

# Computação de Equilíbrios Competitivos

## Método de Scarf

Ian Teixeira Barreiro

Laboratório em Economia, Matemática e Computação (LEMC FEA-RP/USP)  
Universidade de São Paulo

2 de Setembro, 2022

# Tópicos Abordados

- 1 Equilíbrio Geral
- 2 Teorema de Brower
- 3 Método de Scarf
- 4 Exemplo Prático

# Equilíbrio Geral

# Teoria

- Equilíbrio Parcial: fazemos considerações sobre o preço e quantidade de equilíbrio de um mercado como se ele fosse isolado dos demais.
- Equilíbrio Geral: consideramos as interrelações entre vários mercados para determinar o preço e quantidade de equilíbrio em todos eles simultaneamente.
  - Exemplos de interações: bens substitutos, complementares ou insumos

# Teoria

- Equilíbrio Parcial: fazemos considerações sobre o preço e quantidade de equilíbrio de um mercado como se ele fosse isolado dos demais.
- Equilíbrio Geral: consideramos as interrelações entre vários mercados para determinar o preço e quantidade de equilíbrio em todos eles simultaneamente.
  - Exemplos de interações: bens substitutos, complementares ou insumos

# Teoria

- Equilíbrio Geral de Trocas: não há produção na economia. Os agentes trocam suas dotações de modo a aumentar suas utilidades.
- Equilíbrio Geral com Produção: há produção. Os consumidores devem decidir quanto ofertar dos fatores de produção e quanto consumir de modo a maximizar sua utilidade. As firmas devem decidir quanto demandar de fatores de produção e quanto produzir para maximizar seu lucro.

# Teoria

- Equilíbrio Geral de Trocas: não há produção na economia. Os agentes trocam suas dotações de modo a aumentar suas utilidades.
- Equilíbrio Geral com Produção: há produção. Os consumidores devem decidir quanto ofertar dos fatores de produção e quanto consumir de modo a maximizar sua utilidade. As firmas devem decidir quanto demandar de fatores de produção e quanto produzir para maximizar seu lucro.

# Teoria

Problema: determinar um vetor de preços relativos tal que não haja excesso de demanda em nenhum mercado. A esse vetor de preços atingiremos um ótimo de Pareto: não há como melhorar a condição de nenhum agente sem piorar a de outro.



# Formulando Matematicamente o Problema

- Há  $n$  bens:

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n), \quad x_i \geq 0, \quad i = \{1, \dots, n\}$$

- Há  $k$  agentes:

- Utilidades:

$$u(\mathbf{x}) = u^j(x_1, \dots, x_n), \quad j = \{1, \dots, k\}$$

- Dotações :

$$\mathbf{w}^j = (w_1^j, w_2^j, \dots, w_n^j), \quad j = \{1, \dots, k\}$$

- Vetor de preços relativos:

$$\mathbf{p} = (p_1, \dots, p_n)$$

# Formulando Matematicamente o Problema

- Há  $n$  bens:

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n), \quad x_i \geq 0, \quad i = \{1, \dots, n\}$$

- Há  $k$  agentes:

- Utilidades:

$$u(\mathbf{x}) = u^j(x_1, \dots, x_n), \quad j = \{1, \dots, k\}$$

- Dotações :

$$\mathbf{w}^j = (w_1^j, w_2^j, \dots, w_n^j), \quad j = \{1, \dots, k\}$$

- Vetor de preços relativos:

$$\mathbf{p} = (p_1, \dots, p_n)$$

# Formulando Matematicamente o Problema

- Há  $n$  bens:

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n), \quad x_i \geq 0, \quad i = \{1, \dots, n\}$$

- Há  $k$  agentes:

- Utilidades:

$$u(\mathbf{x}) = u^j(x_1, \dots, x_n), \quad j = \{1, \dots, k\}$$

- Dotações :

$$\mathbf{w}^j = (w_1^j, w_2^j, \dots, w_n^j), \quad j = \{1, \dots, k\}$$

- Vetor de preços relativos:

$$\mathbf{p} = (p_1, \dots, p_n)$$

# Formulando Matematicamente o Problema

$$\begin{aligned} \max_{\mathbf{x}} \quad & u^j(\mathbf{x}) \\ \text{s.t} \quad & \\ & \sum_{i=1}^n p_i x_i^j = \sum_{i=1}^n p_i w_i^j \end{aligned}$$

# Formulando Matematicamente o Problema

- Resolvendo o problema de maximização encontramos um vetor de demandas como função apenas dos preços relativos.

$$\mathbf{d}^j = (d_1^j(\mathbf{p}), \dots, d_n^j(\mathbf{p})), j = \{1, \dots, k\}$$

- As funções demanda são homogêneas de grau 0 (importa apenas o preço relativo - não o valor de face dos preços). Isso permite que normalizemos todos os preços de modo que o vetor de preços sempre some 1.
  - Isso terá uma importância grande para a interpretação geométrica do problema que veremos posteriormente (todos os vetores de preços estarão no simplex n-dimensional).
- As demandas de mercado são obtidas somando as demandas de todos os agentes por cada bem.

# Formulando Matematicamente o Problema

- Resolvendo o problema de maximização encontramos um vetor de demandas como função apenas dos preços relativos.

$$\mathbf{d}^j = (d_1^j(\mathbf{p}), \dots, d_n^j(\mathbf{p})), \quad j = \{1, \dots, k\}$$

- As funções demanda são homogêneas de grau 0 (importa apenas o preço relativo - não o valor de face dos preços). Isso permite que normalizemos todos os preços de modo que o vetor de preços sempre some 1.
  - Isso terá uma importância grande para a interpretação geométrica do problema que veremos posteriormente (todos os vetores de preços estarão no simplex n-dimensional).

- As demandas de mercado são obtidas somando as demandas de todos os agentes por cada bem.

# Formulando Matematicamente o Problema

- Resolvendo o problema de maximização encontramos um vetor de demandas como função apenas dos preços relativos.

$$\mathbf{d}^j = (d_1^j(\mathbf{p}), \dots, d_n^j(\mathbf{p})), \quad j = \{1, \dots, k\}$$

- As funções demanda são homogêneas de grau 0 (importa apenas o preço relativo - não o valor de face dos preços). Isso permite que normalizemos todos os preços de modo que o vetor de preços sempre some 1.
  - Isso terá uma importância grande para a interpretação geométrica do problema que veremos posteriormente (todos os vetores de preços estarão no simplex n-dimensional).
- As demandas de mercado são obtidas somando as demandas de todos os agentes por cada bem.

# Formulando Matematicamente o Problema

- Dado que não há produção, para todo vetor de preços a seguinte condição acerca das demandas de mercado e dotação total de bens deve ser satisfeita.
  - Lei de Walras:

$$\sum_{i=1}^n p_i d_i = \sum_{i=1}^n p_i w_i$$
$$\sum_{i=1}^n p_i (d_i - w_i) = 0$$



# Formulando Matematicamente o Problema

- Definamos o excesso de demanda de mercado como:

$$g_i(\mathbf{p}) = d_i(\mathbf{p}) - w_i$$

- Em equilíbrio teremos:

$$g_1(\mathbf{p}) \leq 0$$

$$\vdots$$

$$g_n(\mathbf{p}) \leq 0$$

# Formulando Matematicamente o Problema

- Definamos o excesso de demanda de mercado como:

$$g_i(\mathbf{p}) = d_i(\mathbf{p}) - w_i$$

- Em equilíbrio teremos:

$$g_1(\mathbf{p}) \leq 0$$

$$\vdots$$

$$g_n(\mathbf{p}) \leq 0$$

# O Problema de Existência de um Equilíbrio Competitivo

- Descrevemos o problema e o que entendemos como uma solução.
- O problema de maximização com restrições não é simples. Pode envolver sistemas de equações e inequações não-lineares bastante complexos. **Não é óbvio que existe uma solução.**
- Esse problema foi de grande relevância histórica para a economia matemática.
- Avanços em topologia algébrica permitiram a solução do problema de existência de um equilíbrio competitivo.
  - Teorema de Brouwer

# O Problema de Existência de um Equilíbrio Competitivo

- Descrevemos o problema e o que entendemos como uma solução.
- O problema de maximização com restrições não é simples. Pode envolver sistemas de equações e inequações não-lineares bastante complexos. **Não é óbvio que existe uma solução.**
- Esse problema foi de grande relevância histórica para a economia matemática.
- Avanços em topologia algébrica permitiram a solução do problema de existência de um equilíbrio competitivo.
  - Teorema de Brouwer

# O Problema de Existência de um Equilíbrio Competitivo

- Descrevemos o problema e o que entendemos como uma solução.
- O problema de maximização com restrições não é simples. Pode envolver sistemas de equações e inequações não-lineares bastante complexos. **Não é óbvio que existe uma solução.**
- Esse problema foi de grande relevância histórica para a economia matemática.
- Avanços em topologia algébrica permitiram a solução do problema de existência de um equilíbrio competitivo.
  - Teorema de Brouwer

# O Problema de Existência de um Equilíbrio Competitivo

- Descrevemos o problema e o que entendemos como uma solução.
- O problema de maximização com restrições não é simples. Pode envolver sistemas de equações e inequações não-lineares bastante complexos. **Não é óbvio que existe uma solução.**
- Esse problema foi de grande relevância histórica para a economia matemática.
- Avanços em topologia algébrica permitiram a solução do problema de existência de um equilíbrio competitivo.
  - Teorema de Brouwer

# Teorema de Brower

# Mapeamento

- Generalização da ideia de função para domínios e contradomínios que podem ser conjuntos diversos.
  - Exemplo: a posição de um ponto em uma esfera em que o mapeamento são rotações entorno do eixo
- Nos preocuparemos com os mapeamentos contínuos (pequenas variações nas variáveis independentes geram pequenas variações na variável dependente), em especial os mapeamentos contínuos de um conjunto nele mesmo.
- Mais do que isso nos preocuparemos com os mapeamentos contínuos do simplex nele mesmo.



# Mapeamento

- Generalização da ideia de função para domínios e contradomínios que podem ser conjuntos diversos.
  - Exemplo: a posição de um ponto em uma esfera em que o mapeamento são rotações entorno do eixo
- Nos preocuparemos com os mapeamentos contínuos (pequenas variações nas variáveis independentes geram pequenas variações na variável dependente), em especial os mapeamentos contínuos de um conjunto nele mesmo.
- Mais do que isso nos preocuparemos com os mapeamentos contínuos do simplex nele mesmo.

# Mapeamento

- Generalização da ideia de função para domínios e contradomínios que podem ser conjuntos diversos.
  - Exemplo: a posição de um ponto em uma esfera em que o mapeamento são rotações entorno do eixo
- Nos preocuparemos com os mapeamentos contínuos (pequenas variações nas variáveis independentes geram pequenas variações na variável dependente), em especial os mapeamentos contínuos de um conjunto nele mesmo.
- Mais do que isso nos preocuparemos com os mapeamentos contínuos do simplex nele mesmo.

# Simplex

- Lembrem-se de que falamos que já que as demandas são homogêneas de grau 0 podemos normalizar o vetor preço para que ele sempre some 1.
- Um simplex  $n$ -dimensional é o conjunto de vetores de  $n$  componentes que somam 1.  $x = (x_1, x_2, \dots, x_n)$ ,  $\sum_{i=1}^n x_i = 1$ .
  - É a generalização da ideia de triângulo para dimensões mais altas.

# Simplex

- Lembrem-se de que falamos que já que as demandas são homogêneas de grau 0 podemos normalizar o vetor preço para que ele sempre some 1.
- Um simplex  $n$ -dimensional é o conjunto de vetores de  $n$  componentes que somam 1.  $x = (x_1, x_2, \dots, x_n)$ ,  $\sum_{i=1}^n x_i = 1$ .
  - É a generalização da ideia de triângulo para dimensões mais altas.

# Simplex

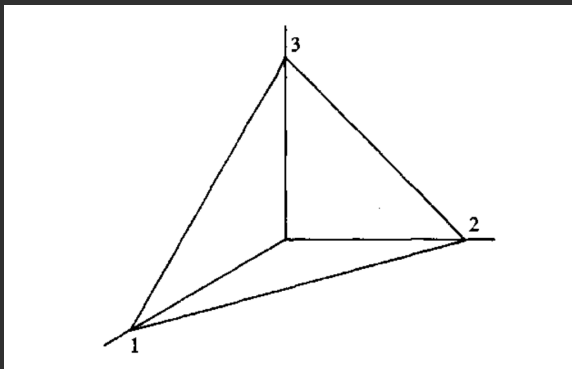


Figure: Um simplex tridimensional

Fonte: (Sarf and Hansen, 1973, p. 19)

# O Simplex e o Equilíbrio Geral

- Temos  $n$  bens, cujos preços relativos são não negativos e somam 1 (estão no simplex).
- Queremos um mapeamento contínuo que nos aproxime do equilíbrio. Uma possibilidade é aumentar o preço relativo dos bens com excesso de demanda e diminuir o preço relativo dos bens com excess de oferta.
- Podemos garantir que o contradomínio do mapeamento esteja no simplex garantindo que as imagens sejam não negativas e somem 1.

■

$$y_i = \frac{x_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)} \quad (1)$$

# O Simplex e o Equilíbrio Geral

- Temos  $n$  bens, cujos preços relativos são não negativos e somam 1 (estão no simplex).
- Queremos um mapeamento contínuo que nos aproxime do equilíbrio. Uma possibilidade é aumentar o preço relativo dos bens com excesso de demanda e diminuir o preço relativo dos bens com excess de oferta.
- Podemos garantir que o contradomínio do mapeamento esteja no simplex garantindo que as imagens sejam não negativas e somem 1.

$$y_i = \frac{x_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)} \quad (1)$$

# O Simplex e o Equilíbrio Geral

- Temos  $n$  bens, cujos preços relativos são não negativos e somam 1 (estão no simplex).
- Queremos um mapeamento contínuo que nos aproxime do equilíbrio. Uma possibilidade é aumentar o preço relativo dos bens com excesso de demanda e diminuir o preço relativo dos bens com excess de oferta.
- Podemos garantir que o contradomínio do mapeamento esteja no simplex garantindo que as imagens sejam não negativas e somem 1.



$$y_i = \frac{x_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)} \quad (1)$$



# Teorema de Brouwer

**Teorema de Brouwer:** Dado um simplex  $n$  dimensional e um mapeamento contínuo  $y = f(x)$  do simplex nele mesmo, haverá um ponto fixo nesse simplex, ou seja, haverá um ponto tal que  $\hat{x} = f(\hat{x})$ .

# Teorema de Brouwer

## ■ Consequência:

$$\hat{x}_i = \frac{\hat{x}_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)}, \forall i \in \{1, \dots, n\}$$

- Isso é possível apenas se todo  $g_i(\mathbf{x}) \leq 0$ , ou seja, se estivermos no equilíbrio competitivo.
- Demonstra-se a existência de equilíbrios competitivos.

# Teorema de Brouwer

- Consequência:

$$\hat{x}_i = \frac{\hat{x}_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)}, \quad \forall i \in \{1, \dots, n\}$$

- Isso é possível apenas se todo  $g_i(\mathbf{x}) \leq 0$ , ou seja, se estivermos no equilíbrio competitivo.
- Demonstra-se a existência de equilíbrios competitivos.

# Teorema de Brouwer

- Consequência:

$$\hat{x}_i = \frac{\hat{x}_i + \max(g_i(\mathbf{x}), 0)}{1 + \sum_{i=1}^n \max(g_i(\mathbf{x}), 0)}, \quad \forall i \in \{1, \dots, n\}$$

- Isso é possível apenas se todo  $g_i(\mathbf{x}) \leq 0$ , ou seja, se estivermos no equilíbrio competitivo.
- Demonstra-se a existência de equilíbrios competitivos.

# Teorema de Brouwer

- Demonstramos a existência de equilíbrios competitivos, mas como os encontramos?
- O procedimento de aumentar o preço relativo quando há excesso de demanda e diminuir o preço caso contrário pode não convergir.
- Surge o método de Scarf para aproximar pontos fixos computacionalmente.

# Método de Scarf

# Intuição

- Vamos dividir o simplex em vários simplex menores (subsimplex) e encontrar aquele simplex menor que possuir em seu interior um ponto fixo.
- A aproximação é tão melhor quanto menores forem os subsimplex.

# Intuição

- Vamos dividir o simplex em vários simplex menores (subsimplex) e encontrar aquele simplex menor que possuir em seu interior um ponto fixo.
- A aproximação é tão melhor quanto menores forem os subsimplex.



# Intuição

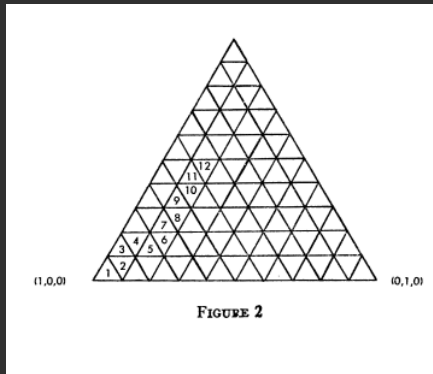


Figure: Exemplo gráfico da aproximação de pontos fixos

Fonte: (Scarf, 1969, p. 672)

# Primitivos

- Os subsimplex são chamados de primitivos e podem ser representados por uma matriz  $n \times n$  onde cada vetor coluna é um ponto no simplex pelo qual um dos lados do primitivo passa.

$$\begin{bmatrix} 19 & 18 & 17 & \underline{0.7} \\ \underline{0} & 18 & 17 & \mathbf{0.1} \\ 19 & \underline{0} & 17 & 0.1 \\ 19 & 18 & \underline{0} & 0.1 \end{bmatrix}$$

# Labels

- Daremos a cada vetor dos primitivos participantes do nosso algoritmo um, "índice" que o identifique.
- Para os vetores interiores ao simplex podemos atribuir seus índices aplicando  $f(x) - x$  em que  $f$  é o mapeamento 1 e buscando o primeiro componente positivo. Para os vetores pertencentes a lados do simplex o índice será o componente constante do lado a que pertencem.
  - Ou seja, os labels serão baseados no primeiro mercado com excesso de demanda do vetor.
  - Na prática basta que atribuamos labels com base em  $\arg \max \frac{g(x)}{x}$ .
- A condição de parada do nosso algoritmo será quando encontrarmos um primitivo onde cada vetor correspondente ao seus lados tenha um label diferente.

# Labels

- Daremos a cada vetor dos primitivos participantes do nosso algoritmo um, "índice" que o identifique.
- Para os vetores interiores ao simplex podemos atribuir seus índices aplicando  $f(x) - x$  em que  $f$  é o mapeamento 1 e buscando o primeiro componente positivo. Para os vetores pertencentes a lados do simplex o índice será o componente constante do lado a que pertencem.
  - Ou seja, os labels serão baseados no primeiro mercado com excesso de demanda do vetor.
  - Na prática basta que atribuamos labels com base em  $\arg \max \frac{g(x)}{x}$ .
- A condição de parada do nosso algoritmo será quando encontrarmos um primitivo onde cada vetor correspondente ao seus lados tenha um label diferente.

# Labels

- Daremos a cada vetor dos primitivos participantes do nosso algoritmo um, "índice" que o identifique.
- Para os vetores interiores ao simplex podemos atribuir seus índices aplicando  $f(x) - x$  em que  $f$  é o mapeamento 1 e buscando o primeiro componente positivo. Para os vetores pertencentes a lados do simplex o índice será o componente constante do lado a que pertencem.
  - Ou seja, os labels serão baseados no primeiro mercado com excesso de demanda do vetor.
  - Na prática basta que atribuamos labels com base em  $\arg \max \frac{g(x)}{x}$ .
- A condição de parada do nosso algoritmo será quando encontrarmos um primitivo onde cada vetor correspondente ao seus lados tenha um label diferente.

# Operação do Algoritmo

- 1 Criamos uma lista de  $n$  vetores nos lados do simplex e  $k$  vetores no interior do simplex.
- 2 Na primeira iteração do nosso algoritmo, estaremos em um subsimplex em um dos cantos do simplex maior, que seja composto por  $n - 1$  vetores nos lados do simplex e 1 vetor interior ao simplex.
- 3 Haverá um lado nesse subsimplex com um label igual ao do vetor que acabou de entrar no subsimplex (o vetor interior). Removemos esse vetor do subsimplex.
- 4 Movemos nosso subsimplex paralelamente ao lado que passava pelo vetor que acabamos de remover, até que atinjamos um novo vetor da lista. Com esse vetor formaremos um novo subsimplex.
- 5 Observamos o label do vetor que acabou de entrar no novo subsimplex. Se a todos os labels dos vetores no subsimplex forem diferentes, atingimos a condição de parada. Caso contrário, voltamos ao passo 3.

# Operação do Algoritmo

- 1 Criamos uma lista de  $n$  vetores nos lados do simplex e  $k$  vetores no interior do simplex.
- 2 Na primeira iteração do nosso algoritmo, estaremos em um subsimplex em um dos cantos do simplex maior, que seja composto por  $n - 1$  vetores nos lados do simplex e 1 vetor interior ao simplex.
- 3 Haverá um lado nesse subsimplex com um label igual ao do vetor que acabou de entrar no subsimplex (o vetor interior). Removemos esse vetor do subsimplex.
- 4 Movemos nosso subsimplex paralelamente ao lado que passava pelo vetor que acabamos de remover, até que atinjamos um novo vetor da lista. Com esse vetor formaremos um novo subsimplex.
- 5 Observamos o label do vetor que acabou de entrar no novo subsimplex. Se a todos os labels dos vetores no subsimplex forem diferentes, atingimos a condição de parada. Caso contrário, voltamos ao passo 3.

# Operação do Algoritmo

- 1 Criamos uma lista de  $n$  vetores nos lados do simplex e  $k$  vetores no interior do simplex.
- 2 Na primeira iteração do nosso algoritmo, estaremos em um subsimplex em um dos cantos do simplex maior, que seja composto por  $n - 1$  vetores nos lados do simplex e 1 vetor interior ao simplex.
- 3 Haverá um lado nesse subsimplex com um label igual ao do vetor que acabou de entrar no subsimplex (o vetor interior). Removemos esse vetor do subsimplex.
- 4 Movemos nosso subsimplex paralelamente ao lado que passava pelo vetor que acabamos de remover, até que atinjamos um novo vetor da lista. Com esse vetor formaremos um novo subsimplex.
- 5 Observamos o label do vetor que acabou de entrar no novo subsimplex. Se a todos os labels dos vetores no subsimplex forem diferentes, atingimos a condição de parada. Caso contrário, voltamos ao passo 3.



# Operação do Algoritmo

- 1 Criamos uma lista de  $n$  vetores nos lados do simplex e  $k$  vetores no interior do simplex.
- 2 Na primeira iteração do nosso algoritmo, estaremos em um subsimplex em um dos cantos do simplex maior, que seja composto por  $n - 1$  vetores nos lados do simplex e 1 vetor interior ao simplex.
- 3 Haverá um lado nesse subsimplex com um label igual ao do vetor que acabou de entrar no subsimplex (o vetor interior). Removemos esse vetor do subsimplex.
- 4 Movemos nosso subsimplex paralelamente ao lado que passava pelo vetor que acabamos de remover, até que atinjamos um novo vetor da lista. Com esse vetor formaremos um novo subsimplex.
- 5 Observamos o label do vetor que acabou de entrar no novo subsimplex. Se a todos os labels dos vetores no subsimplex forem diferentes, atingimos a condição de parada. Caso contrário, voltamos ao passo 3.

# Operação do Algoritmo

- 1 Criamos uma lista de  $n$  vetores nos lados do simplex e  $k$  vetores no interior do simplex.
- 2 Na primeira iteração do nosso algoritmo, estaremos em um subsimplex em um dos cantos do simplex maior, que seja composto por  $n - 1$  vetores nos lados do simplex e 1 vetor interior ao simplex.
- 3 Haverá um lado nesse subsimplex com um label igual ao do vetor que acabou de entrar no subsimplex (o vetor interior). Removemos esse vetor do subsimplex.
- 4 Movemos nosso subsimplex paralelamente ao lado que passava pelo vetor que acabamos de remover, até que atinjamos um novo vetor da lista. Com esse vetor formaremos um novo subsimplex.
- 5 Observamos o label do vetor que acabou de entrar no novo subsimplex. Se a todos os labels dos vetores no subsimplex forem diferentes, atingimos a condição de parada. Caso contrário, voltamos ao passo 3.

# Intuição da Prova

Por que o algoritmo consegue aproximar pontos fixos?

- Daremos labels aos vetores de acordo com o componente com maior excesso de demanda.
- Alcançada a condição de parada teremos um conjunto primitivo em que cada vetor tem excesso de demanda em um mercado diferente.
- Imaginemos que formamos uma lista cada vez mais densa de vetores no simplex, até que no limite tenhamos todos os pontos no simplex. O primitivo final a que chegaremos será um ponto (simplex 0-dimensional) tal que todos os seus componentes tenham  $f(x_i) - x_i \geq 0$ ,  $i = \{1, \dots, n\}$ .

# Intuição da Prova

Por que o algoritmo consegue aproximar pontos fixos?

- Daremos labels aos vetores de acordo com o componente com maior excesso de demanda.
- Alcançada a condição de parada teremos um conjunto primitivo em que cada vetor tem excesso de demanda em um mercado diferente.
- Imaginemos que formamos uma lista cada vez mais densa de vetores no simplex, até que no limite tenhamos todos os pontos no simplex. O primitivo final a que chegaremos será um ponto (simplex 0-dimensional) tal que todos os seus componentes tenham  $f(x_i) - x_i \geq 0$ ,  $i = \{1, \dots, n\}$ .

# Intuição da Prova

Por que o algoritmo consegue aproximar pontos fixos?

- Daremos labels aos vetores de acordo com o componente com maior excesso de demanda.
- Alcançada a condição de parada teremos um conjunto primitivo em que cada vetor tem excesso de demanda em um mercado diferente.
- Imaginemos que formamos uma lista cada vez mais densa de vetores no simplex, até que no limite tenhamos todos os pontos no simplex. O primitivo final a que chegaremos será um ponto (simplex 0-dimensional) tal que todos os seus componentes tenham  $f(x_i) - x_i \geq 0$ ,  $i = \{1, \dots, n\}$ .

# Intuição da Prova

- Mas tanto  $f(x)$  quanto  $x$  estão no simplex. Logo,  $\sum_{i=1}^n f_i(x) = 1$  e  $\sum_{i=1}^n x_i = 1$ .
- Assim,  $\sum_{i=1}^n f_i(x) = \sum_{i=1}^n x_i \longrightarrow \sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$ . Mas lembramos que  $f(x_i) - x_i \geq 0$  de modo que o único jeito de  $\sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$  se verificar é se  $f(x_i) - x_i = 0 \longrightarrow f(x_i) = x_i, \forall i$ .
- Demonstra-se portanto que nesse ponto  $f(x^*) = x^*$ . Encontramos um ponto fixo.

# Intuição da Prova

- Mas tanto  $f(x)$  quanto  $x$  estão no simplex. Logo,  $\sum_{i=1}^n f_i(x) = 1$  e  $\sum_{i=1}^n x_i = 1$ .
- Assim,  $\sum_{i=1}^n f_i(x) = \sum_{i=1}^n x_i \longrightarrow \sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$ . Mas lembramos que  $f(x_i) - x_i \geq 0$  de modo que o único jeito de  $\sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$  se verificar é se  $f(x_i) - x_i = 0 \longrightarrow f(x_i) = x_i, \forall i$ .
- Demonstra-se portanto que nesse ponto  $f(x^*) = x^*$ . Encontramos um ponto fixo.

# Intuição da Prova

- Mas tanto  $f(x)$  quanto  $x$  estão no simplex. Logo,  $\sum_{i=1}^n f_i(x) = 1$  e  $\sum_{i=1}^n x_i = 1$ .
- Assim,  $\sum_{i=1}^n f_i(x) = \sum_{i=1}^n x_i \longrightarrow \sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$ . Mas lembramos que  $f(x_i) - x_i \geq 0$  de modo que o único jeito de  $\sum_{i=1}^n f_i(x) - \sum_{i=1}^n x_i = 0$  se verificar é se  $f(x_i) - x_i = 0 \longrightarrow f(x_i) = x_i, \forall i$ .
- Demonstra-se portanto que nesse ponto  $f(x^*) = x^*$ . Encontramos um ponto fixo.



# Exemplo Prático

# Exemplo

- 5 agentes
- 10 bens

# Dotações

<i>Consumer</i>	<i>Initial stock of commodities</i>									
	1	2	3	4	5	6	7	8	9	10
1	0.6	0.2	0.2	20.0	0.1	2.0	9.0	5.0	5.0	15.0
2	0.2	11.0	12.0	13.0	14.0	15.0	16.0	5.0	5.0	9.0
3	0.4	9.0	8.0	7.0	6.0	5.0	4.0	5.0	7.0	12.0
4	1.0	5.0	5.0	5.0	5.0	5.0	5.0	8.0	3.0	17.0
5	8.0	1.0	22.0	10.0	0.3	0.9	5.1	0.1	6.2	11.0

Figure: Dotações dos agentes

Fonte: (Scarf and Hansen, 1973, p. 66)

# Preferências pelos Bens

Consumer	Utility parameters $a$									
	1	2	3	4	5	6	7	8	9	10
1	1.0	1.0	3.0	0.1	0.1	1.2	2.0	1.0	1.0	0.7
2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
3	9.9	0.1	5.0	0.2	6.0	0.2	8.0	1.0	1.0	0.2
4	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
5	1.0	13.0	11.0	9.0	4.0	0.9	8.0	1.0	2.0	10.0

Figure: Preferências dos agentes

Fonte: (Scarf and Hansen, 1973, p. 66)

# Substitutibilidade entre os Bens

*Utility parameters b*  
*Consumer*

1	2.0
2	1.3
3	3.0
4	0.2
5	0.6

**Figure:** Substitutibilidade entre os bens dos agentes

Fonte: (Scarf and Hansen, 1973, p. 66)

# Utilidades

a = preferências | b = substitutibilidade | x = quantidade de bens

$$u(x) = \sum_{i=1}^{10} a_i^{\frac{1}{b}} x_i^{1-\frac{1}{b}} \quad (2)$$

# Excessos de Demanda de Mercado

$$g(x_i) = \sum_{j=1}^5 \frac{a_{ij} \sum_{i=1}^{10} p_i w_{ij}}{p_i^b \sum_{i=1}^{10} p_i^{1-b} a_{ij}} - w_i \quad (3)$$

# Convergência

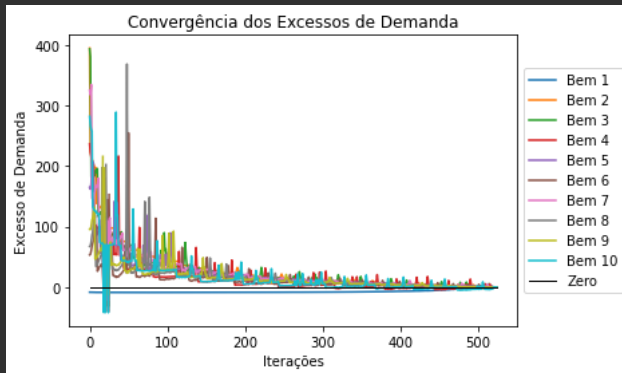


Figure: Convergência dos excessos de demanda de mercado



# Preços Relativos de Equilíbrio

$$p = (0.18717914, 0.10933056, 0.09885215, 0.04317213, 0.11681448, \\ 0.07693998, 0.11691355, 0.1023353, 0.09864703, 0.04937033)$$

# References

1. Scarf, H. An Example of an Algorithm for Calculating General Equilibrium Prices. *The American Economic Review* **59**, 669–677. ISSN: 00028282. <http://www.jstor.org/stable/1813239> (2022) (1969).
2. Scarf, H. E. & Hansen, T. *The computation of economic equilibria, by Herbert Scarf with the collaboration of Terje Hansen*. English, x, 249 p. ISBN: 0300016409 (Yale University Press New Haven, 1973).

# The End