,' NIVEL',15X,' LUCRO'//)
WRITE(21,100)
DO 580 J=1,N
  PRINT 105,N+J,NIVEL(J),LUCRO(J)
  WRITE(21,105)N+J,NIVEL(J),LUCRO(J)
CONTINUE
FORMAT (14,15X,F15.7,5X,C15.5)
PRINT 110
WRITE (21,110)
FORMAT(//)
PRINT 115
WRITE(21,115)
FORMAT(' DEM',24X,'DEMANDA',12X,'OFERTA',14X,'RESIDUO'//)
RES=0.00
DO 590 I=1,N
  RESTO(I)=(OFERTA(I)-DEMAN(I))*(OFERTA(I)-DEMAN(I))
  RES=RES+RESTO(I)
  PRINT 120,I,DEMAN(I),OFERTA(I),DSQRT(RESTO(I))
  WRITE (21,120)I,DEMAN(I),OFERTA(I),DSQRT(RESTO(I))
CONTINUE
FORMAT (13,20X,F15.7,5X,F15.7,5X,F15.8)
PRINT 125
WRITE(21,125)
FORMAT(//,' PREÇOS FINAIS'//)
PRINT 130,(PRECO(I)/ID,I=1,N)
WRITE (21,130)(PRECO(I)/ID,I=1,N)
FORMAT(' ',F10.8)
WRITE(21,135)
FORMAT(//////////)
PRINT *,'NUMERO DE ITERAÇÕES=',ITER
PRINT *,'RES=',DSQRT(RES)
PRINT *,'NEVALP=',NEVALP,'NEVALQ=',NEVALQ,'NPIVOT=',NPIVOT
PRINT 14
READ 15,YX

IF (ICHAVE1.E3.1) GO TO 1001
IF (ICHAVE.E4.1) GO TO 1004

IF(YX1.NE.'N') GO TO 129
RES=0.00
DO 597 I=1,N
  RESTO(I)=(DEMAN(I)-OFERTA(I))*(DEMAN(I)-OFERTA(I))
  RES=RES+RESTO(I)

IF(DSQRT(RES).LE.1.E-10).OR.(1.E0/ID.LE.1.E-11)) THEN
  PRINT *,'***** PRECISAO ATINGIDA *****'
  PRINT 14
```



```
READ 15,YX
IF(YX1.EQ.'N.') THEN
  ICHAVE=1=1
  ICHAVE=1
  GO TO 1000
ELSE
  GO TO 1001
END IF
END IF
K00=1
K0=0
DO 601 I=1,N+1
  DO 602 J=1,N+1
    SSR=0(I,J)=SS(I,J)
  CONTINUE
CONTINUE
CIO=IC
ID=3.00*IO
DO 603 J=1,N
  CNIVEL(J)=NIV+L(J)
  K0=K0+1
  IF((K0.GT.M).OR.(K00.EQ.0)) GO TO 1002
  IF(K0.NE.1) THEN
    PRINT *, '***** COM A COLUNA',JCCLSS(K0-1), 'DE SS NAO CO
1  NSEQUENTES CONVERGENCIA, FUGAMOS PARA A COLUNA',JCCLSS(K0)
  END IF
  JCCLSS=JCCLSS(K0)
  IT=ITER
  DO 605 I=1,N
    COUT(I)=SSREL(I+1,JCCLSS)*3.00
  CONTINUE
  DO 610 I= 1,N
    DO 611 J=1,N
      SINV(I,J)=0.00
    CONTINUE
    SINV(1,I)=1.00
    W(I)=K2(I)
    BASAB(I)=0
    JCCLA(I)=0
  CONTINUE
  DO 620 J=1,N
    NIVEL(J)=0.00
  CONTINUE
  GO TO 15
```

```
*****
AQUI IMPRIMIMOS OS DADOS DO PROBLEMA
*****
```

```
1  WRITE(21,140)
  FORMAT(1F1, ' ***** DADOS DO PROBLEMA *****')
  WRITE(21,145)N,K1,M
  FORMAT(' N DE BARS=',I3,EX,' N DE CONSUMOS=',I3,EX,' N DE ATIVOS='
1  ,I3,/)
  WRITE(21,148) UTIL
  FORMAT(' UTILIDADE ',F15,/)
  WRITE(21,150)
  FORMAT(' SAT-11 DE ATIVIDADES: /')
  DO 640 I=1,N
```




MACAMP

```

WRITE(21,165) (A(I,J),J=1,N)
CONTINUE
FORMAT(<<*>(2X,F9.2))
WRITE(21,166)
FORMAT(//)
WRITE(21,167)
FORMAT(' MATRIZ DE RECURSOS INICIAIS: '//)
DO 545 I=1,K1
WRITE(21,170) (V0(I,J),J=1,N)
CONTINUE
FORMAT(<N>(2X,F9.2))
WRITE(21,168)
WRITE(21,175)
FORMAT(' MATRIZ DE PARAMETROS DE UTILIDADES: '//)
DO 550 I=1,K1
WRITE(21,180) (PARUT(I,J),J=1,N)
CONTINUE
FORMAT(<A>(2X,F9.2))
WRITE(21,169)
WRITE(21,185)
FORMAT(' VETOR DE PARAMETROS B: '//)
WRITE(21,190) (B(I),I=1,K1)
FORMAT(<K1>(2X,F9.2))
STOP
3 IF(IGICLL.EQ.0) GO TO 1003
PRINT *, 'COM O ULTIMO CHUTE SORADO NAO CONSEGUIMOS CONVERGIR.'
PRINT *, ' '
PRINT *, 'OS RESULTADOS OBTIDOS SAO OS SEGUINTE:'
PRINT 14
READ 15, YX
ICHAVE=1
ID=CID
DO 555 I=1,N+1
DO 555 J=1,N+1
SS(I,J)=SSRED(I,J)
DO 560 J=1,N
NIVEL(J)=CNIVEL(J)
CL TO 30
5 PRINT *, '*****NAO CONSEGUIMOS CONVERGENCIA.MUDAR CHUTE*****'
GO TO 20
4 PRINT *, 'QUEM CONTINUAR? (S/N): '
READ 15, YX2
IF(YX2.EQ.'N') GO TO 1001
PRINT *, '***** MUDAR DE CHUTE *****'
WRITE (21,200)
FORMAT('1',70(' ')/14X,'AQUI REINICIAMOS O ALGORITMO COM NOVO CH
1UTE',14X/70(' '))
GO TO 20
END

```

```

SUBROUTINE ROTULO(SS,CHUTE,ROT,A,PREC,DEMAN,N,M,J1,JMAX,IT20,
1 ID,K1,NO,PARUTI,I,J4,NEVALF,NEVALC)

```



```

REAL*8 SS(20,20),CHUTE(10),ZC
REAL*8 RCT(20),A(20,20),PREC(20),DEMAN(20),WC(20)
REAL*8 PARUTI(20,20),S(20),SCMA,IV
J4=0
IF(SS(1,J1).NE.0.00) GO TO 5140
IT20=0
IT9=0
DO 5050 I=1,N
    IF(IT9.EQ.1) GO TO 5087
    IF(SS(I+1,J1).LT.CHUTE(I)) GO TO 5060
    GO TO 5030
    J4=1
    IT9=1
    RCT(1)=0.00
CONTINUE
    RCT(J4)=1.00
    GO TO 5070
    IT7=0
    IT20=0
    DO 5230 I=1,N
        IF(IT7.EQ.1) GO TO 5220
        IF(SS(I+1,J1).NE.0.00) GO TO 5230
        J4=I
        DO 5210 J=1,M
            RCT(J)=0.00
        CONTINUE
        RCT(J4)=1.00
        IT7=1
    CONTINUE
    IF(IT7.EQ.0) GO TO 5390
    GO TO 5370
    IV= -1.037
    DO 5470 J=1,M
        SCMA=0.00
        DO 5420 I=1,N
            SCMA=SS(I+1,J1)*A(I,J)+SCMA
        CONTINUE
        IF(SCMA.GT.IV) GO TO 5450
        GO TO 5470
        JMAX=J
        IV=SCMA
    CONTINUE
    IF(IV.GT.0.10) GO TO 5530

    NEVALP=NEVALP+1
    DO 5480 I=1,N
        PREC(I)=SS(I+1,J1)
    CONTINUE
    CALL DEMAND(DEMAN,PREC,WC,PARUTI,IO,A,K1,S)
    DO 5520 I=1,N
        RCT(I)=DEMAN(I)
    CONTINUE
    GO TO 5570

    NEVALS=NEVALS+1
    IT20=1
    DO 5530 I=1,N
        RCT(I)= -A(I,JMAX)
    CONTINUE
    RETURN

```



END

```
*****
*****
      SUBROUTINE PIVOTACHINV,ROTI,ROTI1,EPS,X,BASAD,JCOLA,JCOLSS,J1,
1  PIV1,N1,PIV0,J2,J3,JMAX,IT20,N,J4,NPIVOT)
```

```
*****
*****
      REAL*8 XINV(20,20),ROTI(20),ROTI1(20),EPS(20),W(20),PIV1(20)
1  ,W1(20),PIV0(10,10),X0,X1,XMIN,TOL,DELTA,ROTM,PHI,PHI1
      INTEGER BASAD(20),JCOLA(20),JCOLSS(20)
```

```
      IF (J4.NE.0) THEN
10  DO 5080 I=1,N
        ROT(I)=BINV(I,J4)
      ELSE
        NPIVOT=NPIVOT+1
        DO 5080 I=1,N
          X0=0.00
          DO 6060 J=1,N
            X0=X0+XINV(I,J)*ROTI(J)
J          CONTINUE
          ROT1(I)=X0
J        CONTINUE
10  DO 6085 I=1,N
          ROT(I)=ROTI(I)
S        CONTINUE
      END IF
```

```
      TOL =1.0-8
      DELTA = 9.0-4
      PHI = 1.037
      DO I=1,N
        IF(ROTI(I).GE.TOL) THEN
          PHI1 = (W(I)+DELTA)/ROTI(I)
          IF(PHI1.LT.PHI) THEN
            PHI = PHI1
          END IF
        END IF
```

```
      END DO
      ROTM= -1.037
      DO I=1,N
        IF((ROTI(I).GE.TOL).AND.(W(I)/ROTI(I).LE.PHI)) THEN
          IF(ROTI(I).GE.ROTM) THEN
            ROTM=ROTI(I)
            J3=I
          END IF
        END IF
      END DO
```

```
      BASAD(J3)=0
      JCOLA(J3)=0
      IF(IT20.EQ.0) GO TO 5830
      BASAD(J3)=1
```



```
JCOLL(J3)=J+47
J2=JCOLLS(J3)
JCOLLS(J3)=J1
```

```
DO 5410 I=1,N
PIV1(I)=-A(1,I)/A(1,J3)
```

```
PIV1(J3)=1.00/A(1,J3)
```

```
DO 5430 I=1,N
W1(I)=W(I)+PIV1(I)*W(J3)
W1(J3)=PIV1(J3)*W(J3)
```

```
DO 5433 I=1,N
```

```
W(I)=W1(I)
```

```
DO 5510 J=1,N
```

```
DO 5500 I=1,N
```

```
PIV2(I,J)=MINV(I,J)+PIV1(I)*MINV(J3,J)
```

```
CONTINUE
```

```
PIV2(J3,J)=PIV1(J3)*MINV(J3,J)
```

```
CONTINUE
```

```
DO 5550 I=1,N
```

```
DO 5550 J=1,N
```

```
MINV(I,J)=PIV2(I,J)
```

```
CONTINUE
```

```
CONTINUE
```

```
RETURN
```

```
END
```

```
*****
*****
SUBROUTINE DEMAN(DEMAN,PREC,WO,PARUT,ID,N,K1,B)
```

```
*****
*****
REAL*8 DEMAN(20),PREC(20),WO(20,20),PARUT(20,20),B(20),S1,
```

```
1 S2,S20,S30,S40,ID
```

```
DO 5120 J=1,N
```

```
DEMAN(J)=0.00
```

```
DO 5120 I=1,K1
```

```
S2=0.00
```

```
S1=0.00
```

```
DO 5080 K=1,N
```

```
S1=S1+PREC(K)*WO(I,K)/ID
```

```
S30=(PREC(K)/ID)**(1.00-B(I))
```

```
S40=S30*PARUT(I,K)
```

```
S2=S2+S40
```

```
CONTINUE
```

```
S1=PARUT(I,J)*S1
```

```
S2=(PREC(J)/ID)**B(I)*S2
```

```
DEMAN(J)=DEMAN(J)+S1/S2
```

```
CONTINUE
```

```
CONTINUE
```

```
RETURN
```

```
END
```

REFERÊNCIAS

- BROADIE, M.N., "An Introduction to the Octahedral Algorithm", Presented at the Workshop on the Application and Solution of Economic Equilibrium Models, [June, 1984].
- BROUWER, L.E., "Über abbildungen von Mannigfaltigkeiten", Math. Ann., 71 [1912], 97-115.
- DEBREU, G., "Continuity Properties of Paretian Utility", International Economic Review, 5, [1964], 285-293.
- EAVES, B.C., "Computing Kakutani Fixed Points", SIAM J. of App. Math., 21,2 [1971], 236-244.
- EAVES, B.C., "Homotopies for Computation of Fixed Points", Math. Programming, 3 [1972], 1-22.
- GARCIA, C.B. e ZANGWILL, W.I., "Pathways to Solutions, Fixed Points and Equilibria", Prentice-Hall, Series in Computational Mathematics, Cleve Moler, Advisor, [1981].
- HANSEN, T. e SCARF, H.E., "Computation of Economic Equilibria", Yale University Press, New Haven [1973].

- MERRILL, O.H., "Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point to Set Mappings", Ph.D. Dissertation, Department of Industrial Engineering, University of Michigan [1972].
- SCARF, H.E., "The Approximation of Fixed Points of Continuous Mappings", SIAM J. of App. Math., 15 [1967], 1328-1343.
- SCARF, H.E., "The Computation of Equilibrium Prices", em H. Scarf e J. Showen eds., Applied General Equilibrium Analysis, Cambridge University Press, Cambridge [1984].
- SAIGAL, R., "On the Convergence Rate of Algorithms for Solving Equations that are Based on Methods of Complementary Pivoting", Math. of O.R., 2 [1977], 108-124.
- SAIGAL, R. e TODD, M.J., "Efficient Acceleration Techniques for Fixed Point Algorithms", SIAM J. of Numerical Analysis, 15 [1978], 997-1007.
- SCARAMUCCI, J. e ARICA, J., "Métodos Homotópicos para a Determinação de Equilíbrios Econômicos". Anais do IX Congresso Nacional de Matemática Aplicada e Computacional (SBMAC), [1986], 373-381, Brasília.

- TODD, M.J., "On Triangulations for Computing Fixed Points". Mathematical Programming, 10 [1976], 322-346.
- TODD, M.J., "The Computation of Fixed Points and Applications" ,
Lecture Notes in Economics and Mathematical Systems, 124 ,
Springer-Verlag, Berlin .Heidelberg, N.Y. [1976].
- TODD, M.J., "Efficient Methods of Computing Economic Equilibria" ,
em H. Scarf e J. Showen eds., Applied General Equilibrium
Analysis, Cambridge University Press, Cambridge [1984].
- VARIAN, H.R., "Microeconomic Analysis", W.W. Norton & Company, New
York, London [1978].
- AWONIYI, S.A. e TODD, M.J., "An Efficient Simplicial Algorithm for
Computing a Zero of a Convex Union of Smooth Functions". Ma-
thematical Programming, 25 [1983], 83-108.



***** DATES OF PRESENTATION *****

* OF DAYS = 6 * OF DAYS = 6 * OF DAYS = 6 * OF DAYS = 6

DATE OF PRESENTATION

DATE OF PRESENTATION

4.00	5.00	1.60	1.60	1.60	1.60	0.60	7.00	8.00
-5.00	-5.00	-2.00	-2.00	-2.00	-2.00	-1.00	-4.00	-5.00
-2.00	-1.00	-2.00	-2.00	-2.00	-2.00	0.00	-2.00	-2.00
-1.00	-2.00	-2.00	-2.00	-2.00	-2.00	0.00	-1.00	-2.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.00	3.50	2.00	2.00	2.00	2.00	0.00	0.00	0.00

DATE OF PRESENTATION

0.00	3.00	1.00	0.10	0.10	0.10	0.00	1.00
0.00	0.00	0.10	0.10	0.10	0.10	0.00	2.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.50
0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.00

DATE OF PRESENTATION

4.00	0.00	0.00	0.00	0.00	0.00	2.00	3.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

DATE OF PRESENTATION

DATE OF PRESENTATION



*
* ALGORITHM OF CICLL *
*

***** CICLL ALGO: 21 *****

ALGORITHM	NIVEL	LUCAO
7	0.0273681	-0.975113018-11
8	0.6928287	0.334599027-11
9	2.3211318	0.191199453-11
10	1.0251870	-0.390395502-11
11	0.0000000	-0.724970705-01
12	0.0000000	-0.203043348-01
13	0.1452107	0.95599743E-11
14	0.0000000	-0.336909507+00

11	DEMANDA	DEBITA	RESIDUO
1	11.2670029	11.2670029	0.555613338-10
2	0.0000000	0.0000000	0.22204460E-15
3	1.0133722	1.0133722	0.236477505-12
4	2.1326040	2.1326040	0.392497232-10
5	22.1603709	22.1603709	0.20956541E-09
6	10.7716824	10.7716824	0.19471513E-09

ALGO: 14418

.21619735
.21-29850
.19253316
.07623413
.11527973
.18385131

PROBLEMA 2.

O tempo de processamento foi de 5 segundos, com um total de 3853 iterações e 39 avaliações da função demanda. O tamanho inicial da malha foi 50 e o CHUTE = (15,5,12,8,10). A convergência foi atingida no ciclo maior 21. Os resultados são mostrados a continuação.



***** TABELA DE PARÂMETROS *****

Nº DE BENS= 6 Nº DE CONSIDERES= 2 Nº DE ATIVIDADES= 6

UTILIDADE= CBBB-BELUGLASS

TABELA DE ATIVIDADES:

100.00	100.00	0.00	0.00	0.00	0.00
-10.00	-11.00	100.00	100.00	100.00	-40.00
-20.00	-21.00	-40.00	-50.00	-50.00	0.00
-37.00	-20.00	-50.00	-40.00	-50.00	0.00
-5.00	-10.00	-5.00	-5.00	-10.00	100.00

TABELA DE RECURSOS INICIAIS:

0.00	1.00	10000.00	0.00	0.00
0.00	0.00	0.00	10000.00	500.00

TABELA DE PARÂMETROS DE UTILIDADE:

0.00	0.00	0.10	0.00	0.00
0.00	0.40	0.00	0.00	0.00

TABELA DE PARÂMETROS 1:

1.00	1.00
------	------



*
* ALUMINIO DE V-SILL *
*

***** CICLO 44101: 21 *****

ACTIVIDADES	NIVEL	LUCRO
0	0.0000000	-0.11300230E+00
1	105.4421171	-0.22381035E-09
2	141.5478425	0.10324536E-09
3	0.0000000	-0.22617327E+00
10	37.0333131	0.11157746E-09
11	15.5034131	-0.20649293E-09

ITEM	DEMANDA	OFERTA	RESIDUO
1	10544.211113	10544.2315113	0.20447427E-07
2	13269.7142357	13269.7142357	0.61656708E-09
3	1500.0000000	1500.0000000	0.27490596E-07
4	0.0000000	0.0000000	0.22737368E-12
5	0.0000000	0.0000000	0.65353932E-12

RESIDUO -14415

- .105446829
- .113002068
- .13701031
- .13067844
- .08755527

PROBLEMA 3.

O tempo de processamento foi de 4 segundos, com um total de 1884 iterações e 115 avaliações da função demanda. O tamanho inicial da malha foi 50 e o CHUTE = (12,8,15,5,10) A convergência foi atingida no ciclo maior 21. Os resultados são mostrados a continuação.



***** DADOS DE APLICACAO *****

NUMERO= 1 N DE CONSECUCOES= 2 N DE ATIVOS= 4

UTILIDADE COORDENADAS

TABELA DE ATIVIDADES:

100.00	100.00	0.00	0.00	0.00	0.00
-10.00	-10.00	100.00	100.00	100.00	-100.00
-50.00	-50.00	-45.00	-52.00	-55.00	0.00
-17.00	-20.00	-50.00	-40.00	-30.00	0.00
-1.00	-10.00	-5.00	-5.00	-15.00	100.00

TABELA DE RECURSOS INICIAIS:

0.00	0.00	10000.00	0.00	0.00
0.00	0.00	0.00	10000.00	300.00

TABELA DE PARAMETROS DE UTILIDADE:

0.50	0.50	0.10	0.00	0.00
0.50	0.40	0.00	0.00	0.00

TABELA DE PARAMETROS N:

1.00	1.00
------	------



*
* ALBERTINE DE MEAFILL *
*

***** CIRCLE MEAFILL: 21 *****

ALBERTINE	ALBERTINE	ALBERTINE
6	113.4000000	0.25003717E-09
7	0.0000000	-0.40000000E+00
8	71.1330000	0.44740351E-09
9	64.4782500	0.41757889E-09
10	0.0000000	-0.60000000E+00
11	4.7167500	-0.11471850E-09

ALBERTINE	ALBERTINE	ALBERTINE	ALBERTINE
1	11339.9999999	11340.0000000	0.13272761E-06
2	13959.9999997	13960.0000000	0.29197928E-06
3	1690.0000000	1690.0000000	0.22737364E-12
4	0.0000000	0.0000000	0.43474735E-12
5	0.0000000	0.0000000	0.34106051E-12

***** ALBERTINE



PROGRAM MERRILL

ALGORITMO DE MERRILL PARA O CALCULO DE PRECOS DE EQUILIBRIO
NUMA ECONOMIA WALRASIANA COM PRODUCAO DETERMINADA POR ATIVI-
DADES LINEARES

SS E' A MATRIZ DO N-SIMPLEX COMPLETO CONSIDERADO EM CADA ITE-
RACAO. AS COLUNAS SAO OS NUMERADORES DOS VERTICES. O NA PRI-
MEIRA LINHA INDICA QUE O RESTO DA COLUNA E' UM VERTICE DO SIM-
PLEX ORIGINAL; 1 INDICA QUE O VERTICE E' ARTIFICIAL.

WD CORRESPONDE A CESTA INICIAL, CHUTE AO CHUTE INICIAL, PARU-
TI A MATRIZ DE PARAMETROS DE UTILIDADES, B AO VETOR DE PARAME-
TROS E A MATRIZ A REPRESENTA AS ATIVIDADES LINEARES.

```
1 REAL*8 RESTO(20),ROT(20),S(20),EPS(20),LUCRO(20),OFERTA(20),
2 NIVEL(20),PARUTI(20,20),DEMAN(20),A(20,20),PRECO(20),BINV(20,
3 20),PIVO(20,20),PIV1(20),W0(20,20),RET1(20),W1(20),W2(20),
X0,X1,Q,S9,W(20)
INTEGER JCOLSS(20),JCOLA(20),BASAD(20)
REAL*8 SSRED(20,20),SS(20,20),CHUTE(20),ID,IS,RES,CID,CNIVEL(20)
CHARACTER YX,YX1,YX2,UTIL*15
```

```
PRINT 12
FORMAT (//////)
PRINT *,'*****'
PRINT *,'*'
PRINT *,'ALGORITMO DE MERRILL'
PRINT *,'*'
PRINT *,'*****'
PRINT 13
FORMAT(//)
```

```
WRITE(21,*) '*****'
WRITE(21,*) '*'
WRITE(21,*) 'ALGORITMO DE MERRILL'
WRITE(21,*) '*'
WRITE(21,*) '*****'
WRITE(21,13)
```

```
PRINT *,'OBSERVACAO:'
PRINT *,'
PRINT *,'ESTE PROGRAMA PRECISA UM ARQUIVO DE DADOS,'
PRINT *,'FDR020.DAT, COM CINCO REGISTROS. O PRIMEIRO CONTEM O'
PRINT *,'TIPO DE FUNCAO DE UTILIDADE (COBB-DOUGLASS,LEONTIEFF,'
PRINT *,'ESC), O SEGUNDO A MATRIZ DE ATIVIDADES, O TERCEIRO A '
PRINT *,'MATRIZ DE COTES (GU RECURSOS INICIAS), O QUARTO A MA-'
PRINT *,'TRIZ DE PARAMETROS DE UTILIDADES E O QUINTO O VETOR B'
PRINT *,'DE PARAMETROS (ESTE VETOR DEVE SER B=(1,...,1),SE A U-'
PRINT *,'TILIDADE FOR COBB-DOUGLASS, OU B=(0,...0), SE FOR LE-'
PRINT *,'ONTIEFF).'
PRINT *,'
PRINT *,'AS MATRIZES ESTAO DADAS POR FILAS,MES RE-'
PRINT *,'GISTROS,E OS NUMEROS DEVEM ESTAR SEPARADOS POR VIRGU-
```




```
PRINT *, 'LAS.'
PRINT *, 'OS RESULTADOS GERADOS SAO CARREGADOS NO '
PRINT *, 'ARQUIVO FOR021.DAT.'
PRINT 14
FORMAT (// ' DAR <ENTER> PARA CONTINUAR ...')
READ 15, YX
FORMAT(A1)

PRINT *, 'DETERMINAR O NUMERO DE BENS:          N='
READ *, N
PRINT *, 'DETERMINAR O NUMERO DE ATIVIDADES: M='
READ *, M
PRINT *, 'DETERMINAR O NUMERO DE CONSUMIDORES: K1='
READ *, K1

PRINT *, 'QUER SAIDA DOS RESULTADOS DE CADA CICLO MAIOR? (S/N)'
READ 15, YX1
```

```
*****
O PRIMEIRO REGISTRO DO ARQUIVO FOR020.DAT E' O TIPO DE FUNCAO
DE UTILIDADE
*****
```

```
READ(20,16) UTIL
FORMAT( A15 )
```

```
*****
O SEGUNDO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ A
*****
```

```
READ(20,*) ((A(I,J),J=1,M),I=1,N)
```

```
*****
O TERCEIRO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ DE COTES
*0
*****
```

```
READ (20,*) ((W0(I,J),J=1,N),I=1,K1)
DO 505 J=1,N
  W(J)=0.00
DO 510 I=1,K1
  W(J)=W(J)+W0(I,J)
CONTINUE
W2(J)=W(J)
CONTINUE
```

```
*****
O QUARTO REGISTRO DO ARQUIVO FOR020.DAT E' A MATRIZ PARUTI
*****
```

```
READ (20,*) ((PARUTI (I,J),J=1,N),I=1,K1)
```

```
*****
```



O QUINTO REGISTRO DO ARQUIVO FORO20.DAT E' O VETOR B (SE A
DEMANDA E' COBB-DOUGLASS O VETOR B E' COMPOSTO DE K1 UNS)

PRINT *, 'SE A DEMANDA E COBB-DOUGLASS O VETOR B =(1.,...,1.)'
PRINT *, 'OU B=(0, ..., 0), SE FOR LEONTIEFF.'

PRINT 14
READ 15, YX

READ (20, *) (B(I), I=1, K1)

ITER=0
K0=0
K00=0
IS=0.00
ICICLO=0
IT=0
NEVALF=0
NEVALS=0
NPIVOT=0

PRINT *, 'DETERMINAR O TAMANHO DA MALHA: ID='

READ *, ID
ICHAVE=0
ICHAVE1=0

DO 500 I=1, N

PRINT *, 'CHUTE(', I, ')='
READ *, CHUTE(I)
IS=I*CHUTE(I)

CONTINUE

AQUI VERIFICAMOS SE O CHUTE ESTA BEM DADO: CHUTE(1)+...+CHUTE(N)=ID

IF (IS.NE.ID) THEN

PRINT *, 'ERRO NO FORNECIMENTO DO CHUTE. TENTAR NOVAMENTE'

PRINT *, 'OBSERVACAO: CHUTE(1) ...+CHUTE(N) = ID '

GO TO 20
END IF

DO 512 I=1, N

DO 514 J=1, N

BINV(I, J)=0.00

CONTINUE

BINV(I, I)=1.00

CONTINUE

AQUI E' GERADO O SIMPLEX , DA TRIANGULACAO, QUE CONTEM O CHU-
TE

SS(1,1)=0

DO 515 I=1, N

SS(I+1,1) = CHUTE(I)

SS(1, I+1) = 1



CONTINUE

GO 520 I=2,N+1

GO 525 J=2,N+1

SS(I,J) = CHUTE(I-1)

CONTINUE

SS(I,I) = CHUTE(I-1)-1

JCOLSS(I-1) = I

CONTINUE

J1 É A COLUNA DE SS QUE ENTRA NA BASE

J1 = 1

AQUI CALCULAMOS O ROTULO, ROT, DO VETOR QUE ENTRA NA BASE:
SS(*,J1)

GO 1 CALL ROTULO (SS,CHUTE,ROT,A,PRECO,DEMAN,N,M,J1,JMAX,IT20,ID,
1 K1,W0,PARUTI,2,J4,NEVALP,NEVALQ)

AQUI CALCULAMOS A COLUNA J3 QUE SAI DA BASE E A COLUNA J2 A
SER SUSTITUIDA NA MATRIZ SS PARA GERAR O SIMPLEX ADJACENTE

GO 1 CALL PIVOTA (BINV,ROT,ROT1,EPS,X,BASAD,JCOLA,JCOLSS,J1,PIV1
1 ,W1,PIV0,J2,J2,JMAX,IT20,N,J4,NPIVOT)

ITER = ITER+1

IF(ITER-IT.GT.300) GO TO 1600

IF(INT(ITER/10.)*10.EQ.ITER) PRINT *, 'ITER=', ITER

IF (ITER.GE.N) GO TO 1540

AQUI CALCULAMOS O SIMPLEX ADJACENTE

GO JANT = J2 - 1

JPOST = J2 + 1

IF(J2.EQ.1) JANT=N+1

IF(J2.EQ.N+1) JPOST=1

GO 530 I=1,N+1

SS(I,J2)=SS(I,JANT)+SS(I,JPOST)-SS(I,J2)

CONTINUE

J1 = J2

GO TO 1430

GO IF(SS(1,J2).EQ.1.D0) GO TO 1560

GO TO 1510

GO N9 = 0

GO 535 J=1,N+1

IF(SS(1,J).EQ.0.D0) GO TO 535

N9=N9+1

CONTINUE

IF(N9.GT.1) GO TO 1510



AQUI CALCULAMOS O VETOR NIVEIS FINAIS

```
DO 540 J=1,N
  IF (BASAC(J).EQ.0) GO TO 540
  IF(JOCLA(J).EQ.0) GO TO 540
  NIVEL(JOCLA(J)) = W(J)
```

CONTINUE

AQUI CALCULAMOS O VETOR DE PRECOS FINAIS

```
DO 545 I=2,N+1
  S9 =0.00
  DO 550 J=1,N+1
    IF(SS(1,J).EQ.1.00) GO TO 550
    S9=S9+SS(I,J)
```

CONTINUE

PREC0(I-1)=S9/(ID*DFLCAT(N))

CONTINUE

AQUI CALCULAMOS O VETOR LUCRO

```
DO 555 J=1,M
  Q=0.00
  DO 560 I=1,N
    Q=Q+PREC0(I)*A(I,J)
  CONTINUE
  LUCRO(J)=Q
```

CONTINUE

AQUI CALCULAMOS O VETOR DEMANDA FINAL

```
DO 565 I=1,N
  PRECO(I)=ID*PREC0(I)
CONTINUE
```

CALL DEMAND(DEMAN,PREC0,W0,PARUT1,ID,N,K1,B)

```
DO 570 I=1,N
  OFERTA(I)=0.00
  DO 575 J=1,M
    OFERTA(I)=OFERTA(I)+A(I,J)*NIVEL(J)
  CONTINUE
  OFERTA(I)=OFERTA(I)+W2(I)
CONTINUE
```

AQUI IMPRIMIMOS OS RESULTADOS DO CICLO MAIOR CORRESPONDENTE

CICLO=CICLO+1



TEMOS A OPCAO DE NAO IMPRIMIR OS RESULTADOS DE CADA CICLO MAIOR
E IMPRIMIR SOMENTE O RESULTADO FINAL:

```
IF (ICHAVE.EQ.1) THEN
ICICLU=ICICLU-1
ELSE
IF (YX1.EQ.'N') GO TO 138
END IF
PRINT 99,ICICLU
WRITE(21,99) ICICLU
FORMAT(' ***** CICLO MAIOR: ',I2,' *****'//)
PRINT 100
FORMAT(' ATIVIDADE',15X,'NIVEL',15X,'LUCRO'//)
WRITE(21,100)
DO 580 J=1,M
PRINT 105,N+J,NIVEL(J),LUCRO(J)
WRITE(21,105)N+J,NIVEL(J),LUCRO(J)
CONTINUE
FORMAT (I4,15X,F15.7,5X,E15.8)
PRINT 110
WRITE (21,110)
FORMAT(//)
PRINT 115
WRITE(21,115)
FORMAT(' SEM',24X,'DEMANDA',15X,'OFERTA',14X,'RESIDUO'//)
RES=0.D0
DO 590 I=1,N
RESTO(I)=(OFERTA(I)-DEMAN(I))*(OFERTA(I)-DEMAN(I))
RES=RES+RESTO(I)
PRINT 120,I,DEMAN(I),OFERTA(I),DSQRT(RESTO(I))
WRITE (21,120)I,DEMAN(I),OFERTA(I),DSQRT(RESTO(I))
CONTINUE
FORMAT (I3,20X,F15.7,5X,F15.7,5X,E15.8)
PRINT 125
WRITE(21,125)
FORMAT(//,' PRECOS FINAIS'//)
PRINT 130,(PRECO(I)/ID,I=1,N)
WRITE (21,130)(PRECO(I)/ID,I=1,N)
FORMAT(' ',F10.8)
WRITE(21,135)
FORMAT(//////////)
PRINT *,'NUMERO DE ITERACOES=',ITER
PRINT *,'RES=',DSQRT(RES)
PRINT *,'NEVALF=',NEVALF,'NEVALD=',NEVALD,'NPIVOT=',NPIVOT
PRINT 14
READ 15,YX

IF (ICHAVE1.EQ.1) GO TO 1001
IF (ICHAVE.EQ.1) GO TO 1004

IF(YX1.NE.'N') GO TO 139
RES=0.D0
DO 597 I=1,N
RESTO(I)=(DEMAN(I)-OFERTA(I))*(DEMAN(I)-OFERTA(I))
RES=RES+RESTO(I)

IF((DSQRT(RES)).LE.1.D-10).OR.(1.D0/ID.LE.1.D-11)) THEN
PRINT *,'***** PRECISAO ATINGIDA *****'
PRINT 14
```



```
READ 15,YX
IF(YX1.EQ.'N') THEN
  ICHAVE1=1
  ICHAVE=1
  GO TO 1000
ELSE
  GO TO 1001
END IF
END IF
K00=1
K0=0
DO 601 I=1,N+1
  DO 602 J=1,N+1
    SSRED(I,J)=SS(I,J)
  CONTINUE
CONTINUE
CID=ID
ID=3.00*ID
DO 603 J=1,M
  CNIVEL(J)=NIV=L(J)
K0=K0+1
IF((K0.GT.N).OR.(K00.EQ.0)) GO TO 1002
IF(K0.NE.1)THEN
  PRINT *, '***** COM A COLUNA',JCOLSS(K0-1), 'DE SS NAO CO
1 NSEGUIMOS CONVERGENCIA, MUDAMOS PARA A COLUNA',JCOLSS(K0)
END IF
JCLSS=JCOLSS(K0)
IT=ITER
DO 605 I=1,N
  CHUTE(I)=SSRED(I+1,JCLSS)*3.00
CONTINUE
DO 610 I= 1,N
  DO 615 J=1,M
    SINV(I,J)=0.00
  CONTINUE
  SINV(I,I)=1.00
  W(I)=W2(I)
  BASAD(I)=0
  JCOLA(I)=0
CONTINUE
DO 620 J=1,M
  NIVEL(J)=0.00
CONTINUE
GO TO 25
```

```
*****
AQUI IMPRIMIMOS OS DADOS DO PROBLEMA
*****
```

```
1 WRITE(21,140)
  FORMAT(1H1, ' ***** DADOS DO PROBLEMA *****')
  WRITE(21,145)N,K1,M
  FORMAT(' * DE BENS=',I3,5X, ' * DE CONSUMORS=',I3,5X, ' * DE ATIVDS='
1 ,I3,/)
  WRITE(21,148) UTIL
  FORMAT(' UTILIDADE ',A13,/)
  WRITE(21,150)
  FORMAT(' MATRIZ DE ATIVIDADES:')
  DO 640 I=1,N
```



VICAMP

```

WRITE(21,155) (A(I,J),J=1,M)
CONTINUE
FORMAT (<M>(2X,F9.2))
WRITE(21,160)
FORMAT(//)
WRITE(21,165)
FORMAT(' MATRIZ DE RECURSOS INICIAS: '/')
DO 645 I=1,K1
    WRITE(21,170) (W0(I,J),J=1,N)
CONTINUE
FORMAT(<N>(2X,F9.2))
WRITE(21,160)
WRITE(21,175)
FORMAT(' MATRIZ DE PARAMETROS DE UTILIDADE: '/')
DO 650 I=1,K1
    WRITE(21,180) (PARUTI(I,J),J=1,N)
CONTINUE
FORMAT(<N>(2X,F9.2))
WRITE(21,160)
WRITE(21,185)
FORMAT(' VETOR DE PARAMETROS B: '/')
WRITE(21,190)(B(I),I=1,K1)
FORMAT(<K1>(2X,F9.2))
STOP
IF(ICICLC.EQ.0) GO TO 1003
PRINT *, 'COM O ULTIMO CHUTE GERADO NAO CONSEGUAMOS CONVERGIR.'
PRINT *, ' '
PRINT *, 'OS RESULTADOS OBTIDOS SAO OS SEGUINTEs:'
PRINT 14
READ 15, YX
ICHAVE=1
IC=CID
DO 655 I=1,N+1
    DO 655 J=1,N+1
        SS(I,J)=SSRED(I,J)
DO 660 J=1,M
    NIVEL(J)=CNIVEL(J)
GO TO 30
PRINT *, '*****NAO CONSEGUAMOS CONVERGENCIA.MUDAR CHUTE*****'
GO TO 20
PRINT *, 'QUER CONTINUAR? (S/N): '
READ 15, YX2
IF(YX2.EQ.'N') GO TO 1001
PRINT *, '***** MUDAR DE CHUTE *****'
WRITE (21,200)
FORMAT('1',70('*')/14X,'AQUI REINICIAMOS O ALGORITMO COM NOVO CH
UTE',14X/70('*'))
GO TO 20
END
```

```

SUBROUTINE RGTULC(SS,CHUTE,ROT,A,PREC0,DEMAN,K,M,J1,JMAX,IT20,
1 ID,K1,W0,PARUTI,B,J4,NEVALF,NEVALC)
```



IICAMP

```
REAL*8 SS(20,20),CHUTE(20),ID
REAL*8 RDT(20),A(20,20),PRECC(20),DEMAN(20),W0(20)
REAL*8 PARUTI(20,20),E(20),SCMA,IV
J4=0
IF(SS(1,J1).EQ.0.00) GO TO 5140
IT20=0
IT9=0
DO 5090 I=1,N
    IF(IT9.EQ.1) GO TO 5080
    IF(SS(I+1,J1).LT.CHUTE(I)) GO TO 5060
    GO TO 5080
    J4=I
    IT9=1
    RDT(I)=0.00
CONTINUE
RDT(J4)=1.00
GO TO 5570
IT7=0
IT20=0
DO 5230 I=1,N
    IF(IT7.EQ.1) GO TO 5230
    IF(SS(I+1,J1).NE.0.00) GO TO 5230
    J4=I
    DO 5210 J=1,N
        RDT(J)=0.00
    CONTINUE
    RDT(J4)=1.00
    IT7=1
CONTINUE
IF(IT7.EQ.0) GO TO 5360
GO TO 5570
IV= -1.037
DO 5470 J=1,M
    SCMA=0.00
    DO 5420 I=1,N
        SCMA=SS(I+1,J1)*A(I,J)+SCMA
    CONTINUE
    IF(SCMA.GT.IV) GO TO 5450
    GO TO 5470
    JMAX=J
    IV=SCMA
CONTINUE
IF(IV.GT.0.00) GO TO 5530

NEVALF=NEVALF+1
DO 5486 I=1,N
    PRECC(I)=SS(I+1,J1)
CONTINUE
CALL DEMAND(DEMAN,PRECC,W0,PARUTI,ID,N,K1,B)
DO 5520 I=1,N
    RDT(I)=DEMAN(I)
CONTINUE
GO TO 5570

NEVALQ=NEVALQ+1
IT20=1
DO 5550 I=1,N
    RDT(I)= -A(I,JMAX)
CONTINUE
RETURN
```




END

NICAMP

```
*****
*****
```

```
      SUBROUTINE PIVOTA(BINV,ROT,ROT1,EPS,W,BASAD,JCOLA,JCOLSS,J1,
1      PIV1,w1,PIV0,J2,J3,JMAX,IT20,N,J4,NPIVOT)
```

```
*****
*****
```

```
      REAL*8 BINV(20,20),ROT(20),ROT1(20),EPS(20),W(20),PIV1(20)
1      ,w1(20),PIV0(20,20),X0,X1,SMIN,TOL,DELTA,ROTM,PHI,PHI1
      INTEGER BASAD(20),JCOLA(20),JCOLSS(20)
```

```
      IF (J4.NE.0) THEN
        DO 6050 I=1,N
          ROT(I)=BINV(I,J4)
        ELSE
          NPIVOT=NPIVOT+1
          DO 6080 I=1,N
            X0=0.00
            DO 6060 J=1,N
              X0=X0+BINV(I,J)*ROT(J)
            CONTINUE
            ROT1(I)=X0
          CONTINUE
          DO 6086 I=1,N
            ROT(I)=ROT1(I)
          CONTINUE
        END IF
```

```
      TOL =1.0-8
      DELTA = 5.0-4
      PHI = 1.037
      DO I=1,N
        IF(ROT(I).GE.TOL) THEN
          PHI1 = (W(I)+DELTA)/ROT(I)
          IF(PHI1.LT.PHI) THEN
            PHI = PHI1
          END IF
        END IF
```

```
      END DO
      ROTM= -1.037
      DO I=1,N
        IF((ROT(I).GE.TOL).AND.(W(I)/ROT(I).LE.PHI)) THEN
          IF(ROTM.GE.ROT(I)) THEN
            ROTM=ROT(I)
            J3=I
          END IF
        END IF
```

```
      END DO
```

```
      BASAD(J3)=0
      JCOLA(J3)=0
      IF(IT20.EQ.0) GO TO 6330
      BASAD(J3)=1
```



```
JCOLA(J3)=JMAX  
J2=JCOLSS(J3)  
JCOLSS(J3)=J1
```

```
DO 6410 I=1,N  
PIV1(I)=-RDT(I)/RDT(J3)
```

```
PIV1(J3)=1.00/RDT(J3)
```

```
DO 6430 I=1,N  
W1(I)=W(I)+PIV1(I)*W(J3)  
W1(J3)=PIV1(J3)*W(J3)  
DO 6433 I=1,N  
W(I)=W1(I)  
DO 6510 J=1,N  
DO 6500 I=1,N  
PIV0(I,J)=BINV(I,J)+PIV1(I)*BINV(J3,J)  
CONTINUE  
PIV0(J3,J)=PIV1(J3)*BINV(J3,J)  
CONTINUE  
DO 6560 I=1,N  
DO 6550 J=1,N  
BINV(I,J)=PIV0(I,J)  
CONTINUE  
CONTINUE  
RETURN  
END
```

```
*****  
*****  
SUBROUTINE DEMAND(DEMAN,PRECC,W0,PARUTI,ID,N,K1,B)
```

```
*****  
*****  
REAL*8 DEMAN(20),PRECC(20),W0(20,20),PARUTI(20,20),S(20),S1,  
1 S2,S20,S30,S40,IC  
DO 8130 J=1,N  
DEMAN(J)=0.00  
DO 8120 I=1,K1  
S2=0.00  
S1=0.00  
DO 8080 K=1,N  
S1=S1+PRECC(K)*W0(I,K)/ID  
S30=(PRECC(K)/ID)**(1.00-B(I))  
S40=S30*PARUTI(I,K)  
S2=S2+S40  
CONTINUE  
S1=PARUTI(I,J)*S1  
S2=(PRECC(J)/ID)**B(I)*S2  
DEMAN(J)=DEMAN(J)+S1/S2  
CONTINUE  
CONTINUE  
RETURN  
END
```

REFERÊNCIAS

- BROADIE, M.N., "An Introduction to the Octahedral Algorithm", Presented at the Workshop on the Application and Solution of Economic Equilibrium Models, [June, 1984].
- BROUWER, L.E., "Über abbildungen von Mannigfaltigkeiten", Math. Ann., 71 [1912], 97-115.
- DEBREU, G., "Continuity Properties of Paretian Utility", International Economic Review, 5, [1964], 285-293.
- EAVES, B.C., "Computing Kakutani Fixed Points", SIAM J. of App. Math., 21,2 [1971], 236-244.
- EAVES, B.C., "Homotopies for Computation of Fixed Points", Math. Programming, 3 [1972], 1-22.
- GARCIA, C.B. e ZANGWILL, W.I., "Pathways to Solutions, Fixed Points and Equilibria", Prentice-Hall, Series in Computational Mathematics, Cleve Moler, Advisor, [1981].
- HANSEN, T. e SCARF, H.E., "Computation of Economic Equilibria", Yale University Press, New Haven [1973].

MERRILL, O.H., "Applications and Extensions of an Algorithm that Computes Fixed Points of Certain Upper Semi-Continuous Point to Set Mappings", Ph.D. Dissertation, Department of Industrial Engineering, University of Michigan [1972].

SCARF, H.E., "The Approximation of Fixed Points of Continuous Mappings", SIAM J. of App. Math., 15 [1967], 1328-1343.

SCARF, H.E., "The Computation of Equilibrium Prices", em H. Scarf e J. Showen eds., Applied General Equilibrium Analysis, Cambridge University Press, Cambridge [1984].

SAIGAL, R., "On the Convergence Rate of Algorithms for Solving Equations that are Based on Methods of Complementary Pivoting", Math. of O.R., 2 [1977], 108-124.

SAIGAL, R. e TODD, M.J., "Efficient Acceleration Techniques for Fixed Point Algorithms", SIAM J. of Numerical Analysis, 15 [1978], 997-1007.

SCARAMUCCI, J. e ARICA, J., "Métodos Homotópicos para a Determinação de Equilíbrios Econômicos". Anais do IX Congresso Nacional de Matemática Aplicada e Computacional (SBMAC), [1986], 373-381, Brasília.

- TODD, M.J., "On Triangulations for Computing Fixed Points". Mathematical Programming, 10 [1976], 322-346.
- TODD, M.J., "The Computation of Fixed Points and Applications" , Lecture Notes in Economics and Mathematical Systems, 124 , Springer-Verlag, Berlin .Heidelberg, N.Y. [1976].
- TODD, M.J., "Efficient Methods of Computing Economic Equilibria" , em H. Scarf e J. Showen eds., Applied General Equilibrium Analysis, Cambridge University Press, Cambridge [1984].
- VARIAN, H.R., "Microeconomic Analysis", W.W. Norton & Company, New York, London [1978].
- AWONIYI, S.A. e TODD, M.J., "An Efficient Simplicial Algorithm for Computing a Zero of a Convex Union of Smooth Functions". Mathematical Programming, 25 [1983], 83-108.
-