

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

Tese de Doutorado

Métodos Quantitativos para Construção, Calibragem e Solução de
Modelos Computáveis de Equilíbrio Geral de Grande Porte

Orlando Frederico José Godoy Bordoni

Campinas, 21 de fevereiro de 2001

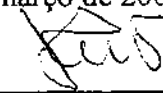
Pós-Graduação em Matemática Aplicada



Métodos Quantitativos para Construção, Calibragem e Solução de Modelos Computáveis de Equilíbrio Geral de Grande Porte

Este exemplar corresponde à redação final da tese devidamente corrigida e defendida por Orlando Frederico José Godoy Bordoni e aprovada pela comissão julgadora.

Campinas, 6 de março de 2001



Prof. Dr. José Antonio Scaramucci

Orientador

Banca examinadora:

- 1 José Antonio Scaramucci
- 2 Joaquim José Martins Guilhoto
- 3 Joaquim Bento de Souza Ferreira Filho
- 4 Petrônio Pulino
- 5 Francisco de Assis Magalhães Gomes Neto
- 6 Stephen Anthony de Castro

Tese apresentada ao **Instituto de Matemática, Estatística e Computação Científica**, Unicamp, como requisito parcial à obtenção do Título de DOUTOR em Matemática Aplicada

UNIDADE	30		
N.º CHAMADA:	F. 110210		
	B644m		
V.	Ex.		
TOMBO BC/	44437		
PROC.	16-892101		
C	<input type="checkbox"/>	D	<input checked="" type="checkbox"/>
PREC.	R\$ 11,00		
DATA	15/05/01		
N.º CPD			

CM-00155041-1

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

B644m Bordoni, Orlando Frederico José Godoy
Métodos quantitativos para construção, calibragem e solução de
modelos computáveis de equilíbrio geral de grande porte / Orlando Frederico
José Godoy Bordoni – Campinas, [S.P. :s.n.], 2001.

Orientador : José Antonio Scaramucci

Tese (doutorado) - Universidade Estadual de Campinas, Instituto de
Matemática, Estatística e Computação Científica.

1. Equilíbrio econômico – Modelos matemáticos. 2. Sistemas não-lineares.
I. Scaramucci, José Antonio. II. Universidade Estadual de Campinas. Instituto de
Matemática, Estatística e Computação Científica. III. Título.

Tese de Doutorado defendida em 21 de fevereiro de 2001

e aprovada pela Banca Examinadora composta pelos Profs. Drs.



Prof (a). Dr (a). JOSÉ ANTONIO SCARAMUCCI



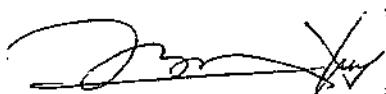
Prof (a). Dr (a). FRANCISCO DE ASSIS MAGALHÃES GOMES NETO



Prof (a). Dr (a). PETRÔNIO PULINO



Prof (a). Dr (a). JOAQUIM JOSÉ MARTINS GUILHOTO



Prof (a). Dr (a). JOAQUIM BENTO DE SOUZA FERREIRA FILHO



Prof. Dr. Stephen Anthony de Castro
(UnB)

À memória de meu pai,
Prof. Orlando Bordoni.

Com ele aprendi que uma coisa sempre *pode* ser feita de muitas maneiras –
mas que todas as coisas devem ser feitas de uma única maneira: a correta.

AGRADECIMENTOS:

a minha esposa, Berlina,

a meu filho, Orlando,

*por seu amor, apoio e compreensão incondicionais.
sem vocês eu não teria conseguido.*

a minha mãe, minha família,

por seu carinho, apoio e disposição em ajudar.

a meu orientador,

Prof. José Antonio Scaramucci:

*pela paciência em me mostrar o caminho,
por acreditar em meu trabalho.*

aos professores do Departamento de Matemática Aplicada,

Petrônio, Clóvis, Márcia, Martínez, Moretti:

*por disporem de seu tempo quando precisei de
auxílio.*

aos colegas de mestrado e amigos de sempre,

Esmê, Cris Vaz, Cris Wolf, Daniel, Arica, Gudêlia,

*em cada página deste trabalho tem um pouco de cada
um de vocês.*

ao pessoal da secretaria,

Fátima, Cidinha, Tânia e Ednaldo,

por sua atenção e disposição em ajudar.

a CEPLAC,

por seu apoio institucional.

ao Programa de Pós Graduação em Matemática Aplicada da

Unicamp,

*pelos auxílios financeiros prestados em recursos da
CAPES, do CNPQ e do FAEP/Unicamp.*

RESUMO

É crescente a demanda por análises quantitativas de questões econômicas globais em áreas como comércio internacional, desenvolvimento e crescimento econômico, agricultura e meio ambiente. A resolução de modelos computáveis de equilíbrio geral (CEG) para modelos de grande porte é a principal contribuição deste trabalho. Métodos quantitativos para a construção, calibragem e solução de modelos CEG de grande porte são apresentados na forma de rotinas computacionais destinadas ao pesquisador em economia. O algoritmo ACI (Atualização de uma Coluna por Iteração) – um método quase-newtoniano secante para resolução de sistemas não-lineares de grande porte (Gomes-Ruggiero [27], Martínez [41]) – é aqui acrescentado ao *Pegasus* – um sistema de arquitetura aberta de suporte computacional à análise aplicada de equilíbrio geral (Scaramucci [56], Bordoni [5]). Algumas características introduzidas nessa implementação, como o uso de alocação dinâmica de memória, o armazenamento das equações do modelo em forma simbólica, o cálculo do jacobiano do sistema não-linear por diferenças finitas e o armazenamento esparsa de matrizes viabilizam a resolução de modelos econômicos de grande porte de maneira eficiente e flexível. Outra contribuição apresentada aqui é um conjunto de procedimentos para construir modelos CEG de grande porte a partir de informações do banco de dados do GTAP (*Global Trade Analysis Project*) [34].

PALAVRAS CHAVE

Equilíbrio geral, modelos matemáticos, métodos quase-Newton, sistemas de equações não-lineares de grande porte, problemas de complementaridade não-linear, GTAP (*Global Trade Analysis Project*).

ABSTRACT

There has been a growing interest for quantitative analysis of global economic issues in areas such as international trade, economic development and growth, agriculture and environment. The main contribution of this work is to offer tools for solving large-scale computational general equilibrium (CGE) models. Quantitative methods for constructing, calibrating, and solving large-scale CGE models are presented as computational routines for the economic equilibrium practitioner. The one-column updating (ACI) algorithm – a quasi-Newton secant method for solving large-scale nonlinear systems (Gomes-Ruggiero [27], Martínez [41]) has been added to *Pegasus* – an open-ended computational system for applied general equilibrium (AGE) analysis (Scaramucci [56], Bordon[5]). Features introduced in this implementation include the use of dynamic memory allocation, symbolic storage of model equations, finite-difference evaluation of the system Jacobian, and sparse matrix storage make it feasible to solve large-scale economic models in a efficient and flexible way. Another contribution presented here is a collection of procedures for constructing large-scale CGE models using the GTAP database.

KEYWORDS

General equilibrium, mathematical models, quasi-Newton methods, large-scale systems of nonlinear equations, nonlinear complementarity problems, GTAP (*Global Trade Analysis Project*).

Sumário

1. Introdução	1
2. Revisão da Literatura	7
2.1 Aproximações lineares do modelo walrasiano	9
2.2 Modelos computáveis de equilíbrio geral	11
2.2.1 Os modelos originados por Johansen: ORANI, GTAP	11
2.2.2 Os modelos originados por Adelman e Robinson	13
2.2.3 Os modelos originados por Scarf e Mathiesen	15
2.3 Pacotes para modelagem e resolução de modelos CEG	17
2.3.1 O pacote GAMS	18
2.3.2 O pacote Gempack	19
2.3.3 O pacote <i>Pegasus</i>	20
3. Material e métodos	23
3.1 Equilíbrio econômico como um problema de complementaridade	24
3.1.1 A definição da economia	25
3.1.2 A oferta excedente e o preço de um bem	30
3.1.3 O lucro negativo e o nível de operação de uma atividade	32
3.1.4 A formalização do equilíbrio econômico	35
3.2 Extensões à teoria – Exemplos	38
3.2.1 Calculando a solução de equilíbrio	41
3.2.2 Calibrando o modelo com a técnica de geração de colunas	44
3.2.3 Fazendo o escalamento do modelo	48
3.2.4 Simulação de cenários alternativos	53
3.2.5 Acrescentando restrições institucionais	55
3.2.6 Modelando impostos e subsídios	58
3.3 Os dados: descrição do banco de dados do GTAP	63
3.3.1 O arcabouço de modelo do GTAP	64
3.3.2 Os arquivos de dados do GTAP	71
4. Aplicações	81
4.1 Aplicação do método ao modelo e dados do GTAP	82
4.2 Um modelo GTAP8x9 com países do Mercosul	90
5. Conclusão	97
6. Referências Bibliográficas	101
7. Apêndices	107
A1 – O armazenamento simbólico de expressões analíticas	107
A2 – O método Johansen/Gempack	111
A3 – O algoritmo SLCP, para resolução de $f(z) \geq 0 : z \geq 0$	118
A4 – O algoritmo LCP, para resolução de $M(z)+q \geq 0 : z \geq 0$	123
A5 – O algoritmo ACI, para resolução de $f(z) = 0$	127
A6 – Tabelas GTAP	141
A7 – O pacote GTAPproc	153
A8 – Tabelas do modelo GTAP8x9	167

1. Introdução

*Human knowledge and human power meet in one;
for where the cause is not known the effect cannot be produced...
(Francis Bacon, Aphorism III, First Book of Aphorisms)*

A busca pela compreensão das causas e efeitos dos fenômenos que nos cercam é uma das características mais notáveis da raça humana. O processo se inicia com a observação dos fenômenos em alguma área de interesse. Com estudo e reflexão, dados observados se transformam em conhecimentos, e com rigor científico o conjunto de conhecimentos é organizado e disciplinado em uma teoria.

A construção de um modelo explicativo baseado em fundamentos teóricos que possibilite simular os fenômenos estudados é um importante passo auxiliar no processo, possibilitando focalizar idéias e visualizar resultados. Scarf [57] relata que Irving Fisher, um dos pais fundadores de Finanças, em "*Mathematical Investigations in the Theory of Value and Prices*" [24], publicado em 1892, descreveu uma máquina analógica – mecânica e hidráulica – para o cálculo dos preços de equilíbrio de um modelo de equilíbrio geral competitivo:

"... Pelo menos duas versões da máquina de Fisher foram construídas e aparentemente funcionaram com sucesso. As máquinas desafortunadamente se perderam, mas existem diversas fotografias que podem ser vistas na reedição do trabalho de Fisher realizada em 1961 pela Yale University Press.

O equipamento parece antiquado e estranho nesta era de computadores digitais. Imersos em um grande tanque cheio de água existem diversos recipientes com perfis irregulares relacionados com as utilidades marginais dos consumidores para os vários bens. Cada recipiente é

construído parcialmente de couro flexível, parecendo como um fole que se expande e contrai em resposta à mudanças nos preços. Os recipientes são conectados por um sistema elaborado de varas, dobradiças e tubos cheios de água.

Para especificar as receitas iniciais em dinheiro dos consumidores, um conjunto de êmbolos é ajustado nos tamanhos adequados, e no modelo de troca pura, uma série similar de ajustamentos é feita para providenciar informação sobre os estoques iniciais de bens antes que as trocas aconteçam. Os níveis dos preços competitivos e as alocações de bens são então determinadas quando o sistema atinge um estado físico de equilíbrio."

O objetivo deste trabalho é o mesmo de Fisher em seu experimento – uma máquina para calcular os preços de equilíbrio de uma economia. Se então os melhores recursos tecnológicos disponíveis na época foram utilizados, aqui também o serão: será utilizada uma máquina virtual em um computador digital.

Assim, aqui serão apresentados métodos quantitativos para a implementação, calibragem e resolução de modelos computáveis de equilíbrio econômico de grande porte expressos como rotinas computacionais destinadas ao pesquisador em economia, que se supõe ter apenas um conhecimento trivial com o uso de um editor de texto em computadores. Por meio dessas rotinas, dados e modelo econômico informados pelo pesquisador em arquivos de texto são convertidos no problema matemático equivalente, que por sua vez é resolvido por um algoritmo numericamente eficiente, e os resultados obtidos colocados de volta em um arquivo texto.

As rotinas fazem parte do pacote *Pegasus*, introduzido por Scaramucci [56] e Bordoni [5], um sistema de arquitetura aberta de suporte computacional para análise

aplicada de equilíbrio econômico, composto por dois conjuntos principais de procedimentos. Um desses conjuntos é voltado para a informação e calibragem dos dados econômicos e a construção do modelo matemático, e foi desenvolvido em *Mathematica* (um ambiente de programação numérica e simbólica) por Scaramucci [56].

O segundo conjunto, ao qual se deu fundamentalmente a participação do autor deste trabalho, é voltado para a resolução numérica do modelo matemático, que em [5] era feita por uma versão em linguagem C do procedimento LCP – originalmente codificado em Fortran por Rutherford [52].

Em uma contribuição que se apresenta neste trabalho é acrescentada ao *Pegasus* a capacidade de resolução de modelos de grande porte, através de uma implementação do algoritmo ACI (Atualização de uma Coluna por Iteração), proposto por Martínez [41] e Gomes-Ruggiero [27]. O novo procedimento ACI é dotado das seguintes características:

1) Alocação dinâmica da memória em função do tamanho do modelo, evitando a necessidade de se dimensionar o programa para o maior modelo que possa ser resolvido.

2) Estrutura de dados para armazenamento das equações do modelo em forma simbólica e procedimentos para sua avaliação (ver apêndice A1), evitando a necessidade de elaboração e compilação de rotinas em linguagens de programação para cada novo modelo.

3) Cálculo do jacobiano do sistema por diferenças finitas, evitando a necessidade de informar ou calcular analiticamente o jacobiano.

4) Fatoração da matriz jacobiana através do pacote Harwell/MA28 [48].

Juntamente com o armazenamento esparsa de matrizes já existente na versão preliminar, essas características viabilizam a resolução de modelos econômicos de grande porte de maneira eficiente e flexível.

A modelagem e resolução de equilíbrios econômicos computáveis está apoiada em um tripé composto por dados, modelo econômico e método de resolução. Embora este trabalho esteja centrado no método de resolução, dada a interdependência do conjunto, necessariamente também serão abordados dados e modelos econômicos. Destaque-se que embora sejam fundamentais a corrente teórica em que o modelo econômico está inserido e a qualidade dos dados utilizados para a validade e aplicação das análises econômicas quantitativas obtidas, estes dois componentes do tripé acima mencionado aqui não serão discutidos em profundidade, restringindo-se este trabalho ao estudo do terceiro componente, o método de resolução do problema matemático equivalente.

Quanto ao modelo econômico, o *Pegasus* utiliza a formulação do equilíbrio econômico como um problema de complementaridade não-linear apresentada por Mathiesen [42]. Essa formulação supõe um modelo walrasiano de equilíbrio competitivo, que pode ser estendido para incluir distorções da economia como impostos e subsídios, competição imperfeita, "mark-up" na formação de preços, ou restrições para acomodar elementos macroeconômicos no modelo.

Com relação aos dados, supõe-se apenas que as informações utilizadas sejam consistentes como as de uma Matriz de Contabilidade Social (*Social Accounting Matrix*). No exemplo de aplicação dos métodos a um modelo de grande porte aqui apresentada serão utilizadas as informações do banco de dados do GTAP (*Global Trade Analysis Project*) [34], que tem por objetivo principal a análise de questões econômicas globais multi-regionais relacionadas à comércio, tecnologia, recursos e ambiente. (O GTAP é uma associação de mais de 400 pesquisadores do mundo todo interessados em conduzir análises quantitativas de questões econômicas globais que oferece, basicamente, um arcabouço de modelo global e os dados correspondentes, sendo que a nova versão 5.0 do banco de dados

inclui informações de comércio bilateral, transporte, proteção e insumo-produto para 57 setores produtivos e 65 regiões, para o ano de 1997.)

É importante destacar que os métodos computacionais implementados no *Pegasus* não se aplicam somente ao arcabouço de modelo e dados do GTAP, podendo ser utilizados para resolver um espectro variado de modelos CEG.

Existem na literatura diversos modelos CEG para o Brasil, sendo que primeiro foi o de Lysy e Taylor [38], seguido por diversos outros (ver Guilhoto [25]). Citando-se apenas os modelos aplicados de grande porte mais recentes apresentados para o Brasil, os métodos do *Pegasus* poderiam ser utilizados para modelos do tipo do EFES, apresentado por Haddad e Domingues [29] para projeções setoriais da economia brasileira, do modelo para análise de câmbio e tarifas apresentado por Najberg et al. [46], do MEGABRAS, apresentado por Ferreira Filho [22] em um estudo da evolução da agricultura brasileira, e do PAPA, apresentado por Guilhoto [25] para planejamento de políticas agrícolas.

É crescente a demanda por análises quantitativas de questões econômicas em diversas áreas, tais como comércio internacional, desenvolvimento e crescimento econômico, agricultura e meio ambiente. Os métodos de implementação, calibragem e resolução de modelos computáveis de grande porte aqui apresentados poderiam ser utilizados em todas essas áreas.

2. Revisão da Literatura¹

*If I have been able to see further,
it was only because I stood on the shoulders of giants.
(Isaac Newton)*

Em meados do século 18 o economista francês François Quesnay buscava compreender as causas e efeitos dos fenômenos que observava: (i) o proprietário de terras que recebe uma certa quantia de dinheiro pelo arrendamento de suas propriedades e gasta metade em produtos agrícolas e metade em produtos de artesãos, (ii) o agricultor que usa parte de suas receitas para pagar o arrendamento da terra e parte para pagar os produtos industriais necessários à produção agrícola, (iii) artesãos que compram comida e matérias primas, e assim por diante. Em o "*Tableau Économique*" [49], publicado em 1758, Quesnay propõe contabilizar os gastos dos agentes de uma economia em uma representação tabular das quantidades gastas, por produto e agente.

Mais de um século depois Léon Walras, outro economista francês, utilizou uma forma tabular semelhante à de Quesnay para representar a demanda derivada da produção de bens. Entretanto, utilizar uma matriz de coeficientes de produção foi apenas uma faceta de seu marcante trabalho, "*Éléments d'Économie Politique Pure*" [64], publicado em 1874, onde Walras aplicou noções de mecânica newtoniana no desenvolvimento de uma teoria de equilíbrio geral econômico descrevendo as interações de produtores e consumidores por equações de excesso de demanda, sendo que preços e quantidades produzidas dos bens deveriam ser determinados simultânea e interdependentemente de maneira a satisfazer a igualdade entre produção e consumo.

¹ Esta seção é baseada em Miller & Blair [43], Manne [40], Robinson [50] e Dixon et alii [16].

Walras é considerado o fundador da teoria de equilíbrio geral, e modelos baseados em seus fundamentos são conhecidos como modelos *walrasianos*, em contraposição a modelos econômicos em que os preços são determinados simplesmente por fatores tecnológicos sem depender da demanda de consumidores.

Quanto à existência de preços de equilíbrio, aparentemente Walras se satisfazia em observar que as condições de esgotamento dos mercados (*market clearing*), quando colocadas na forma matemática apropriada, correspondiam a um sistema com tantas equações quanto incógnitas. Embora essa condição geralmente seja uma garantia de existência de solução para um sistema de equações *lineares*, quando se consideram equações *não-lineares* pode ocorrer que o sistema não tenha solução, ou, existindo uma solução, esta não tenha significado sob o ponto de vista econômico.

A prova teórica da existência de tais preços e quantidades seria apresentada quase um século depois, em 1954, quando Keneth Arrow e Gerard Debreu formalizam um modelo walrasiano lógico preciso das interações de consumidores e produtores em "*Existence of equilibrium for a competitive economy*" [4], sendo as equações de excesso de demanda formuladas matematicamente como um mapeamento de preços em preços, onde pelo menos um "ponto" (um vetor n -dimensional de preços) é mapeado em si mesmo, ou seja, um mapeamento de "ponto fixo". A prova, entretanto, não era *construtiva* – no sentido de possibilitar um método de cálculo da solução de equilíbrio – o que aconteceria somente em 1974 com o algoritmo de ponto fixo apresentado por Herbert Scarf [58], unindo as proposições teóricas de existência com o cálculo do equilíbrio em modelos especificados numericamente.

A seguir neste capítulo serão apresentados métodos para resolução de modelos *computáveis* de equilíbrio econômico – modelos que possam ser implementados

empiricamente e cujos preços e quantidades de equilíbrio dos bens possam ser calculados. Serão apresentadas em maior detalhe os métodos que se aplicam a modelos computáveis de *grande porte*, com ênfase no modelo matemático, nas técnicas computacionais de resolução e nos programas de computador (*pacotes*) que as implementam.

2.1 Aproximações lineares do modelo walrasiano

Já em 1936, muito antes da prova construtiva de existência do equilíbrio, se inicia o desenvolvimento empírico de modelos computáveis de equilíbrio econômico, com Wassily Leontief utilizando uma "tabela de transações" para descrever a economia dos Estados Unidos da América do Norte – na descrição do próprio Leontief, "uma tentativa de construir um Tableau Économique dos Estados Unidos" – em "*Quantitative Input-Output Relations in the Economic System of the United States*" [36].

O método apresentado por Leontief, hoje conhecido como *análise de insumo-produto*, é uma aproximação do modelo walrasiano que supõe tecnologias lineares de produção, redundando em um sistema de equações lineares resolvido por inversão de matrizes. O modelo original de Leontief considerava endógenos todos os setores da economia, resultando em um sistema de equações lineares homogêneas com solução não trivial múltipla.

A resolução do modelo apresentado era feita usando inicialmente máquinas mecânicas de calcular de grande capacidade em 1935, e posteriormente em 1943, o Mark I, um calculador automático eletro-mecânico controlado por fita perfurada – os computadores só iriam surgir em 1946 com o ENIAC, o primeiro computador eletrônico digital.

A partir da década de 50, modelos lineares de insumo-produto foram muito utilizados em planejamento de desenvolvimento, sendo esse método hoje estendido por muitos trabalhos, incluindo modelos dinâmicos, modelos de consistência inter-setorial e o uso da Matriz de Contabilidade Social. Por um lado, modelos de insumo-produto são muito úteis, pois a partir de suposições muito simples de tecnologias e demanda consegue-se representar a interdependência dos principais componentes de uma economia; por outro lado, não possuem um mecanismo endógeno para preços.

Outra aproximação para o modelo walrasiano usa técnicas de programação linear²: em 1952 Samuelson publica "*Spatial price equilibrium and linear programming*" [55], onde o equilíbrio econômico é formulado como um problema de otimização resolvido por programação linear. Esse método introduz a maximização explícita da função preferência de um consumidor representativo e restrições de desigualdade na estrutura linear de igualdade do modelo de insumo-produto, permitindo a escolha de soluções alternativas.

Embora tenham surgido diversos trabalhos na comunidade acadêmica com o método de programação linear, a utilização em modelos aplicados nunca chegou a se firmar devido a dificuldades teóricas e práticas: especializações irreais e comportamento extremo (*bang-bang*) originados das suposições de linearização, a determinação da ponderação da função objetivo em modelos para vários consumidores, a interpretação dos preços-sombra, etc. Os modelos então existentes podem ser vistos em Manne [39]. As dificuldades teóricas dos modelos de programação linear são discutidas em Taylor [62]. Vantagens e desvantagens dos métodos de insumo-produto e programação linear são apresentadas em Dervis et al. [14].

² Conforme relato de Dantzig e Cottle [11], a primeira implementação em computador do método simplex foi feita em 1951, no computador SEAC do United States Bureau of Standards.

2.2 Modelos computáveis de equilíbrio geral

O primeiro modelo computável de equilíbrio geral (modelo CEG) foi apresentado por Johansen em 1960 em *A Multi-Sectoral Study Of Economic Growth* [35], uma descrição da economia norueguesa através de um modelo computacionalmente simples, formulado de maneira a reagir às variações de preços, assim permitindo um tratamento satisfatório das possibilidades de substituição no consumo e na produção. Em seu método, basicamente, as equações que caracterizam um equilíbrio walrasiano competitivo eram diferenciadas em logaritmos em relação ao tempo, obtendo-se com essa transformação um sistema de equações simultâneas, lineares em taxas de crescimento.

O objetivo principal de Johansen era prever o desenvolvimento da economia norueguesa, e para isso dividiu o sistema linearizado em duas partes, uma parte dependente de variáveis que considerou endógenas, e outra parte dependente de variáveis que considerou exógenas, fixando estas últimas em valores previstos para um momento futuro por técnicas econométricas externas ao modelo. O sistema resultante desse particionamento era resolvido por fatoração da matriz de coeficientes, obtendo-se como solução uma previsão das taxas de crescimento das variáveis endógenas em função das taxas de crescimento previstas para as variáveis exógenas.

2.2.1 Os modelos originados por Johansen: ORANI, GTAP

O método originado por Johansen foi aprofundado e estendido em um modelo CEG de grande porte para a Austrália – o modelo ORANI – por Dixon et al. [17] (cuja extensão

mais recente é o modelo MONASH [44]), e tem sido usado em várias aplicações desde então, sendo o método padrão de modelagem e resolução do GTAP.

O formato atual supõe que o modelo walrasiano possa ser representado por um sistema de equações descrevendo *i)* a demanda final por bens, *ii)* as decisões da indústria (descrevendo demandas de insumos e composição de produtos), *iii)* o lucro econômico zero (relacionando preços e custos dos bens), *iv)* as condições de esgotamento de mercados e *v)* outras equações, incluindo agregados úteis tais como índices de preços, fluxos agregados de comércio, emprego agregado, rendas dos consumidores, renda líquida do governo, etc.

Essas equações devem também impor que as demandas dos consumidores sejam restritas por sua renda e visem a maximização da utilidade, as condições de lucro econômico zero sejam aplicadas, que a demanda e oferta se igualem. Supõe-se que as preferências e tecnologias sejam representadas por funções utilidade e tecnologia diferenciáveis.

Supondo-se que esse sistema de equações possa ser representado por $F(\mathbf{V}) = \mathbf{0}$, onde F é uma função vetorial m -dimensional de n variáveis (representadas pelo vetor \mathbf{V}), diferenciável, com $n > m$, então, como no método original, aplicando-se a diferenciação total em $F(\mathbf{V}) = \mathbf{0}$, obtém-se a versão linearizada

$$\nabla F(\mathbf{V}) d\mathbf{V} = \mathbf{0},$$

onde $\nabla F(\mathbf{V})$ é o jacobiano de $F(\mathbf{V})$. Fazendo $\mathbf{A}(\mathbf{V}) = \nabla F(\mathbf{V})$ e $\mathbf{v} = d\mathbf{V}$, tem-se:

$$\mathbf{A}(\mathbf{V}) \mathbf{v} = \mathbf{0}$$

onde $\mathbf{A}(\mathbf{V})$ é uma matriz $m \times n$ cujos componentes são função de \mathbf{V} , e \mathbf{v} é o vetor n -dimensional de variações de \mathbf{V} .

Na prática, modeladores experientes dispensam a construção preliminar da função F , trabalhando diretamente no sistema $A(V) v = 0$, onde outras transformações, geralmente, podem ainda ser aplicadas à matriz $A(V)$, possibilitando que o vetor v seja interpretado como variações percentuais ou variações em logaritmos, mais adequadas às análises econômicas.

Para a obtenção de resultados, como no modelo de Johansen, as variáveis v devem ser particionadas em endógenas (v_1) e exógenas (v_2). A partir de uma solução de equilíbrio V^* , para calcular as alterações nas variáveis endógenas em resposta às alterações dadas nas variáveis exógenas, $A(V^*)$ deve também ser particionada, obtendo-se

$$A_1(V^*) v_1 + A_2(V^*) v_2 = 0, \quad \text{com } v_2 \text{ dado,}$$

onde os vetores v_1 e v_2 têm dimensões m e $(n-m)$ respectivamente, e as matrizes A_1 e A_2 (formadas pelas colunas de A correspondentes aos elementos de v_1 e v_2) têm dimensões $m \times m$ e $m \times (n-m)$ respectivamente. Supondo-se que $A_1(V^*)$ tenha inversa e rescrevendo a expressão, tem-se:

$$v_1 = -A_1(V^*)^{-1} A_2(V^*) v_2, \quad \text{com } v_2 \text{ dado.}$$

Esse problema tem sido muito estudado, sendo conhecido na literatura como um "problema de valor inicial de equações diferenciais ordinárias", e no Apêndice A2 é feita uma apresentação detalhada desse método de resolução.

2.2.2 Os modelos originados por Adelman e Robinson

No início da década de 70, inspirados no conceito de modelos de equilíbrio geral resolvidos numericamente originado por Johansen, Adelman e Robinson [2] apresentam um modelo CEG para países em desenvolvimento, um modelo não-linear, multi-setorial que

procurava simular o funcionamento de uma economia de mercado, resolvendo para preços e quantidades de mercado simultaneamente. A formulação então apresentada tem sido utilizada em muitos trabalhos, sendo uma das indicadas pelo Banco Mundial para modelos de equilíbrio econômico .

Nesse enfoque, o modelo CEG é tratado como uma coleção de equações algébricas não-lineares resolvida por meio de métodos numéricos. Em Dervis, de Melo e Robinson [14] é apresentada a formulação e o método de resolução então utilizado, este último consistindo em uma *estratégia de solução* e um *algoritmo de solução*.

Por estratégia de solução os autores se referiam à maneira como as equações do modelo eram substituídas e rearranjadas para reduzir ao máximo possível o tamanho do problema antes de resolvê-lo em um computador. Por algoritmo de solução se referiam ao algoritmo numérico utilizado em computador para resolver o conjunto reduzido de equações (o algoritmo de solução então utilizado era um método quase-newtoniano do tipo secante proposto por Broyden [7], com a implementação de Powel [47]).

A estratégia de solução básica era reduzir as equações do modelo por substituição a dois conjuntos de equações de excesso de demanda, um para o mercado de fatores e outro para o de produtos; isso possibilitaria que no algoritmo de solução fosse utilizado um procedimento iterativo: por exemplo, a partir de preços de fatores (*wages*) iniciais estimados se obtêm os preços dos produtos – com os quais se obtêm novos preços para os fatores e o processo torna a ser repetido – até que seja obtida uma solução de equilíbrio para esses dois conjuntos de mercados.

Castro [9] analisa esse e outros procedimentos iterativos semelhantes de outros autores em que preços de fatores e produtos são obtidos "em série", e a possibilidade de que procedimentos desse tipo possam dar margem a soluções não-walrasianas; questiona

também modelos apresentados por outros autores onde se encontra, muitas vezes, "*a complex and at times confusing mixture of economic theory, mathematics, computer algorithms, econometric equations not to mention various ad hoc assumptions applied when the model is programmed for the computer*".

Essa situação, em que era necessário construir os programas de computador e que os modelos tinham que ser manipulados até se "encaixarem no computador" para produzir resultados, se altera a partir de 1988, quando Brooke, Kendrick e Meeraus introduzem o GAMS (*General Algebraic Modeling System*) [8] (que será descrito em uma próxima seção), oferecendo uma linguagem de modelagem independente do algoritmo de resolução, possibilitando aos modeladores dedicarem seus esforços principalmente à construção do modelo econômico, sem preocupações com a construção de programas de computador.

A formulação de Adelman e Robinson, bem como a de trabalhos independentes de vários outros autores (como por exemplo Dixon [15]) podem ser vistas como uma consequência natural de modelos de insumo-produto acrescidos da possibilidade de substituição neoclássica na produção e demanda, resultando em um sistema de equações de excesso de demanda complementadas por outras equações para dar consistência ao modelo (ver Robinson [50], p. 888). Como resultam de uma linha de trabalho com ênfase maior na experimentação numérica da teoria econômica, poderiam ser chamados de "modelos de economistas".

2.2.3 Os modelos originados por Scarf e Mathiesen

Uma outra linha de trabalho, com ênfase maior na experimentação de técnicas matemáticas (e portanto, em contraposição, "modelos de matemáticos"), tem origem no

trabalho de Arrow e Debreu [4], que formulam o equilíbrio econômico como um problema de achar um ponto fixo em um mapeamento de preços em preços, e Scarf [58] que apresenta um algoritmo para resolver esse problema. O método também é descrito em Scarf [59] e variações em Broadie [6] e Shoven e Whalley [61]. Alguns exemplos de aplicação são Shoven e Whalley [60], Whalley [63] e Feltenstein [19] inicialmente e Feltenstein [20], York [67], Gropp [28] e Feltenstein e Shah [21] mais recentemente.

Um *mapeamento* pode ser entendido como uma generalização da noção de função. Considerando espaços n -dimensionais, o problema de achar um ponto fixo, ou seja, um ponto que seja mapeado em si mesmo, pode ser resolvido através de técnicas combinatoriais, como a de Scarf. Para funções homogêneas de grau zero pode-se provar que o algoritmo apresentado por Scarf sempre converge para uma solução. Entretanto, mesmo para modelos de tamanho moderado, o algoritmo de ponto fixo pode ser excessivamente caro em termos computacionais.

O problema de encontrar um ponto fixo também pode ser formulado de maneira equivalente como um problema de complementaridade, que no caso linear dispõe de um algoritmo combinatorial eficiente, proposto por Lemke [37], apresentado no apêndice A4. Utilizando esse método, Mathiesen [42] apresenta um formato de complementaridade não-linear para resolução do modelo CEG, resolvido por uma seqüência de problemas de complementaridade linear, apresentado no apêndice A3. Nesse caso, como as equações de excesso de demanda da formulação de Arrow-Debreu são definidas como sendo menores ou iguais a zero, e o formato de complementaridade é definido em termos de equações maiores ou iguais a zero, utiliza-se o artifício de multiplicar as equações de excesso de

demanda por -1, obtendo-se equações equivalentes que devem então ser chamadas, mais adequadamente, de *equações de excesso de oferta*.

Em 1988 Rutherford [52] apresenta o MPSGE (*Mathematical Programming System for General Equilibrium Analysis*), baseado no método de Mathiesen onde a linguagem de modelagem do problema econômico é independente do método de resolução.

São também dessa época uma série de trabalhos de um grupo de alunos de pós-graduação do Departamento de Matemática Aplicada do IMECC, da Unicamp: o estudo de um algoritmo de ponto fixo para o cálculo do equilíbrio econômico de Arica [3], uma implementação em Fortran do método de Mathiesen para o cálculo do equilíbrio econômico de Rotondo [51], a resolução do problema de equilíbrio econômico por complementaridade, em uma versão preliminar do *Pegasus*, de Bordoni [5], o cálculo de equilíbrios econômicos por otimização de Zullo [68] e o tratamento de economias distorcidas acrescentado a essa versão do *Pegasus* por Proença [48].

2.3 Pacotes para modelagem e resolução de modelos CEG

O objetivo maior no desenvolvimento de pacotes para modelagem e resolução de equilíbrios computáveis é liberar os pesquisadores da maior parte das dificuldades relacionadas à utilização de linguagens de programação de computadores, permitindo que se concentrem nos aspectos econômicos de seus modelos.

Existem diversos pacotes para resolução de modelos CEG. Para modelos de grande porte, os mais conhecidos são o GAMS (Brooke et al. [8]), que atualmente tem o MPSGE (Rutherford [53]) incorporado, e o Gempack (Harrison e Pearson [31]).

Será vista a seguir uma breve apresentação desses pacotes para modelos de grande porte citados, e também do *Pegasus*, que a partir das contribuições apresentadas neste trabalho também pode ser relacionado nessa classificação

2.3.1 O pacote GAMS

O pacote GAMS (*General Algebraic Modeling System*) é um pacote de modelagem para resolução de problemas de programação matemática, representados como sistemas de equações algébricas. Foi desenvolvido no Banco Mundial por Meeraus e seus colegas, sendo o manual de utilização do sistema da autoria de Brooke, Kendrick e Meeraus [8].

No projeto do pacote foram fundidos conceitos da teoria de banco de dados relacionais e de programação matemática visando atender as necessidades de planejamento estratégico de modeladores. A teoria de banco de dados relacionais fornece a estrutura necessária para a organização e manipulação dos dados, e a programação matemática uma maneira de descrever um problema e diversos métodos para resolvê-lo.

Uma característica importante do pacote é serem os modelos definidos de maneira independente dos métodos de resolução, possibilitando que modeladores possam dedicar todos seus esforços principalmente para a definição das equações algébricas do modelo (sem preocupações com a construção de programas de computador), com a certeza de que, caso surjam novos métodos de resolução mais eficientes que os existentes, poderão ser utilizados sem que sejam necessárias alterações no modelo.

A linguagem GAMS é direcionada para conjuntos, e foi inicialmente projetada para facilitar o desenvolvimento de modelos de programação linear e não-linear na indústria e na agricultura. Assim, não seria necessário escrever uma equação para cada bem, tempo,

fábrica ou mercado: as equações são escritas para os conjuntos, possibilitando uma grande economia de tempo (e erros!) no desenvolvimento de um modelo. Na versão mais recente podem ser resolvidos problemas de otimização lineares, não-lineares, inteiros mistos (*mixed integer*), inteiros mistos não-lineares, além de problemas de complementaridade mistos (*MCP - Mixed Complementarity Problem*).

Atualmente o GAMS é utilizado para quase todo tipo de modelos computáveis que possam ser expressos como um sistema de equações algébricas. Desde o início do desenvolvimento do sistema, modelos representativos são colecionados em uma biblioteca, fornecida com o sistema (e também disponível na página da Internet do GAMS, <http://www.gams.com>) com muitos exemplos de modelos CEG. Existem versões do pacote GAMS para os mais diversos tipos de sistemas operacionais e computadores, desde microcomputadores até computadores de grande porte.

Os comandos em linguagem GAMS têm uma sintaxe aproximada de definições em matemática, visando a definição de relações algébricas livres de ambigüidade. Entretanto, o domínio adequado dessa sintaxe pode exigir um tempo significativo dos modeladores (como também o entendimento das listagens com resultado), embora muito menor que a construção de programas de computador específicos equivalentes.

2.3.2 O pacote Gempack

O Gempack é um sistema computacional para a implementação e resolução de modelos de equilíbrio econômico geral e parcial, podendo tratar um conjunto abrangente de comportamentos econômicos. O pacote tem sido utilizado em mais de 50 organizações pelo mundo afora (universidades, instituições públicas e empresas do setor privado) em uma

grande quantidade de modelos econômicos: para diversos países (como os modelos ORANI [17] e MONASH [44]), modelos de comércio multi-regionais (modelo GTAP [34]), modelos regionais e modelos intertemporais (ou dinâmicos). O sistema pode ser instalado em diversos tipos de computadores e oferece também um conjunto de programas utilitários para a manipulação do banco de dados econômicos e dos resultados das simulações.

No Gempack a implementação do modelo econômico é feita a partir de uma representação algébrica das equações do modelo, inspirada na linguagem do GAMS. Essa linguagem algébrica não permite explicitamente desigualdades no sistema a ser resolvido, embora várias espécies de comportamento de otimização, cuja formalização seja complicada ou não conhecida analiticamente, possam ser representadas diretamente através das condições de primeira ordem do problema. As equações podem ser dadas em sua forma não-linear ou na forma linearizada – que é o padrão do pacote –, ou até mesmo em uma mistura das duas formas. Uma característica única do pacote é a condensação algébrica automatizada de equações, que torna possível a resolução de modelos muito maiores do que normalmente seria praticável.

2.3.3 O pacote *Pegasus*

O *Pegasus* (Bordoni [5], Scaramucci [56]), como o GAMS, é uma linguagem de modelagem independente do método de resolução.

Em sua versão preliminar, inteiramente codificada em Fortran, os modelos de equilíbrio econômico eram resolvidos pelo método de Mathiesen [42] com o suporte do MILES – um pacote para resolução de problemas de complementaridade linear baseado no Minos (Murtaugh et al. [44]) e apresentado por Rutherford [52]. O formato da linguagem

era inspirado no MPSX, um antigo pacote para resolução de problemas de programação linear da IBM. O formato da linguagem, apropriado para atividades lineares, era inadequado para representar atividades de produção e consumo não-lineares, restringindo novos desenvolvimentos.

Em uma nova versão, foram abandonados o formato MPSX e o Fortran, conservando-se, porém, o conceito da independência entre linguagem de modelagem e método de resolução. Optou-se, então, pelo *Mathematica* [66] – um pacote de programação numérica e simbólica –, que possibilitou o desenvolvimento de uma linguagem de modelagem de equilíbrio econômico estruturada e flexível, definida em termos de funções e procedimentos do *Mathematica*. Entretanto, o método de resolução continuou sendo baseado no MILES, porém agora como uma função externa ao *Mathematica*, em uma versão em linguagem C.

Devido à independência da linguagem e métodos de resolução foi possível depois acrescentar ao *Pegasus/Mathematica* uma implementação do método ACI de Gomes-Ruggiero [27] para resolução de sistemas de equações não-lineares de grande porte, sem que tivesse sido necessário alterar a linguagem e os modelos de equilíbrio econômico através dela definidos.

Uma descrição do *Pegasus/Mathematica* pode ser encontrada em Scaramucci [56]; as contribuições mais recentes são apresentadas ao longo deste trabalho.

3. Material e métodos

*Nessuna humana investigazione si pio dimandara vera scienza
s'essa non passa per le matematiche dimonstrazione.
(Leonardo da Vinci)*

As seções iniciais deste capítulo serão dedicadas a apresentar o método de cálculo de equilíbrio econômico modelado *sistemicamente* como um problema de complementaridade não-linear, como apresentado por Mathiesen [42], e às extensões à teoria contidas em Scaramucci [56], que serão vistas por meio de exemplos didáticos.

Os exemplos didáticos serão utilizados também para apresentar a implementação em computador do método, que utiliza o pacote *Pegasus*, um conjunto de procedimentos implementados em *Mathematica* (um ambiente de programação numérica e simbólica) e na linguagem C de programação de computadores: os procedimentos para informação dos dados e construção do problema de complementaridade equivalente ao problema de equilíbrio econômico foram desenvolvidos em *Mathematica* por Scaramucci no trabalho citado anteriormente, e os procedimentos para resolução numérica do problema de equilíbrio econômico foram desenvolvidos na linguagem C pelo autor como parte integrante deste trabalho, utilizando o método de resolução de sistemas não-lineares de grande porte apresentada por Gomes-Ruggiero [27].

Em seguida será feita a descrição dos dados do GTAP (*Global Trade Analysis Project* – Projeto Análise de Comércio Internacional), que serão utilizados no capítulo seguinte, onde em um exemplo serão apresentados os procedimentos desenvolvidos em *Mathematica* como contribuição deste trabalho para a aplicação do método a esses dados.

3.1 Equilíbrio econômico como um problema de complementaridade

O modelo walrasiano de equilíbrio econômico se caracteriza entre outras coisas por contrabalançar a produção e o consumo de bens, e postula a existência de preços de equilíbrio para os quais a oferta e a demanda de bens se igualam, com a possível exceção de que se houver excesso de oferta de algum bem, o preço desse bem deve ser zero.

O cálculo dos preços de equilíbrio pode ser modelado como um problema de complementaridade não-linear, como apresentado por Mathiesen [42]. O conceito de complementaridade será introduzido a seguir.

Em modelos matemáticos muitas vezes acontece que duas variáveis, por exemplo x e y (podendo x e y assumirem valores maiores ou iguais a zero, ou seja, $x \geq 0$, $y \geq 0$), apresentarem o seguinte comportamento:

$$\begin{cases} \text{se } x > 0 \text{ então } y = 0 \\ \text{se } y > 0 \text{ então } x = 0 \end{cases}$$

Em matemática diz-se que x e y são *complementares* e a representação matemática dessa *condição de complementaridade* se vale do fato de o produto do *par complementar* $x \geq 0$, $y \geq 0$ ser sempre zero, ou seja, $xy = 0$. Essa relação de complementaridade das duas variáveis geralmente é causada por existir alguma dependência entre x e y , e supondo que a função $f(x)$ represente essa dependência, então $y = f(x)$.

O problema de encontrar um valor de x tal que $y = f(x)$ satisfaça a condição de complementaridade $x \geq 0$, $y \geq 0$ e $xy = 0$ (ou a condição equivalente mais usada $xf(x) = 0$) é chamado um *problema de complementaridade*, e se a função $f(x)$ que representa a relação de dependência entre x e y for não-linear, ter-se-á um *problema de complementaridade não-linear*.

Deve ser ressaltado que todo problema de complementaridade é um sistema de equações de *desigualdades fracas*, possibilitando representar problemas de otimização envolvendo as condições de folga complementar.

O problema de complementaridade não-linear tem sido muito pesquisado, existindo algoritmos eficientes para sua resolução (ver Ferris e Pang [23]).

A formulação quando se tem $\mathbf{x} = \{x_1, \dots, x_n\}$ e $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_n(\mathbf{x})\}$ pertencentes ao \mathbf{R}^n é:

$$\left. \begin{array}{l} f_i(\mathbf{x}) \geq 0 \\ x_i \geq 0 \\ x_i f_i(\mathbf{x}) = 0 \end{array} \right\} \text{ para } i = 1, \dots, n$$

Ou, usando-se a forma vetorial:

$$\left. \begin{array}{l} \mathbf{f}(\mathbf{x}) \geq \mathbf{0} \\ \mathbf{x} \geq \mathbf{0} \\ \mathbf{x}^T \mathbf{f}(\mathbf{x}) = 0 \end{array} \right\} \text{ ou ainda: } \mathbf{f}(\mathbf{x}) \geq \mathbf{0} : \mathbf{x} \geq \mathbf{0}$$

A notação $\mathbf{f}(\mathbf{x}) \geq \mathbf{0} : \mathbf{x} \geq \mathbf{0}$, onde o sinal ":" significa "*complementar a*", é utilizada para representar o sistema acima. Observe-se que neste trabalho os nomes de vetores, matrizes e conjuntos são grafados em **negrito** e os de escalares em *itálico*, que também é utilizado para destacar conceitos importantes.

3.1.1 A definição da economia

Nos moldes do modelo de Arrow-Debreu [4], os componentes básicos de uma economia – bens, consumo e produção – serão definidos a seguir, e dessa base conceptual será formalizado o equilíbrio econômico como um problema de complementaridade.

Será considerado um bem de uma economia qualquer coisa que tenha um preço associado e para o qual haja oferta e demanda em um mercado correspondente. O conceito de bem utilizado não faz diferenciação quanto ao uso do bem, sejam insumos, fatores, bens de demanda intermediária da produção, bens destinados à demanda final ou mesmo serviços e abstrações como bens de capital. Também não serão feitas diferenciações de modelo, tipo ou cor de um bem, considerando-se um bem idealizado representativo do setor que o produz, embora essa hipótese possa ser relaxada para a introdução de competição imperfeita.

Dessa maneira, seja $\mathbf{B} = \{B_1, \dots, B_{n_b}\}$ o conjunto de bens da economia, com n_b elementos, e seja $\mathbf{p} = \{p_1, \dots, p_{n_b}\}$ o vetor coluna de preços associados aos bens, com a condição adicional que os preços sejam maiores ou iguais a zero.

Seja $\mathbf{C} = \{C_1, \dots, C_{n_c}\}$ o conjunto de consumidores, com n_c elementos, que dispõem, conforme um comportamento de não-saciação com relação aos bens consumidos, todas suas dotações iniciais de bens maximizando sua utilidade segundo suas preferências. Supõe-se ainda que sejam tomadores de preço e que tenham conhecimento perfeito dos preços dos bens.

Seja a função utilidade de cada consumidor $U_k(\mathbf{p})$, para $k = 1, \dots, n_c$, uma função adequada dos preços dos bens, como por exemplo as formas funcionais Cobb-Douglas ou CES (elasticidade constante de substituição) ou CDE (diferença constante de elasticidades) (ver Scaramucci [56]).

Considere-se que cada consumidor k possui dotações iniciais dos n_b bens, maiores ou iguais a zero, quantificadas em um vetor coluna $\mathbf{e}_k = \{e_{1,k}, \dots, e_{n_b,k}\}$. Seja $\mathbf{b} = \{b_1, \dots, b_{n_b}\}$

o vetor coluna das dotações iniciais *totais* dos bens, onde a dotação inicial total do bem i é obtida agregando-se as dotações iniciais individuais $e_{i,k}$ de cada consumidor:

$$b_i = \sum_{k=1}^{n_c} e_{i,k}, \quad \text{para } i = 1, \dots, n_b \quad (1)$$

Seja $S = \{S_1, \dots, S_{n_s}\}$ o conjunto de setores produtivos, com n_s elementos, com tecnologias CRS (retornos constantes à escala) e comportamento econômico competitivo, sendo tomadores de preço e tendo conhecimento perfeito dos preços dos bens. Seja $y = \{y_1, \dots, y_{n_s}\}$ o vetor coluna dos níveis de operação associados às atividades de produção, com a condição que os níveis sejam maiores ou iguais a zero.

A suposição de retornos constantes à escala possibilita que a produção de bens seja definida inicialmente por meio de *atividades lineares* de produção, e posteriormente estendida para *atividades não-lineares*.

No primeiro caso será considerada a tecnologia CRS Leontief, onde as atividades de produção são descritas através de coeficientes de insumo-produto de fatores fixos consistentes com nível de produção unitária. Os n_b coeficientes lineares de produção associados a cada atividade S_j são especificados em um vetor coluna $\mathbf{a}_j = \{a_{1,j}, \dots, a_{n_b,j}\}$, para $j = 1, \dots, n_s$. A forma matricial do conjunto de vetores coluna $\{\mathbf{a}_1, \dots, \mathbf{a}_{n_s}\}$ é a matriz \mathbf{A} de coeficientes tecnológicos, que terá n_b linhas e n_s colunas:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n_s} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n_s} \\ \cdot & \cdot & \dots & \cdot \\ a_{n_b,1} & a_{n_b,2} & \dots & a_{n_b,n_s} \end{pmatrix}$$

Considerando-se na matriz \mathbf{A} o coeficiente $a_{i,j}$ de um bem i e uma atividade j , para $i = 1, \dots, n_b$ e $j = 1, \dots, n_s$, convencionou-se que se $a_{i,j} < 0$ o bem i será um *insumo* da

atividade j e se $a_{i,j} > 0$ o bem i será um *produto* (bruto) da atividade j . Obviamente, se $a_{i,j} = 0$ o bem i não terá participação na atividade j .

Para o caso de atividades com tecnologias CRS não-lineares de produção (supondo-se que sejam funções adequadas dos preços dos bens, como as formas funcionais Cobb-Douglas, CES ou CDE), será utilizada a técnica de "geração de coluna", descrita em [56], que "gera" os coeficientes $a_{i,j}$ que compõem o vetor "coluna" \mathbf{a}_j da matriz \mathbf{A} que irá representar cada atividade não-linear. A partir da função de produção são obtidas as expressões dos coeficientes que minimizam o custo a preços \mathbf{p} , usando-se a notação $a_{i,j}(\mathbf{p})$ para indicar que cada coeficiente é uma função dos preços de todos os bens que compõe a atividade, e que será utilizada neste texto somente quando for necessário explicitar a dependência de preços. Esta extensão para atividades não-lineares não causará alterações no modelo teórico tendo em vista que o método de resolução a ser utilizado é iterativo – a cada iteração, os preços são fixos (aqueles obtidos na iteração anterior); quando substituídos em $a_{i,j}(\mathbf{p})$ os transformam em coeficientes fixos.

Após a caracterização da demanda e oferta sob o ponto de vista de consumidores e setores produtivos respectivamente, obtém-se as expressões das funções de oferta e de demanda agregadas.

A *oferta de mercado* (líquida)³ para o bem i é a soma das dotações iniciais totais b_i com a quantidade produzida do bem i descontadas as quantidades do mesmo bem usadas como insumo em outras atividades. No caso de atividades CRS, a produção é obtida para cada bem i agregando-se no conjunto das atividades o produto do coeficiente $a_{i,j}$ do bem i pelo nível y_j da atividade j . Observe-se que como o sinal dos coeficientes dos bens usados

³ Se os preços forem estritamente positivos, a oferta de mercado será igual à demanda final, como será discutido na próxima seção.

como insumos é negativo, parte da oferta e a demanda *intermediária* do bem i se compensam automaticamente e o resultado dessa agregação é a quantidade *líquida* ofertada do bem i . Desse modo, a oferta $s_i(\mathbf{y})$ do bem i será:

$$s_i(\mathbf{y}) = b_i + \sum_{j=1}^{n_s} a_{i,j} y_j, \quad \text{para } i = 1, \dots, n_b \quad (2)$$

Ou, usando notação vetorial:

$$\mathbf{s}(\mathbf{y}) = \mathbf{b} + \mathbf{A} \mathbf{y} \quad (3)$$

Para obter a *demanda de mercado* do bem i se aplica a técnica de geração de coluna à função utilidade de cada consumidor (ver [56]), obtendo-se a partir de $U_k(\mathbf{p})$ as quantidades $d_{i,k}(\mathbf{p})$ de cada bem i demandadas pelo consumidor k . Como nas atividades de produção, a notação $d_{i,k}(\mathbf{p})$ será usada somente quando for necessário explicitar a dependência dos preços. A coluna gerada com as quantidades $d_{i,k}$ forma, por analogia com as atividades de produção, uma *atividade de consumo* $\{d_{1,k}, \dots, d_{n_b,k}\}$, e o conjunto de atividades de consumo forma uma *matriz de consumo*, com n_b linhas e n_c colunas.

A demanda de mercado do bem i é obtida agregando-se, entre todos consumidores, as demandas individuais do bem i , e é representada por uma função $d_i(\mathbf{p})$:

$$d_i(\mathbf{p}) = \sum_{k=1}^{n_c} d_{i,k}(\mathbf{p}), \quad \text{para } i = 1, \dots, n_b \quad (4)$$

Assim obtidas, as demandas $d_i(\mathbf{p})$ refletem os comportamentos de não-saciação no consumo de bens e maximização da utilidade pressupostos para os consumidores. Dessa maneira, a lei de Walras, que estabelece que o valor dos bens demandados deve ser igual ao valor total das dotações iniciais dos consumidores, será satisfeita:

$$\sum_{i=1}^{n_b} p_i d_i(\mathbf{p}) = \sum_{i=1}^{n_b} p_i b_i \quad (5)$$

3.1.2 A oferta excedente e o preço de um bem

Uma vez definidos os elementos básicos da economia, como preparação para a formalização do equilíbrio se analisará o comportamento do preço de um bem quando acontece um desequilíbrio entre a oferta e demanda.

Se há demanda excedente, o preço tenderá a crescer, estimulando crescimento da oferta e diminuição da demanda até se obter o equilíbrio novamente, a um preço maior e com a demanda excedente voltando a ser zero. Então, com relação à demanda excedente de um bem e seu preço, é possível afirmar que no equilíbrio *sempre* haverá apenas *uma* possibilidade: um preço maior que zero e uma demanda excedente igual a zero.

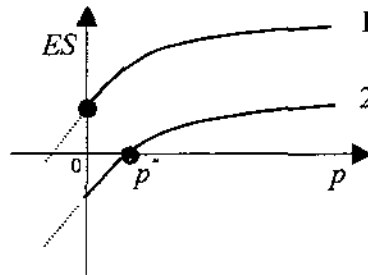
Se há oferta excedente, o preço tenderá a diminuir, ocorrendo um ajuste na oferta e na demanda até que a situação se estabilize novamente com um preço menor (ainda maior que zero) e com a oferta excedente voltando a ser zero. Entretanto, pode ocorrer que a demanda nunca se iguale à oferta, e assim o preço cairá a zero e a oferta excedente ainda continuará sendo positiva⁴.

Dessa maneira, considerando a oferta excedente de um bem e seu preço, o que se pode afirmar é que no equilíbrio *sempre* ocorrerá uma de *duas* possibilidades: *ou* o preço é maior que zero com a oferta excedente igual a zero, *ou* a oferta excedente é maior que zero com o preço igual a zero. Observe-se a existência de um comportamento complementar entre a oferta excedente de um bem e seu preço:

$$\begin{cases} \text{se oferta excedente} > 0 \text{ então preço} = 0 \\ \text{se preço} > 0 \text{ então oferta excedente} = 0 \end{cases}$$

⁴ A rigor, existe ainda a possibilidade matemática de oferta excedente e preço serem zero ao mesmo tempo, que corresponderia a um equilíbrio econômico degenerado, um caso limite de qualquer uma das duas possibilidades consideradas.

Essas duas possibilidades são mostradas no gráfico da oferta excedente ES em função do preço p do bem, a seguir. A curva 1 representa o caso de um bem que existe em abundância na natureza (o ar que respiramos, por exemplo), ou seja, a oferta excedente é positiva; sem escassez, o preço do bem deve ser zero. O caso ilustrado na curva 2 é o de um bem escasso em que oferta e demanda se igualam ($ES = 0$) a um preço p estritamente positivo:



Observe-se também que a oferta excedente de um bem tende tipicamente a ser uma função monotônica crescente do preço. Isso porque, se o preço de um bem é baixo, não há estímulo à sua produção, ao mesmo tempo em que pode haver uma grande procura por ele. Assim, tem-se oferta menor que demanda e, portanto, $ES < 0$. À medida que o preço sobe, a oferta aumenta e a demanda cai até que $ES = 0$ e, eventualmente, $ES > 0$.

Assim, voltando à proposição de equilíbrio econômico walrasiano do início desta seção, agora reformulada como um problema de complementaridade não-linear: afirma-se que uma economia como a descrita anteriormente está em equilíbrio se existirem preços dos bens para os quais:

E1) A oferta excedente de todos os bens seja maior ou igual a zero (ou seja, nenhum bem apresenta demanda excedente).

E2) O preço de todos os bens seja maior ou igual a zero.

E3) Se a oferta excedente de qualquer bem for maior que zero, então o preço desse bem deve ser zero. Por outro lado, se o preço de qualquer bem for maior que zero, então a oferta excedente desse bem deve ser zero.

Para formalizar as proposições econômicas E1–E3, supondo-se que s_i^* seja a oferta obtida considerando que y^* sejam os níveis ótimos das atividades de produção de bens a preços \mathbf{p} , então seja ES_i a oferta excedente de mercado do bem i , obtida pela diferença entre a oferta e a demanda desse bem:

$$ES_i(\mathbf{p}) = s_i^* - d_i(\mathbf{p}), \quad \text{para } i = 1, \dots, n_b \quad (6)$$

Ou, usando notação vetorial:

$$\mathbf{ES}(\mathbf{p}) = \mathbf{s}^* - \mathbf{d}(\mathbf{p}) \quad (7)$$

Assim, a representação matemática das proposições econômicas E1–E3 será dada por M1–M3 se existirem preços \mathbf{p} dos bens para os quais:

$$\left. \begin{array}{l} \text{M1) } ES_i(\mathbf{p}) \geq 0 \\ \text{M2) } p_i \geq 0 \\ \text{M3) } p_i ES_i(\mathbf{p}) = 0 \end{array} \right\} \text{ para } i = 1, \dots, n_b \quad (8)$$

Ou, usando notação vetorial:

$$\left. \begin{array}{l} \text{M1) } \mathbf{ES}(\mathbf{p}) \geq \mathbf{0} \\ \text{M2) } \mathbf{p} \geq \mathbf{0} \\ \text{M3) } \mathbf{p} \mathbf{ES}(\mathbf{p}) = 0 \end{array} \right\} \text{ ou } \mathbf{ES}(\mathbf{p}) \geq \mathbf{0} : \mathbf{p} \geq \mathbf{0} \quad (8')$$

3.1.3 O lucro negativo e o nível de operação de uma atividade

Como indicado anteriormente, o sistema (8) depende da obtenção de y^* , os níveis ótimos de operação das atividades, que são encontrados pelos setores produtivos como

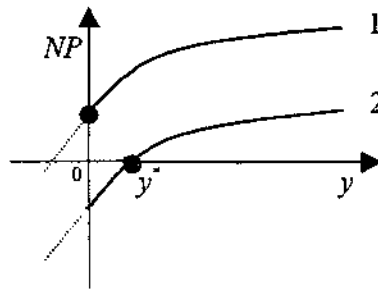
solução do problema de maximização do lucro de cada setor em função dos preços dos bens e da tecnologia utilizada.

Considerando que os setores são tomadores de preço (e portanto suas ações não afetam os preços da economia), e que as tecnologias usadas na produção dos bens são de *retornos constantes à escala*, tem-se que o lucro deverá ser menor ou igual a zero, pois se o lucro em algum setor fosse positivo, haveria a entrada de outros produtores nesse mercado, e o conseqüente excesso de oferta ocasionaria a diminuição do preço do bem até que o lucro se tornasse zero.

Sob essas hipóteses, se há prejuízo a atividade não será operada, e se a atividade é operada então o lucro deve ser zero. O possível lucro ou prejuízo é representado pelo conceito de *lucro negativo*, definido como o valor agregado dos insumos menos o valor do bem produzido. Nesse contexto, observe-se a existência novamente de um comportamento complementar, sendo o par complementar agora composto pelo nível de operação da atividade e o possível lucro ou prejuízo a ela associado:

$$\begin{cases} \text{se } \textit{lucro negativo} > 0 \text{ então } \textit{nível} = 0 \\ \text{se } \textit{nível} > 0 \text{ então } \textit{lucro negativo} = 0 \end{cases}$$

Como anteriormente, essas duas possibilidades são mostradas em um gráfico, nesse caso o do lucro negativo NP em função do nível y da atividade, a seguir. A curva 1 representa o caso de uma atividade de produção que dá prejuízo e por isso não é operada, ou seja, o lucro negativo é maior que zero e o nível de operação da atividade é zero. O caso ilustrado na curva 2 é o de um setor produtivo em que custos e receitas da produção se equivalem ($NP = 0$) a um nível de produção estritamente positivo.



Observe-se novamente a monotonicidade crescente das curvas acima. Se o nível de operação da atividade não é intensivo, o mercado de insumos é pouco pressionado, ao mesmo tempo que o mercado do produto ainda está pouco abastecido. Assim, o preço do produto tende a ser maior "relativamente" ao custo dos insumos – qualquer que seja o significado disso – e com isso, $NP < 0$. À medida que a atividade é operada mais intensivamente, essa situação tende a se inverter, obtendo-se $NP = 0$ e, eventualmente, $NP > 0$.

Assim, o conhecimento econômico para obtenção dos níveis ótimos de operação das atividades é dado pelas proposições:

E4) O lucro negativo de cada atividade deve ser maior ou igual a zero (ou seja, nenhuma atividade produz lucro).

E5) O nível de operação de todas as atividades deve ser maior ou igual a zero.

E6) Uma atividade com lucro negativo maior que zero tem nível de operação zero e se o nível de operação é maior que zero então o lucro negativo deve ser zero.

Seja NP_j o lucro negativo unitário da atividade j , obtido pela agregação do valor dos insumos necessários à produção de uma unidade do bem menos o valor unitário do bem. Como a convenção de sinal para os coeficientes $a_{i,j}$ é o inverso do que se deseja, a expressão para o lucro negativo unitário é obtida invertendo-se o sinal da agregação, para todos os bens da atividade, do produto dos coeficientes $a_{i,j}$ pelos preços correspondentes:

$$NP_j(\mathbf{p}) = - \sum_{i=1}^{n_b} p_i a_{i,j}, \quad \text{para } j = 1, \dots, n_y \quad (9)$$

Ou, usando notação vetorial:

$$\mathbf{NP}(\mathbf{p}) = -\mathbf{p}^T \mathbf{A} \quad (10)$$

E a representação matemática das proposições E4–E6 é dada por M4–M6:

$$\left. \begin{array}{l} \text{M4) } NP_j(\mathbf{p}) \geq 0 \\ \text{M5) } y_j \geq 0 \\ \text{M6) } y_j NP_j(\mathbf{p}) = 0 \end{array} \right\} \text{ para } j = 1, \dots, n_y \quad (11)$$

Ou, usando notação vetorial:

$$\left. \begin{array}{l} \text{M4) } \mathbf{NP}(\mathbf{p}) \geq \mathbf{0} \\ \text{M5) } \mathbf{y} \geq \mathbf{0} \\ \text{M6) } \mathbf{y} \mathbf{NP}(\mathbf{p}) = 0 \end{array} \right\} \text{ ou } \mathbf{NP}(\mathbf{p}) \geq \mathbf{0} : \mathbf{y} \geq \mathbf{0} \quad (11')$$

3.1.4 A formalização do equilíbrio econômico

Os sistemas M1–M3 e M4–M6 são interdependentes, e devem ser resolvidos simultaneamente como um sistema único M1–M6:

$$\left. \begin{array}{l} \text{M1) } ES_i(\mathbf{p}) \geq 0 \\ \text{M2) } p_i \geq 0 \\ \text{M3) } p_i ES_i(\mathbf{p}) = 0 \end{array} \right\} \text{ para } i = 1, \dots, n_b \quad (12)$$

$$\left. \begin{array}{l} \text{M4) } NP_j(\mathbf{p}) \geq 0 \\ \text{M5) } y_j \geq 0 \\ \text{M6) } y_j NP_j(\mathbf{p}) = 0 \end{array} \right\} \text{ para } j = 1, \dots, n_y$$

Ou, usando a notação de complementaridade:

$$\left. \begin{array}{l} \mathbf{ES}(\mathbf{p}) \geq \mathbf{0} : \mathbf{p} \geq \mathbf{0} \\ \mathbf{NP}(\mathbf{p}) \geq \mathbf{0} : \mathbf{y} \geq \mathbf{0} \end{array} \right\} \quad (12')$$

O sistema (12) deve satisfazer a lei de Walras, que estabelece que o valor total dos bens demandados deve ser igual ao valor total das dotações iniciais dos consumidores. De

fato, da condição de complementaridade M3 em (12), considerada no equilíbrio, e das expressões definidas em (7) e (3), se obtém:

$$\mathbf{p}^T \mathbf{b} + \mathbf{p}^T \mathbf{A} \mathbf{y} - \mathbf{p}^T \mathbf{d}(\mathbf{p}) = 0 \quad (13)$$

Da condição de complementaridade M6 em (12), considerada no equilíbrio, decorre a condição equivalente:

$$\mathbf{p}^T \mathbf{A} \mathbf{y} = 0 \quad (14)$$

Portanto, substituindo (14) em (13) tem-se:

$$\mathbf{p}^T \mathbf{b} - \mathbf{p}^T \mathbf{d}(\mathbf{p}) = 0$$

E finalmente, rearranjando os termos:

$$\mathbf{p}^T \mathbf{d}(\mathbf{p}) = \mathbf{p}^T \mathbf{b}$$

A expressão acima é a forma vetorial equivalente à expressão (5), e portanto o sistema (12) satisfaz a lei de Walras: se todos mercados menos um estão em equilíbrio, então o mercado restante também deverá estar em equilíbrio.

Como as funções de demanda são homogêneas de grau 0, o sistema é funcionalmente dependente (Dervis et al. [14], p. 495). De fato, se um vetor \mathbf{p} de preços é solução do sistema (12), o vetor $\alpha \mathbf{p}$ também o será, para qualquer α positivo. Essa indeterminação pode ser resolvida eliminando-se uma equação de oferta excedente, e o bem correspondente será o *numerário* da economia, com o preço fixado em nível unitário, sendo todos os outros preços a ele relativos. No equilíbrio, oferta e demanda serão iguais no mercado omitido, pela lei de Walras.

Como visto, tanto $\mathbf{ES} \geq \mathbf{0} : \mathbf{p} \geq \mathbf{0}$ como $\mathbf{NP} \geq \mathbf{0} : \mathbf{y} \geq \mathbf{0}$ são tipicamente funções monotônicas crescentes, e essa é uma condição importante para a existência de uma solução na teoria de complementaridade não-linear, como pode ser visto em Ferris e Pang [23], que fizeram um levantamento atualizado das aplicações desse problema em diversas áreas.

É usual renomear as equações e variáveis do sistema (12), substituindo as variáveis \mathbf{p} e \mathbf{y} pela variável \mathbf{z} , de dimensão n_b+n_y , e as funções **ES** e **NP** por uma função \mathbf{f} de dimensão n_b+n_y , como indicado no apêndice A3, obtendo-se:

$$\mathbf{f}(\mathbf{z}) \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0} \quad (12'')$$

Se por construção do modelo econômico pode-se garantir que na solução se tem $\mathbf{z} > \mathbf{0}$ (como será visto nas próximas seções), o problema (12'') se reduz ao sistema de equações não-lineares:

$$\mathbf{f}(\mathbf{z}) = \mathbf{0}, \text{ com } \mathbf{z} \geq \mathbf{0} \quad (15)$$

Nas próximas seções apresenta-se um exemplo didático que ilustra a utilização de dois métodos de resolução: o algoritmo SLCP (Seqüência de Problemas de Complementaridade Linear), proposto por Mathiesen [42], Rutherford [52], que pode ser aplicado tanto para o sistema (12) como para o problema (15), e o algoritmo ACI (Atualização de uma Coluna por Iteração), um método secante para resolução de sistemas não-lineares proposto por Martínez [41], Gomes-Ruggiero [27], que pode ser aplicado ao sistema (15), sendo mais eficiente que o SLCP para modelos de grande porte. Esses algoritmos são descritos nos apêndices "A3 – O algoritmo SLCP, para resolução de $\mathbf{f}(\mathbf{z}) \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0}$ " e "A5 – O algoritmo ACI, para resolução de $\mathbf{f}(\mathbf{z}) = \mathbf{0}$ ".

3.2 Extensões à teoria – Exemplos

Considere-se um exemplo didático muito simples de uma economia, composta de três bens, um consumidor e três setores produtivos.

Os bens são trabalho, capital e um produto (indicados por L, K e P), sendo \mathbf{p} o vetor de preços associados aos bens, com $p_L, p_K, p_P \geq 0$.

As dotações iniciais e_i dos bens possuídas pelo consumidor (indicado por C) são:

	C
L	85
K	40
P	0

Os consumidor dispõe as dotações iniciais e_i no consumo $d_i(\mathbf{p})$ de bens maximizando sua utilidade, aqui representada pela forma funcional Cobb-Douglas:

$$U(\mathbf{p}) = \prod_i d_i(\mathbf{p})^{\alpha_i}, \quad \text{onde } \sum_i \alpha_i = 1, \quad \text{para } i = L \text{ e } P$$

Os coeficientes α_i que expressam as preferências do consumidor por cada bem são fornecidos a seguir:

	C
L	0.2
P	0.8

Para a obtenção da demanda por bens basta lembrar que na forma funcional Cobb-Douglas da função utilidade os α_i são frações positivas que somam 1, indicando as frações da renda $r(\mathbf{p})$ gastas com cada bem. A demanda para o bem i será dada então por:

$$d_i(\mathbf{p}) = \frac{\alpha_i r(\mathbf{p})}{p_i}, \quad \text{onde } r(\mathbf{p}) = \sum_i e_i p_i, \quad \text{para } i = L \text{ e } P$$

Obtendo dessa maneira a demanda de cada bem, tem-se:

$$d_L = \frac{0.2 (85 p_L + 40 p_K)}{p_L} \quad \text{e} \quad d_P = \frac{0.8 (85 p_L + 40 p_K)}{p_P}$$

No modelo, a demanda do consumidor é composta por d_P (consumo de produtos desejáveis) e d_L , que indica o trabalho retido na forma de lazer.

A produção do bem P é realizada através de três atividades lineares de coeficientes técnicos fixos do tipo de Leontief, indicadas por P1, P2 e P3, sendo y o vetor de níveis de produção dessas atividades, com $y_{P1}, y_{P2}, y_{P3} \geq 0$. Os coeficientes tecnológicos consistentes com produção unitária dessas atividades são:

	P1	P2	P3
L	- 0.5	- 0.8	- 0.8
K	- 0.5	- 0.2	- 0.5
P	1	1	1

Aplicando-se o método de modelagem apresentado nas seções anteriores à economia descrita, obtém-se as seguintes equações de oferta excedente e lucro negativo:

$$\left. \begin{aligned}
 ES_L &= 85 - 0.5 y_{P1} - 0.8 y_{P2} - 0.8 y_{P3} - d_L \\
 ES_K &= 40 - 0.5 y_{P1} - 0.2 y_{P2} - 0.5 y_{P3} \\
 ES_P &= y_{P1} + y_{P2} + y_{P3} - d_P \\
 NP_{P1} &= p_L 0.5 + p_K 0.5 - p_P \\
 NP_{P2} &= p_L 0.8 + p_K 0.2 - p_P \\
 NP_{P3} &= p_L 0.8 + p_K 0.5 - p_P
 \end{aligned} \right\} \quad (16)$$

Em preparação para as próximas seções, as quantidades demandadas d_L e d_P serão calculadas a seguir de outra forma, mais adequada aos desenvolvimentos que se seguirão, porém equivalente, como o produto de *coeficientes de demanda hicksiana a nível unitário de utilidade* por um *nível de atuação* da demanda, à semelhança de uma atividade de produção. O consumo deverá ser aproximado pela função CES, com uma elasticidade de substituição próxima de um. Essa aproximação é muito utilizada em economia tendo em vista que a forma funcional Cobb-Douglas é um caso particular da função CES quando a elasticidade de substituição σ tende a 1, ou seja,

$$\lim_{\sigma \rightarrow 1} (\alpha_L q_L^{1/\sigma} + \alpha_P q_P^{1/\sigma})^{\sigma/(\sigma-1)} = q_L^{\alpha_L} q_P^{\alpha_P}$$

Assim, os coeficientes unitários h_L e h_P serão dados por:

$$h_L = \left(\frac{(0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000}}{p_L} \right)^{0.999}$$

$$h_P = \left(\frac{(0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000}}{p_P} \right)^{0.999}$$

E as demandas d_L e d_P serão dadas por:

$$d_L = h_L u_C$$

$$d_P = h_P u_C$$

onde u_C é o nível de atuação da demanda, indicando o *consumo aparente* do consumidor, que por ser uma medida representativa da utilidade, será chamado de *nível de utilidade* do consumidor.

É necessário acrescentar, ainda, para cada consumidor, uma equação que condicione a demanda por bens desse consumidor à diferença entre seus gastos totais e sua renda. Chamando de NE_C essa diferença, tem-se:

$$NE_C = (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K)$$

A forma complementar dessa equação é:

$$NE_C(p) \geq 0 : u_C \geq 0$$

Introduzindo o novo formato da demanda e a equação acima em (16), tem-se:

$$\left. \begin{aligned} ES_L &= 85 - 0.5 y_{P1} - 0.8 y_{P2} - 0.8 y_{P3} - h_L u_C \\ ES_K &= 40 - 0.5 y_{P1} - 0.2 y_{P2} - 0.5 y_{P3} \\ ES_P &= y_{P1} + y_{P2} + y_{P3} - h_P u_C \\ NP_{P1} &= p_L 0.5 + p_K 0.5 - p_P \\ NP_{P2} &= p_L 0.8 + p_K 0.2 - p_P \\ NP_{P3} &= p_L 0.8 + p_K 0.5 - p_P \\ NE_C &= (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K) \end{aligned} \right\} \quad (17)$$

A resolução desse problema será vista a seguir.

3.2.1 Calculando a solução de equilíbrio

O sistema (17) será resolvido através dos procedimentos para o cálculo do equilíbrio do *Pegasus/Mathematica*⁵. Os comandos que direcionam o *Mathematica* para o diretório que contém o *Pegasus* e o carrega, disponibilizando seus procedimentos, são:

```
In[1]:=
  SetDirectory["\Pegasus"]
  Get["Pegasus.s3"];
Out[1]=
  C:\Pegasus
  Pegasus 1.2 (October 30, 1999)
```

Observe-se que os comandos de entrada aceitos são precedidos por *In[...]* e que os resultados (quando houverem) são precedidos por *Out[...]*.

Os dados da economia são informados a seguir – na produção as atividades com seus coeficientes, no consumo o consumidor C com suas dotações iniciais e demandas:

```
In[3]:=
  producao[1]={
    P1->{ L-> -0.5, K-> -0.5, P-> 1 },
    P2->{ L-> -0.8, K-> -0.2, P-> 1 },
    P3->{ L-> -0.8, K-> -0.5, P-> 1 }
  };
  consumo[1]={
    C->{
      { L-> 85, K-> 40 },
      { L->.2((.2 p[L]^0.001+.8 p[P]^0.001)^1000/p[L])^0.999,
        P->.8((.2 p[L]^0.001+.8 p[P]^0.001)^1000/p[P])^0.999 }
    }
  };
```

O procedimento `$FindEquilibrium` utiliza os dados de produção, dotações, consumo e numerário (o bem P nesse caso) para montar as equações do sistema (17) e resolvê-lo, utilizando o algoritmo SLCP:

⁵ Para isso será utilizada uma interface de comunicação entre o Word e o *Mathematica*, que envia os comandos ao *Mathematica* e após sua execução insere os resultados no texto do documento.


```

In[5]:=
modelo[1]=$FindEquilibrium[ $Production->producao[1],
                             $Consumption->consumo[1],
                             $Numeraire->P, $Report->None, $Solver->SLCP
                             ];
$Solution[modelo[1]]
Out[5]=
{p[L] -> 1., p[K] -> 1., p[P] -> 1, y[P1] -> 66.6667,
 y[P2] -> 33.3333, y[P3] -> 0., u[C] -> 125.}

```

Observe-se que no resultado obtido os preços dos bens, os níveis das atividades e o nível de utilidade do consumidor C são:

p_L	p_K	p_P	y_{P1}	y_{P2}	y_{P3}	u_C
1.	1.	1.	66.6667	33.3333	0.	125.

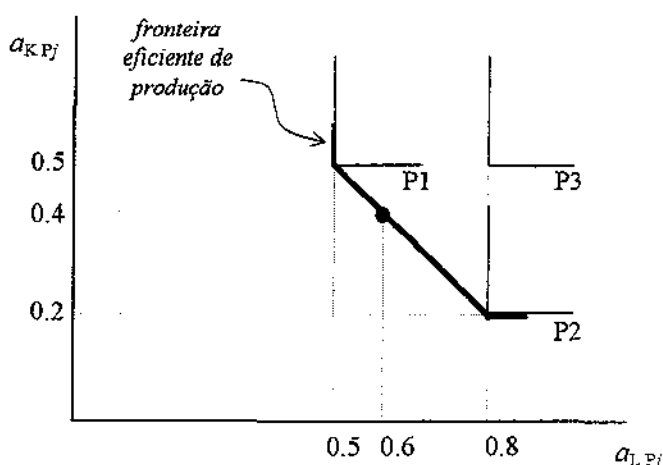
A seguir, as quantidades demandadas e produzidas pelas atividades P1, P2 e P3 no equilíbrio, onde observa-se que as quantidades de L, K e P (60, 40 e 100) são uma combinação convexa das atividades P1 e P2:

$a_{ij} y_j$	$a_{iP1} y_{P1}$	$a_{iP2} y_{P2}$	$a_{iP3} y_{P3}$	<i>total</i>
L	- 0.5x66.6667	- 0.8x33.3333	- 0.8x0.0	- 60
K	- 0.5x66.6667	- 0.2x33.3333	- 0.5x0.0	- 40
P	1x66.6667	1x33.3333	1x0.0	100

O problema combinatorial resolvido pelo algoritmo SLCP, nesse caso, pode ser ilustrado através do seguinte problema de programação linear:

$$\begin{aligned}
 \min_{y_{P1}, y_{P2}, y_{P3}} \quad & p_L (a_{LP1} y_{P1} + a_{LP2} y_{P2} + a_{LP3} y_{P3}) + p_K (a_{KP1} y_{P1} + a_{KP2} y_{P2} + a_{KP3} y_{P3}) \\
 \text{s. a.} \quad & y_{P1} + y_{P2} + y_{P3} = 100
 \end{aligned}$$

Para diversos valores de p_L e p_K , as quantidades usadas dos fatores L e K associadas às soluções do programa linear acima formam a *fronteira eficiente de produção*, mostrada no gráfico a seguir juntamente com as atividades P1, P2 e P3:



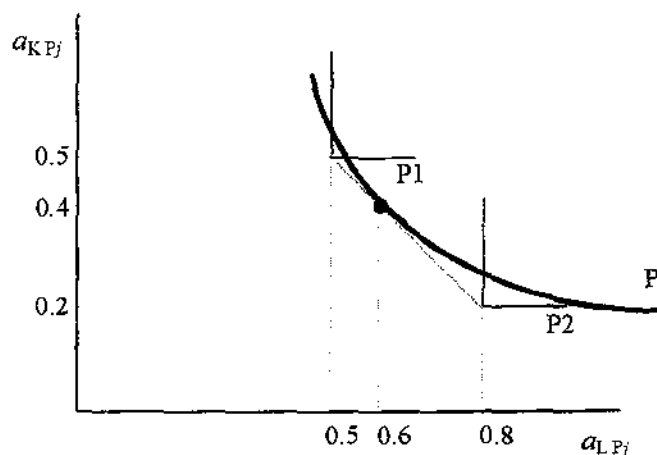
Notar que a atividade P3 jamais seria operada, por não estar na fronteira eficiente de produção. No equilíbrio econômico, o nível dessa atividade seria zero e o algoritmo ACI não seria capaz de resolver o modelo; eventualmente, o jacobiano do sistema linear associado torna-se singular:

```
In[7]:=
modelo[1]=$FindEquilibrium[ $Production->producao[1],
    $Consumption->consumo[1],
    $Numeraire->P, $Report->None, $Solver->ACI
];
$Solution[modelo[1]]
MA28: matrix is structurally singular, rank = 5
*** Error return from LU factorization routine
*** Matrix is not full rank
```

Se o modelo for construído apenas com as atividades P1 e P2 (das quais já se sabe que serão maiores que zero no equilíbrio), o algoritmo ACI encontrará a mesma solução encontrada pelo algoritmo SLCP para essas atividades:

```
In[9]:=
producao[2]={
    P1->{ L-> -0.5, K-> -0.5, P-> 1 },
    P2->{ L-> -0.8, K-> -0.2, P-> 1 }
};
modelo[2]=$FindEquilibrium[ $Production->producao[2],
    $Consumption->consumo[1],
    $Numeraire->P, $Report->None, $Solver->ACI
];
$Solution[modelo[2]]
Out[11]=
{p[L] -> 1., p[K] -> 1., p[P] -> 1, y[P1] -> 66.6667,
 y[P2] -> 33.3333, u[C] -> 125.}
```

Nos modelos em que o algoritmo ACI será utilizado, os coeficientes das atividades de produção serão obtidos por *calibragem* dos valores demandados e produzidos observados (40, 60 e 100, respectivamente para L, K e P), como será visto na próxima seção. Nesse caso, todas as atividades que produzem um determinado bem são substituídas por uma única função de produção P, e seu nível será sempre maior que zero:



Em resumo, se por construção do modelo econômico é possível garantir que a solução será estritamente positiva, o problema de complementaridade não-linear se reduz a um sistema de equações não-lineares, e o algoritmo ACI pode ser utilizado para resolvê-lo.

3.2.2 Calibrando o modelo com a técnica de geração de colunas

Na seção anterior, o modelo da economia foi construído a partir de informações conhecidas dos coeficientes tecnológicos de produção e das preferências dos consumidores. Entretanto, na prática, é possível apenas observar os valores finais dos fluxos de renda das transações efetuadas pelos diversos agentes econômicos, produtores e consumidores. A observação direta dos coeficientes técnicos das diferentes atividades envolvidas na produção seria impraticável, pois exigiria informações detalhadas dos processos produtivos internos. Tipicamente o economista aproxima toda complexidade eventualmente existente na produção por uma função de produção ajustada aos valores observados usando também

um parâmetro de curvatura, a elasticidade de substituição, estimada econometricamente ou simplesmente copiada de estudos similares. Admita-se então que o modelador dispõe de um banco de dados da economia apresentada anteriormente na forma de uma Matriz de Contabilidade Social (MCS – ou SAM, de *Social Account Matrix*, como preferem alguns autores):

		P	L	K	C	Total
<i>Atividades</i>	P - produto				100	100
<i>Fatores</i>	L - trabalho	60			25	85
	K - capital	40				40
<i>Instituições</i>	C - domicílios		85	40		125
<i>Total</i>		100	85	40	125	

Uma MCS é uma descrição de uma economia onde são tabulados os fluxos de renda de todos os agentes considerados dessa economia, sendo que a cada agente são associados uma linha e uma coluna – as receitas são colocadas na linha e as despesas na coluna. Do pressuposto que despesas devem ser iguais às receitas tem-se que os totais da linha e coluna de um agente determinado devem ser iguais, o que deve ocorrer para todos os agentes.

Um modelo CEG de uma dada economia deve ser *calibrado* a partir de um banco de dados como a MCS apresentada. O modelo geralmente é especificado em termos de variáveis dependentes e independentes (endógenas e exógenas) e parâmetros. Alguns parâmetros, como por exemplo as elasticidades de substituição dos fatores, são via de regra obtidos da literatura, como mencionado acima. Outros parâmetros devem ser ajustados de maneira a garantir a aderência dos valores calculados pelo modelo aos valores observados contidos no banco de dados. Este ajuste, necessário para a existência de uma solução de equilíbrio para o modelo, é conhecido como "calibragem do modelo".

Na produção, partir dos valores observados, é possível aplicar a técnica de geração de colunas para calibrar e obter os coeficientes $a_i(\mathbf{p})$ que compõem o vetor coluna que irá

representar a atividade P. Os coeficientes $a_i(\mathbf{p})$ são calibrados (ajustados) pelo *Pegasus* de forma a minimizar os custos de produção de uma forma funcional CES (Elasticidade Constante de Substituição) de produção através do procedimento de geração de colunas `$CES`, como descrito em Scaramucci [56].

Considerando-se que em uma economia competitiva em equilíbrio os preços relativos p_i dos bens sejam iguais à unidade e admitindo-se que 1,3 seja um valor típico para a elasticidade de substituição entre trabalho e capital para regiões em desenvolvimento, para utilização do procedimento `$CES` informa-se os nomes dos bens, os pares $\{p_i, q_i\}$ de preços e quantidades de cada bem, e a elasticidade de substituição:

```
In{12}:=
  $CES[ ({ L, K }->P), { {1,60}, {1,40} }, {1,100}, {1.3} ]
Out{12}=
      1                               1.3
(L -> -0.6 (-----)
           0.4      0.6      3.33333
           (----- + -----)
           0.3      0.3
           p[K]      p[L]
           p[L]

      1                               1.3
(K -> -0.4 (-----)
           0.4      0.6      3.33333
           p[K] (----- + -----)
           0.3      0.3
           p[K]      p[L]

P -> 1)
```

Para o consumo, na seção anterior, os coeficientes unitários de demanda foram aproximados pela forma funcional CES a partir das preferências do consumidor, e também nesse caso o procedimento `$CES` pode ser utilizado para calibrar os coeficientes da demanda a partir das quantidades demandadas observadas.

Os coeficientes $h_i(\mathbf{p})$ são calibrados pelo *Pegasus* de forma a maximizar a utilidade do consumidor através do procedimento de geração de colunas `$CES`. Devem ser informados os nomes dos bens, os pares $\{p_i, q_i\}$ de preços e quantidades de cada bem, e a elasticidade de substituição próxima de 1:

```

In[13]:=
$CES[ { L, P }, { {1,25}, {1,100} }, {0.999} ]
Out[13]=
(L -> 0.2 (-----),
          0.001      0.001 1000.
          (0.2 p[L]  + 0.8 p[P] )  0.999
          -----)
          p[L]
(P -> 0.8 (-----) ]
          0.001      0.001 1000.
          (0.2 p[L]  + 0.8 p[P] )  0.999
          -----)
          p[P]

```

Os coeficientes de produção e demanda obtidos acima são:

$$\begin{aligned}
 a_L &= -0.6 \left(\frac{1}{\left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{p_L^{0.3}} \right)^{10/3} p_L} \right)^{1.3} \\
 a_K &= -0.4 \left(\frac{1}{\left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{p_L^{0.3}} \right)^{10/3} p_K} \right)^{1.3} \\
 h_L &= 0.2 \left(\frac{(0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000.}}{p_L} \right)^{0.999} \\
 h_P &= 0.8 \left(\frac{(0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000.}}{p_P} \right)^{0.999}
 \end{aligned}$$

E substituindo-se esses coeficientes no sistema (17), obtém-se:

$$\left. \begin{aligned}
 ES_L &= 85 - a_L y_P - h_L u_C \\
 ES_K &= 40 - a_K y_P \\
 ES_P &= y_P - h_P u_C \\
 NP_P &= p_L a_L + p_K a_K - p_P \\
 NE_C &= (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K)
 \end{aligned} \right\} \quad (18)$$

Para obter diretamente no modelo os novos coeficientes, o procedimento \$CES será usado nas definições da produção e do consumo:

```

In[15]:=
producao[3]={
  P->$CES[ {{L, K}->P}, {{1,60}, {1,40}}, {1,100}, {1.3} ]
};
consumo[2]={
  C->{
    { L-> 85, K-> 40 },
    $CES[ { L, P }, { {1,25}, {1,100} }, {0.999} ]
  }
};

```

Assim, no procedimento de cálculo do equilíbrio as informações de $\$Production$ e $\$Consumption$ serão alteradas, mantendo-se o mesmo numerário:

```
In{17}:=
  modelo[3]=$FindEquilibrium[ $Production->producao[3],
                             $Consumption->consumo[2],
                             $Numeraire->P, $Report->None, $Solver->ACI
                             ];
  $Solution[modelo[3]]
Out{18}=
  {p[L] -> 1., p[K] -> 1., p[P] -> 1, y[P] -> 100., u[C] -> 125.}
```

E nesse caso a solução de equilíbrio obtida acima é:

p_L	p_K	p_P	y_P	u_C
1.	1.	1.	100.	125.

Observe-se na solução agora obtida que os resultados para preços e nível de utilidade do consumidor são iguais aos encontrados anteriormente, e a quantidade produzida do bem P pela atividade P é igual à soma do resultado das atividades P1 e P2. A substituição dos coeficientes fixos de produção pelos $a_i(\mathbf{p})$ é importante também para representar tecnologias de produção que se ajustem às mudanças nos preços, possibilitando a construção de modelos mais próximos da realidade econômica.

3.2.3 Fazendo o escalamento do modelo

Tanto o algoritmo SLCP como o ACI se baseiam na linearização do sistema de equações, e a fatoração da matriz jacobiana é fundamental na resolução do sistema linearizado (ver apêndices A3–A5).

Em Duff, Erisman e Reid [13] é feita uma análise aprofundada dos erros introduzidos ao obter a fatoração de uma matriz, onde é ressaltado que a minimização desses erros depende fundamentalmente de a matriz ser bem escalada (*well-scaled*), no sentido de que os elementos que a compõem não apresentem uma variação muito grande de valores. Afirmando ser surpreendentemente difícil corrigir automaticamente matrizes mal escaladas

analisando diretamente os dados (por ser geralmente impossível distinguir o que é significativo e o que não é). Escolher bem as unidades de medida do modelo, por exemplo, pode ser uma boa maneira de se obter uma matriz jacobiana bem escalada. Da mesma maneira, em Dervis et al. [14]: *"It is a common observation in numerical analysis that derivatives are tricky – much harder, for example, than integrals – and our experience probably has much to do with how we specified the functions to be solved"*.

Aplicando-se o método definido nas seções anteriores em modelos reais de comércio internacional, os preços relativos apresentariam uma pequena variação em torno de 1; no entanto, como os valores de fluxo de renda são da ordem de 10^{12} , os níveis de operação das atividades também seriam dessa grandeza, e essa grande variação torna difícil ou impossível a fatoração do jacobiano num modelo de grande porte. No exemplo de economia aqui apresentado essa variação de valores não causa problemas por ser menor (da ordem de 10^2), mas permite ilustrar a solução proposta para essa dificuldade.

De uma maneira geral, essa variação decorre das unidades de medida escolhidas para definir as atividades de produção, como no caso da atividade P do exemplo: o nível de operação é o valor "observado" ($y_P=100$) e seus coeficientes de insumo-produto são unitários. Entretanto, o nível de operação de uma atividade e seus coeficientes de insumo-produto podem ser arbitrariamente escolhidos desde que o *produto* entre coeficientes e nível não seja alterado.

Assim, essa relação de unidades de medida pode ser invertida: arbitrando-se ao nível de operação da atividade P o valor unitário ($y_P = 1$), os coeficientes de insumo-produto serão dados pelas quantidades "observadas" q_i , no caso de coeficientes fixos:

	$a_i = q_i$
L	-60
K	-40
P	100

Para se obter essa inversão no caso de coeficientes dependentes de preços $a_i(\mathbf{p})$, acrescentou-se ao procedimento `$CES` a opção `$OutputLevel` com esse objetivo, sendo os coeficientes obtidos automaticamente com os valores observados:

```
In[19]:*
$CES[ {{ L, K }->P},
      { {1,60}, {1,40} }, {1,100}, {1.3},
      $OutputLevel->100 ]
Out[19]=
      1                                     1.3
L -> -60. (-----) ,
           0.4      0.6      3.33333
           (----- + -----)      p[L]
           0.3      0.3
           p[K]      p[L]

      1                                     1.3
K -> -40. (-----) ,
           0.4      0.6      3.33333
           p[K] (----- + -----)
           0.3      0.3
           p[K]      p[L]

P -> 100)
```

Note-se que substituindo-se os preços de equilíbrio ($p_L = 1$ e $p_K = 1$) no resultado obtém-se os valores "observados" propostos: $a_L = -60$, $a_K = -40$ e $a_P = 100$ para os $a_i(\mathbf{p})$.

O problema de unidades de medida inadequadas também ocorre no consumo, e uma inversão semelhante de unidades de medida deve ser realizada. A opção `$OutputLevel` também deve ser aplicada no consumo, e assim, o procedimento `$CES` obtém os coeficientes $h_i(\mathbf{p})$ com o valor das quantidades demandadas e o nível de utilidade (o consumo aparente) passa a ser unitário:

```
In[20]:=
$CES[ { L, P }, { {1,25}, {1,100} }, {0.999}, $OutputLevel->125 ]
```

```
Out[20]=
      0.001      0.001 1000.
(0.2 p[L]      + 0.8 p[P]      ) 0.999
{L -> 25. (-----) ,
           p[L]
      0.001      0.001 1000.
(0.2 p[L]      + 0.8 p[P]      ) 0.999
P -> 100. (-----) }
```

Dessa maneira, fazendo-se:

$$a_L = -60. \left(\frac{1}{\left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{p_L^{0.3}} \right)^{10/3} p_L} \right)^{1.3}$$

$$a_K = -40. \left(\frac{1}{\left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{p_L^{0.3}} \right)^{10/3} p_K} \right)^{1.3}$$

$$h_L = 25. \left(\frac{\left((0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000.} \right)^{0.999}}{p_L} \right)$$

$$h_P = 100. \left(\frac{\left((0.2 p_L^{0.001} + 0.8 p_P^{0.001})^{1000.} \right)^{0.999}}{p_P} \right)$$

E substituindo-se esses coeficientes no sistema (18), obtém-se:

$$\left. \begin{aligned} ES_L &= 85 - a_L y_P - h_L u_C \\ ES_K &= 40 - a_K y_P \\ ES_P &= 100 y_P - h_P u_C \\ NP_P &= p_L a_L + p_K a_K - p_P 100 \\ NE_C &= (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K) \end{aligned} \right\} \quad (19)$$

Para se obter esses valores diretamente no modelo, as definições da produção e do consumo do exemplo deverão ser alteradas usando procedimentos \$CES de geração de coluna com a nova opção \$OutputLevel, de maneira a obter automaticamente os coeficientes com os valores observados:

```
In[21]:=
producao[4]={ P->$CES[ {{ L, K }->P},
```

```

        { {1,60}, {1,40} }, {1,100}, {1.3},
        $OutputLevel->100 ]
    };
consumo[3]={ C->{
    { L-> 85, K-> 40 },
    $CES[ {L, P}, {{1,25}, {1,100}}, {0.999},
    $OutputLevel->125 ]
    }
};

```

Utilizando-se a produção e consumo assim definidos, tem-se:

```

In[23]:=
modelo[4]=$FindEquilibrium[ $Production->producao[4],
    $Consumption->consumo[3],
    $Numeraire->P, $Report->None, $Solver->ACI
];
$Solution[modelo[4]]
Out[24]=
{p[L] -> 1., p[K] -> 1., p[P] -> 1, y[P] -> 1., u[C] -> 1.}

```

A solução de equilíbrio obtida acima é:

p_L	p_K	p_P	y_P	u_C
1.	1.	1.	1.	1.

Observe-se que na solução obtida com a inversão das unidades de medida os níveis das atividades e níveis de utilidade – que num modelo de grande porte poderiam ser até da ordem de 10^{12} sem a inversão –, bem como preços, têm agora um intervalo de definição em torno da unidade, tornando possível uma matriz jacobiana bem escalada.

A seguir será apresentado o jacobiano de um modelo onde os dados originais do exemplo resolvido acima foram multiplicados por 10^3 para ressaltar as diferenças, inicialmente sem e depois com escalamento dos dados:

	p_L	p_K	y_P	u_C
p_L	19980,29	-0,312	-0,6	-0,2
p_K	-0,312	0,312	-0,4	
y_P	0,6	0,4		
u_C	-60000	-40000		1,0

(observe-se acima que a linha e coluna referentes ao numerário foram eliminadas).

Jacobiano com escalamento dos dados do modelo				
	p_L	p_K	y_P	u_C
p_L	51180	-31200	-60000	-25000
p_K	-31200	31200	-40000	
y_P	60000	40000		
u_C	-60000	-40000		125000

A experiência computacional com modelos de grande porte, representando economias reais, tem demonstrado que o método de escalamento do modelo econômico, como aplicado ao exemplo, é indispensável para que a fatoração da matriz jacobiana seja obtida com êxito. No apêndice A6.7 é apresentada uma tabela de distribuição de frequência dos elementos do jacobiano de um modelo desse tipo que será apresentado no próximo capítulo, onde se pode observar o efeito do escalamento dos dados, semelhante ao do exemplo acima.

3.2.4 Simulação de cenários alternativos

Modelos computáveis de equilíbrio geral são usados tipicamente para a avaliação de políticas econômicas. A análise de simulação de cenários permite avaliar as consequências da alteração de alguma característica da economia sob estudo.

O modelo é inicialmente calibrado para reproduzir os valores das variáveis econômicas – fluxos de renda, por exemplo – nele representadas, como visto na seção anterior.

A seguir, é dado um "choque" à economia, que corresponde a alterar o valor de alguma variável exógena ou impor uma determinada condição a ser satisfeita pelas variáveis endógenas do modelo. Por exemplo, em um modelo de comércio internacional, algum país pode decidir aumentar as tarifas de importação ou, ainda, estabelecer uma cota de importação para um bem produzido em outra região.

O novo equilíbrio é então computado e as implicações das alterações introduzidas podem ser quantificadas. Assim, a simulação de cenários em modelos CEG corresponde grosso modo à análise de sensibilidade em matemática ou estática comparativa em economia.

Considere-se, por exemplo, que no último modelo a dotação inicial de trabalho diminua de 85 para 80. O comando a seguir redefine o consumo, alterando apenas a dotação do bem L para 80:

```
In[25]:=
consumo[4]= { C->{
    { L-> 80, K-> 40 },
    $CES[ {L, P}, {{1,25}, {1,100}}, {0.999},
    $OutputLevel->125 ]
    }
};
```

Mantendo o numerário e a produção do último modelo, e obtendo-se a nova solução de equilíbrio, tem-se:

```
In[27]:=
modelo[5]=$FindEquilibrium[ $Production->producao[4],
    $Consumption->consumo[4],
    $Numeraire->P, $Report->None, $solver->ACI
];
$Solution[modelo[5]]
Out[28]=
{p[L] -> 1.01945, p[K] -> 0.971721, p[P] -> 1, y[P] -> 0.963394,
u[C] -> 0.959694}
```

A seguir, as soluções de equilíbrio antes e após do choque:

	p_L	p_K	p_P	y_P	u_C
Base	1.	1.	1.	1.	1.
Choque	1.01945	0.971721	1.	0.963394	0.959694

Observe-se que devido à escassez do trabalho, os níveis de produção de P e da utilidade do consumidor diminuem, ao mesmo tempo que o preço do trabalho sobe; por outro lado, um excesso relativo de oferta de capital faz com que seu preço diminua.

3.2.5 Acrescentando restrições institucionais

Ao se modelar uma economia, muitas vezes é necessário restringir um ou mais componentes do modelo econômico, como por exemplo impor limites a preços, estabelecer quotas, atingir metas de importação ou exportação, etc.

Suponha-se que o modelo CEG admite apenas uma solução. A introdução de uma restrição adicional, não acompanhada de outras providências, em geral, torna o modelo sem solução. Porém, se a restrição for introduzida na forma de uma condição de complementaridade, a variável complementar associada – chamada de *multiplicador* em analogia às variáveis duais da programação matemática – pode ser usada para distorcer a economia, fazendo-a acomodar a restrição adicional. Essa formulação depende essencialmente da capacidade do modelador de entender o fenômeno econômico envolvido. Por exemplo, para uma restrição fixando um valor mínimo para a taxa salarial, o multiplicador associado poderia ser a taxa de desemprego, que corresponde à fração das dotações de trabalho que deve ser eliminada da economia. Alternativamente, o multiplicador poderia ser usado para distorcer a economia na forma de um subsídio a ser concedido à mão-de-obra usada na produção. Essa idéia será exemplificada a seguir.

Assim, considere-se que no modelo [4], definido na penúltima seção (3.2.3), se deseje que o preço do trabalho (L) na atividade de produção (P) seja ao menos 10% maior que o preço do bem P:

$$p_L \geq 1.1 p_P$$

Ou seja:

$$p_L - 1.1 p_P \geq 0$$

Fazendo-se:

$$NR = p_L - 1.1 p_P,$$

a restrição pode ser introduzida como a condição de complementaridade

$$NR \geq 0 : m_{NR} \geq 0$$

onde m_{NR} é a potência da taxa de subsídio à mão-de-obra usada na produção.

Para distorcer a economia, todas as ocorrências da variável p_L (taxa salarial) existentes na produção devem ser multiplicadas por m_{NR} . No comando *Mathematica* a seguir, isso é feito através da regra $\{p[L] \rightarrow m[NR] p[L]\}$ aplicada ao resultado do procedimento $\$CES$:

```
In[29]:=
SCES[ ({ L, K }->P),{ {1,60}, {1,40} },
      {1,100},{1.3}, $OutputLevel->100 ] /. {p[L] -> m[NR] p[L]}
Out[29]=
{L -> -60. (-----) 1.3
          0.4      0.6      3.33333
m[NR] p[L] (----- + -----)
          0.3      0.3
          p[K]      (m[NR] p[L])
          1
K -> -40. (-----) 1.3
          0.4      0.6      3.33333
p[K] (----- + -----)
          0.3      0.3
          p[K]      (m[NR] p[L])
P -> 100}
```

Portanto, os coeficientes condicionados da atividade P são:

$$a_L = -60. \left(\frac{1}{m_{NR} p_L \left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{(m_{NR} p_L)^{0.3}} \right)^{10/5}} \right)^{1.3}$$

$$a_K = -40. \left(\frac{1}{p_K \left(\frac{0.4}{p_K^{0.3}} + \frac{0.6}{(m_{NR} p_L)^{0.3}} \right)^{10/5}} \right)^{1.3}$$

Deve ser notado que essa alteração na produção, por sua vez, causará uma alteração à renda do consumidor. O valor da alteração à renda r_{NR} é o produto do coeficiente a_L (um

valor negativo), do preço p_L , do nível de operação da atividade y_P e da taxa de subsídio ($m_{NR} - 1$):

$$r_{NR} = (-a_L) p_L y_P (m_{NR} - 1)$$

Notar que no equilíbrio m_{NR} será menor que um e, assim, $r_{NR} < 0$ corresponde a diminuição de renda sofrida pelo consumidor.

Com os coeficientes distorcidos na forma mostrada acima, introduzindo a alteração r_{NR} na renda do consumidor e acrescentando-se, também, a restrição como uma condição de complementaridade no sistema (19), tem-se:

$$\left. \begin{aligned} ES_L &= 85 - a_L y_P - h_L u_C \\ ES_K &= 40 - a_K y_P \\ ES_P &= 100 y_P - h_P u_C \\ NP_P &= m_{NR} p_L a_L + p_K a_K - p_P 100 \\ NR &= p_L - 1.1 p_P \\ NE_C &= (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K + r_{NR}) \end{aligned} \right\} \quad (20)$$

No *Pegasus*, a distorção da produção pela potência da taxa de subsídio m_{NR} e a atribuição da alteração à renda decorrente são feitas automaticamente utilizando-se a construção a seguir, que informa a regra de distorção e o consumidor cuja renda deverá ser alterada:

$$\{ \{ p_L \rightarrow m_{NR} p_L \} \rightarrow C \}$$

Observem-se as definições da atividade de produção P distorcida por m_{NR} e da restrição NR, a seguir:

```
in{30}:=
  producao[5]={ P-> {
    $CES[ {{ L, K }->P},
    { {1,60}, {1,40} }, {1,100},{1.3},
    $outputLevel->100 ],
    {{p[L] -> m[NR] p[L]}->C}
  }
};
restricao[1]={ NR -> p[L] - 1.1 p[P]};
```


Utilizando-se o consumo definido anteriormente, a nova produção e introduzindo a restrição com a opção `$Constraints`, tem-se:

```
In[32]:=
modelo[6]=$FindEquilibrium[
    $Production->producao[5],
    $Consumption->consumo[3],
    $Constraints->restricao[1],
    $Numeraire->P,$Report->None, $Solver->ACI
];
$Solution[modelo[6]]
Out[33]=
{p[L] -> 1.1, p[K] -> 1.01415, p[P] -> 1, y[P] -> 1.01843,
 m[NR] -> 0.900646, u[C] -> 0.999219}
```

A solução de equilíbrio obtida é:

p_L	p_K	p_P	y_P	m_{NR}	u_C
1.1	1.01415	1.	1.01843	0.900646	0.999219

Assim, $(1 - m_{NR}) * 100 = 9.93\%$ é o subsídio necessário para manter o preço do trabalho no mínimo 10% maior que o preço do bem P. Observa-se nesse resultado que o nível de utilidade do consumidor quase não é alterado (99,92% do caso base), e que o preço do capital e o nível de produção crescem pouco mais de 1%.

3.2.6 Modelando impostos e subsídios

Impostos e subsídios são tão comuns em economias reais a ponto de as próprias bases de dados que as caracterizam se apresentarem já distorcidas por eles. Assim sendo, é importante saber como tratá-los no exercício de modelagem.

Impostos ou subsídios podem ser considerados como um diferencial sobre o preço de mercado de um bem que um agente da economia paga ou recebe ao comercializar o bem. Considere-se o conceito de *preço de agente* de um bem como sendo o preço de mercado do bem mais impostos (ou menos subsídios):

$$preço_{agente} = preço_{mercado} \pm diferencial$$

O formato de \pm *diferencial* para impostos ou subsídios pode ser unificado em um fator multiplicativo que represente os dois casos. Para isso, seja t o quociente entre os preços de agente e de mercado de um bem, obtido dividindo-se o preço de agente pelo preço de mercado:

$$t = \frac{\text{preço}_{\text{agente}}}{\text{preço}_{\text{mercado}}}$$

Observe-se que dessa forma se o valor de t for maior que um corresponderá a um imposto e se for menor que um corresponderá a um subsídio. Assim, o preço de agente do bem poderá ser obtido multiplicando-se o preço de mercado pelo fator t :

$$\text{preço}_{\text{agente}} = t \text{ preço}_{\text{mercado}}$$

Considere-se a existência de um imposto de 20% sobre o trabalho na atividade de produção em uma economia semelhante à do exemplo em estudo. Supõe-se a existência de um governo que recebe os impostos do setor produtivo P e transfere as rendas obtidas para o consumidor C (neste exemplo singelo, não poderia ser de outra maneira). De antemão pode-se argüir que, se por um lado o imposto aumenta o preço do produto, por outro lado o consumidor terá uma renda adicional equivalente, e o efeito observável serão os valores nominais aumentados dos fluxos da economia.

De fato, na Matriz de Contabilidade Social dessa economia é possível observar a influência do imposto nos valores dos fluxos em relação à MCS anterior:

		P	L	K	C	G	Total
<i>Atividades</i>	P - produto				112		112
<i>Fatores</i>	L - trabalho	60			25		85
	K - capital	40					40
<i>Instituições</i>	C - domicílios		85	40		12	137
	G - governo	12					12
<i>Total</i>		112	85	40	137	12	

Note-se o valor 12 pago pelo produtor ao governo na forma de imposto sobre o trabalho usado. O valor total pago pelo fator trabalho é, assim, 72. Entretanto, o valor efetivamente recebido do produtor pelos trabalhadores é 60. Portanto, 60 é o valor do trabalho na produção a *preço de mercado* e 72 é o valor do trabalho na produção a *preço de agente*. Em consonância com a hipótese de retornos constantes à escala o valor do produto deve também aumentar de 12, para 112. Com isso, o valor do produto no consumo também sobe para 112 e o consumo aparente passa a 137. Como o imposto arrecadado é repassado ao consumidor C via governo, esse aumento de sua renda possibilita o consumo do produto mais caro.

O valor de t_L pode ser calculado a partir dos valores "observados" a preços de mercado e agente do trabalho:

$$t_L = \frac{\text{preço}_{\text{agente}}}{\text{preço}_{\text{mercado}}} = \frac{\frac{\text{valor}_{\text{agente}}}{q_L}}{\frac{\text{valor}_{\text{mercado}}}{q_L}} = \frac{\text{valor}_{\text{agente}}}{\text{valor}_{\text{mercado}}} = \frac{72}{60} = 1,2$$

Como anteriormente, os coeficientes a_L , a_K , h_L , e h_P serão obtidos através do procedimento SCES, observando-se que na definição do par $\{p_i, q_i\}$ deve ser utilizado o *preço de agente* para p_i , que é o preço de mercado corrigido pelo imposto. Considerando-se que os preços de mercado são unitários no equilíbrio, verifica-se facilmente que:

$$\{p_i, q_i\} \rightarrow \left\{ \frac{\text{valor}_{\text{agente}}}{\text{valor}_{\text{mercado}}}, \text{valor}_{\text{mercado}} \right\}$$

Além disso, todas ocorrências de p_L devem ser alteradas para $t_L p_L$, realizado através da regra $\{p[L] \rightarrow 72/60 p[L]\}$ na expressão a seguir. Assim, na produção, tem-se:

```

In[34]:=
$CES[ {{ L, K }->P}, { {72/60,60}, {1,40} }, {1,112},{1.3},
      $OutputLevel->112 ] /. {p[L]->72/60 p[L]}
Out[34]=
{L -> -60. (-----) 1.3
          (----- + -----) p[L]
          0.3          0.3
          p[K]        p[L]
          1
          -----) 1.3
          0.357143 0.642857 3.33333
          p[K] (----- + -----)
          0.3          0.3
          p[K]        p[L]
          P -> 112)

```

Como o imposto não é aplicado no consumo, nesse caso p_L não deve ser corrigido, mas os valores nominais devem ser utilizados:

```

In[35]:=
$CES[ { L, P }, { {1,25}, {1,112} }, {0.999},
      $OutputLevel->137 ]
Out[35]=
{L -> 25. (-----) 0.999
          (0.182482 p[L]0.001 + 0.817518 p[P]0.001 1000.)
          -----)
          p[L]
          P -> 112. (-----) 0.999
          (0.182482 p[L]0.001 + 0.817518 p[P]0.001 1000.)
          -----)
          p[P]

```

Observe-se que acima são mostrados valores aproximados; internamente a precisão do *Mathematica* é muito maior. Com essa ressalva, os coeficientes são:

$$\begin{aligned}
 a_L &= -60. \left(\frac{1}{\left(\frac{0.357143}{p_K^{0.3}} + \frac{0.642857}{p_L^{0.3}} \right)^{10/3}} p_L \right)^{1.3} \\
 a_K &= -40 \left(\frac{1}{\left(\frac{0.357143}{p_K^{0.3}} + \frac{0.642857}{p_L^{0.3}} \right)^{10/3}} p_K \right)^{1.3} \\
 h_L &= 25. \left(\frac{\left((0.182482 p_L^{0.001} + 0.817518 p_P^{0.001})^{1000.} \right)^{0.999}}{p_L} \right) \\
 h_P &= 112. \left(\frac{\left((0.182482 p_L^{0.001} + 0.817518 p_P^{0.001})^{1000.} \right)^{0.999}}{p_P} \right)
 \end{aligned}$$

O valor v_{t_L} arrecadado com o imposto na produção (que, na prática, será repassado diretamente ao consumidor, prescindindo-se das linhas e colunas sombreadas nas tabelas) é uma função do coeficiente a_L (um valor negativo), do preço p_L , do nível de operação da atividade y_P e do imposto t_L :

$$v_{t_L} = (-a_L) p_L y_P (t_L - 1)$$

Utilizando-se no sistema (19) os coeficientes obtidos acima, onde as ocorrências do preço p_L são multiplicadas pelo fator t_L , e acrescentando-se à renda do consumidor, na última equação, o valor correspondente ao imposto, tem-se

$$\left. \begin{aligned} ES_L &= 85 - a_L y_P - h_L u_C \\ ES_K &= 40 - a_K y_P \\ ES_P &= 100 y_P - h_P u_C \\ NP_P &= p_L a_L + p_K a_K - p_P 100 \\ NE_C &= (h_L p_L + h_P p_P) u_C - (85 p_L + 40 p_K + v_{t_L}) \end{aligned} \right\} \quad (21)$$

Como visto anteriormente num caso semelhante, para informar ao *Pegasus* a existência do imposto sobre o bem L na atividade P de modo que automaticamente todas as ocorrências de p_L sejam substituídas pelo preço corrigido pelo fator t_L e que o valor correspondente ao imposto seja calculado e acrescentado à renda do consumidor C, utiliza-se a construção $\{ \{ p_L \rightarrow t_L p_L \} \rightarrow C \}$, que deve ser posicionada após a função $\$CES$ na definição da atividade. Observe-se a utilização dessa construção e dos valores a preço de agente e mercado na definição da produção do exemplo, a seguir:

```
In[36]:=
  producao[6]={ P->{
    $CES[ {{ L, K }->P},
          { {72/60,60}, {1,40} }, (1,112),(1.3),
          $outputLevel->112 ],
    {{p[L]->72/60 p[L]}->C}
  }
};
```

Na definição do consumo devem ser utilizados os valores nominais:

```
In[37]:=
consumo[5]={ C->{
  { L-> 85, K-> 40 },
  $CES[ { L, P },
    { {1,25}, {1,112} }, {0.999},
    $OutputLevel->137 ]
  }
};
```

Utilizando-se a produção e consumo assim definidos com `$FindEquilibrium`:

```
In[38]:=
modelo[7]=$FindEquilibrium[
  $Production->producao[6],
  $Consumption->consumo[5],
  $Numeraire->P,$Report->None, $Solver->ACI
];
$Solution[modelo[7]]
Out[39]=
{p[L] -> 1., p[K] -> 1., p[P] -> 1, y[P] -> 1., u[C] -> 1.}
```

A solução de equilíbrio obtida é:

p_L	p_K	p_P	y_P	u_C
1.	1.	1.	1.	1.

O modelo acima não pode ser comparado com o anterior. Afinal, a introdução de um imposto distorce a economia, levando à perda de bem estar. Os preços e níveis de atividade e utilidade têm valor unitário por construção, compondo um novo caso base.

3.3 Os dados: descrição do banco de dados do GTAP

Segundo Thomas Hertel, seu fundador e diretor, o Global Trade Analysis Project (Projeto Análise de Comércio Internacional) foi estabelecido em 1992 com o objetivo de diminuir o "custo de entrada" para os que buscam conduzir análises quantitativas de questões econômicas globais, com uma metodologia econômica abrangente, consistindo o projeto de um banco de dados globais, um método padrão de modelagem, programas de computador para manipular os dados e implementar o modelo padrão e uma rede de

pesquisadores, interligados através da Internet⁶, com um interesse comum em questões de análise de comércio e recursos multi-regionais

A apresentação a seguir é feita baseada em Hertel [34], e tem como objetivo mostrar didaticamente as *agregações* dos componentes do modelo e depois *identificar* as informações do banco de dados associadas às agregações, omitindo-se a análise aprofundada da fundamentação teórica do modelo, que deve ser apreciada no original.

3.3.1 O modelo padrão do GTAP

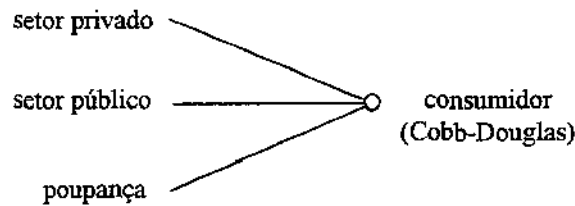
O modelo GTAP básico é um modelo de equilíbrio econômico computável multi-regional estático e visa representar a produção, comércio, investimento e consumo de bens a nível global com o objetivo de descrever a economia mundial e possibilitar simulações dos efeitos de políticas econômicas, principalmente nas áreas de comércio internacional e recursos regionais. Suas características mais marcantes são o tratamento do comportamento do consumo privado, as atividades de comércio e transporte internacionais, e o relacionamento entre poupança global e investimento.

Como em todo modelo, o que se tem é uma representação simplificada da realidade, sendo as nações e setores produtivos agrupados segundo critérios economicamente coerentes. Na versão 4 do banco de dados⁷ as nações são agrupadas em 45 regiões representativas e os bens agrupados em 50 setores representativos, descritos no apêndice A6. No texto a seguir será considerada a existência de m regiões e n setores representativos, i e j índices de bens, r e s índices de regiões e o índice k para os fatores primários (terra, trabalho e capital).

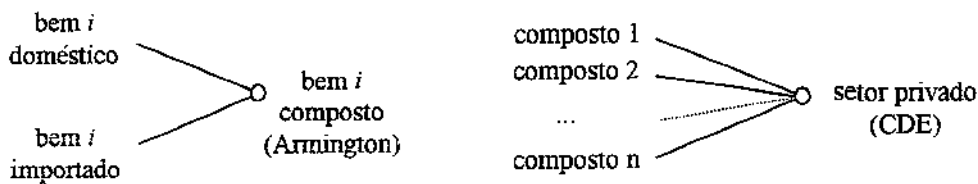
⁶ O endereço da página do GTAP na Internet é www.agecon.purdue.edu/gtap

⁷ Até a apresentação deste trabalho a versão 5 ainda não estava disponível.

Em cada região supõe-se a existência de um consumidor representativo cujos gastos são distribuídos entre setor privado, setor público e poupança através de uma função utilidade agregada Cobb-Douglas, sendo que as receitas advêm de impostos e da venda de dotações de fatores primários às firmas produtoras. Como evidenciado no texto original, essa formulação tem algumas vantagens e outras tantas desvantagens, sendo a principal destas a falha em subordinar os gastos do governo às receitas (Nessa formulação, como um corte em impostos leva a uma redução nos preços, a receita regional real cresce e os gastos reais do setor governamental acompanharão esse crescimento). A maior vantagem é dispor do indicador inequívoco de bem-estar fornecido pelo valor da função utilidade.

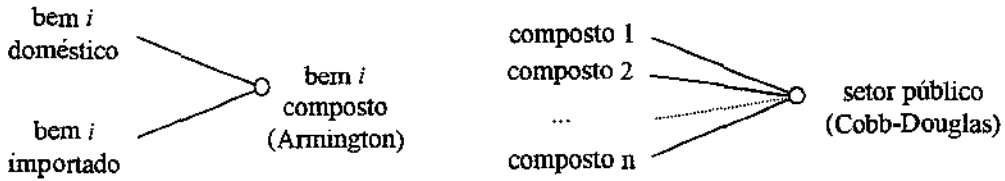


A demanda do setor privado por bens é satisfeita através de uma função CDE (*Constant Difference of Elasticities*), sendo cada bem demandado um composto entre o bem produzido domesticamente e o bem correspondente importado, diferenciados através da hipótese de Armington por uma função CES (*Constant Elasticity of Substitution*).

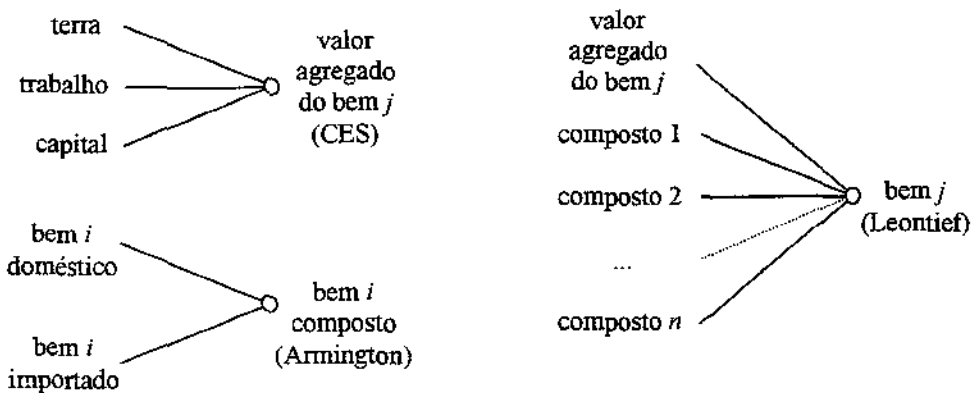


A demanda do setor público é semelhante à do setor privado, sendo utilizada uma função Cobb-Douglas no lugar da a função CDE. A separação da demanda intermediária do consumo em setores privado e público será útil para modelar cenários com políticas diferentes para cada um desses setores. Tem o mesmo objetivo o direcionamento das

importações para agentes específicos (setores privado e público e firmas, como será visto mais à frente).



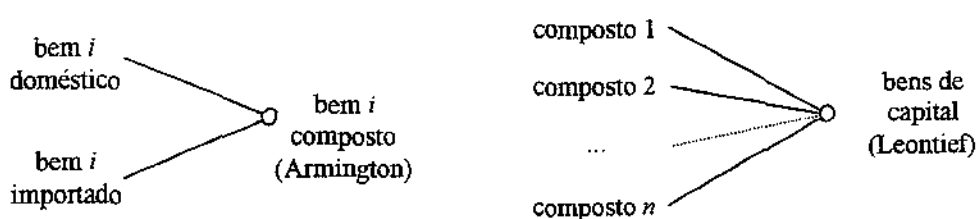
Supõe-se a produção em cada região dividida em setores, e considera-se a existência de apenas um bem produzido por uma firma em cada setor, ambos representativos do mesmo. As firmas combinam fatores primários, bens intermediários e importações através de funções que formam uma "árvore tecnológica" na produção de bens. A função Leontief é usada na junção final (raiz) da árvore devido à hipótese de separabilidade na produção – a escolha dos níveis ótimos dos fatores primários é feita independentemente dos preços dos bens intermediários. A alocação dos fatores primários de baixa mobilidade entre os setores produtivos é modelada pela função CET (*Constant Elasticity of Transformation*), e o valor agregado é obtido por uma CES. Como no consumo, na demanda intermediária das firmas bens domésticos e importados são diferenciados através da hipótese de Armington por uma função CES.



Quanto à sua finalidade, os bens são produzidos para 1) demanda final dos setores privado e público, 2) exportação, 3) demanda intermediária das firmas e 4) a montagem de *bens de capital* (*investment goods* no original) para investimento.

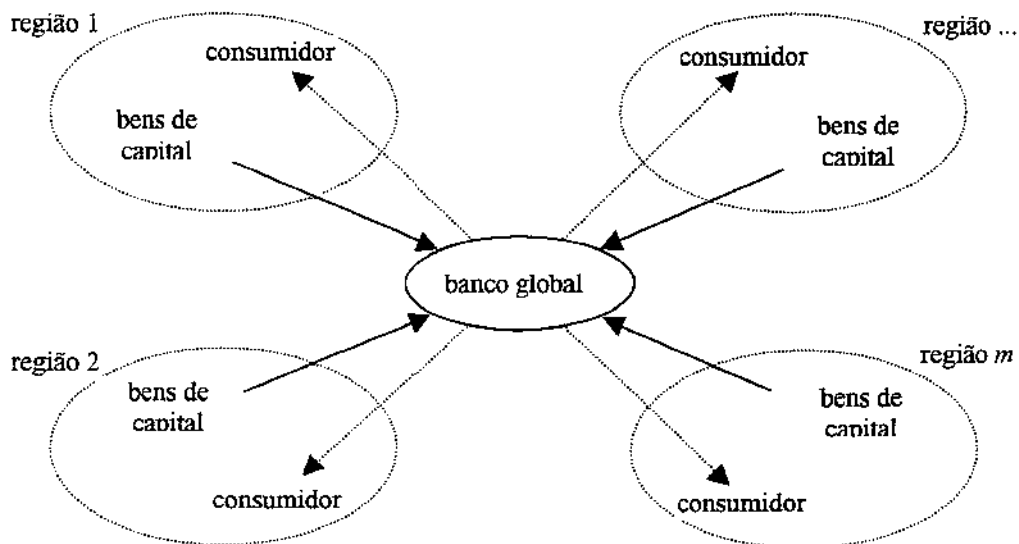
Embora os bens de capital façam parte do conjunto de bens produzidos, não fazem parte do conjunto de bens comercializáveis (*tradeables*), utilizados nas demandas intermediárias da região. Têm um tratamento diferenciado no modelo, tanto na "produção" como no "consumo" e não são tributados por impostos. (Os bens de capital regionais serão agregados em um *bem de capital global* para investimento, utilizado para satisfazer as decisões de poupança dos consumidores, como será visto mais à frente.)

Na produção, a atividade fictícia de montagem de uma unidade de capital para investimento é análoga à produção de um bem comum, exigindo entradas em proporções fixas dos compostos intermediários, cada um dos quais é uma combinação de bens domésticos e importados. Entretanto, sua "árvore tecnológica" não possui o ramo de valor agregado, pois o uso de terra, trabalho e capital já está incorporado nos compostos intermediários usados na produção do bem de investimento:



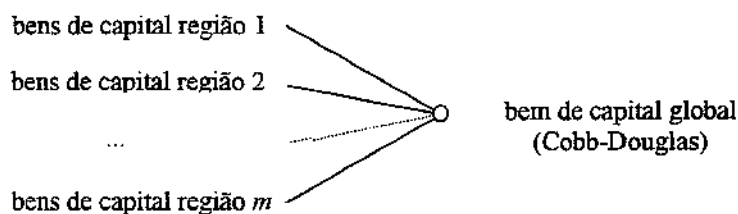
O "consumo" (decisões de poupança) dos bens de capital é feito através da intermediação de um *banco global*, que agrega o resultado líquido (total produzido menos parcela de depreciação) dos bens de capital produzidos nas diversas regiões em um *bem de capital global* que será posteriormente disponibilizado como investimento aos consumidores regionais. Nas palavras do autor, "o banco global monta uma carteira

(*portfolio*) de bens de capital regionais e vende ações dessa carteira aos consumidores regionais, para satisfazer a demanda desses consumidores por poupança”.



A alocação do estoque global de capital em investimentos regionais pode ser feita de duas maneiras: em uma delas, a composição do estoque é alterada em função da taxa de retorno sobre o capital; na outra, supõe-se que composição do estoque não é alterada.

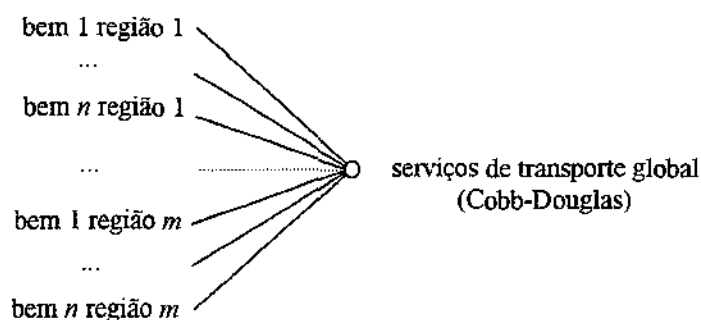
A agregação dos bens de capital é feita por uma função Cobb-Douglas:



O bem de capital global agregado pelo banco global é considerado o numerário do modelo no fechamento padrão.

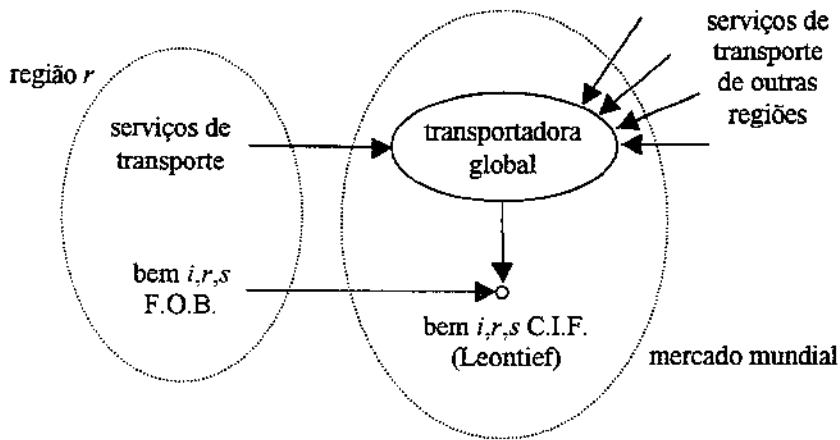
Existe um segundo setor global no modelo, uma *transportadora global*, que agrega as contribuições dos bens para os serviços de comércio, transporte e seguros de cada região (através de uma função Cobb-Douglas) e os disponibiliza como um bem composto *serviços*

de transporte global utilizado para o transporte de bens entre regiões. Esse formato é utilizado tendo em vista não estarem disponíveis dados associando serviços de transporte de rotas específicas a regiões determinadas.

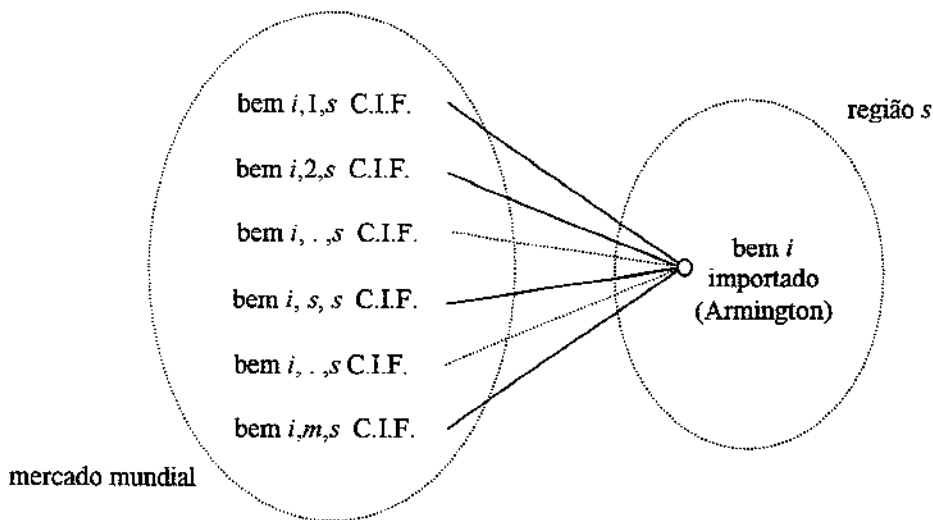


Para uma análise mais aprofundada dos fluxos de comércio internacional, as transações internacionais são individualizadas por origem e destino. Entretanto, para associar impostos de exportação ao consumidor da região exportadora e impostos de importação ao consumidor da região importadora é feita uma separação fictícia da transação de comércio internacional em duas transações interligadas, uma que transfere o bem da região exportadora para um "mercado mundial" fictício e outra que transfere o bem deste mercado mundial para a região importadora.

Assim, na primeira transação supõe-se que o bem é transferido da região r para o mercado mundial, e o valor correspondente dos serviços de transporte para o todo o percurso (de r até s) é agregado ao valor global F.O.B. do bem por uma função Leontief, obtendo-se o valor global C.I.F. do bem exportado de r para s colocado no mercado mundial:



Na segunda transação o bem comercializado de r para s é agregado a todas as outras importações do mesmo bem C.I.F. originados de diversas regiões exportadoras e destinados à mesma região s colocadas no mercado mundial. Nessa agregação as importações são diferenciadas através da hipótese de Armington por uma função CES:



Deve ser notado que é possível um bem ser exportado para ou importado de sua própria região, e isso se deve ao fato que as regiões (no caso mais geral) serem compostas por agrupamentos de diversas nações.

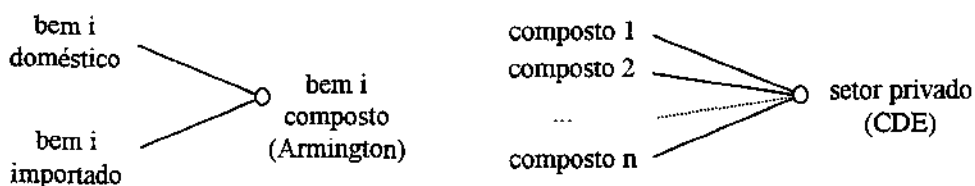
3.3.2 Os arquivos de dados do GTAP

Na versão 4 do banco de dados as informações das 45 regiões e 50 setores estão contidas em um arquivo binário de nome GLOBAL.HAR, de onde são extraídas agregações mais compactas através do programa DATA-AGG, fornecido com o banco de dados. Até a versão 6.0 do Gempack as agregações assim obtidas são fisicamente contidas em três arquivos: dois do tipo HAR (*Header Array*), um deles composto de matrizes de valores numéricos e o outro contendo os conjuntos dos nomes de bens e regiões; o terceiro é do tipo DAT (texto), contendo parâmetros de modelagem.

Será visto inicialmente o arquivo de matrizes numéricas, as quais contém informações de dotações, produção, investimento e consumo para cada região e informações bilaterais de comércio e transporte entre regiões. A produção é detalhada com os dados de entrada-saída que contabilizam as ligações inter-setoriais na região.

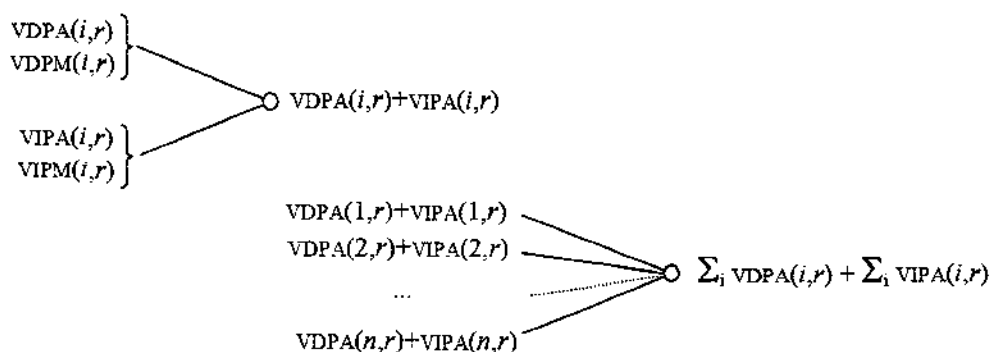
As informações são relativas às transações realizadas por agentes econômicos em mercados de bens (ambos pertencentes a uma mesma região), existindo assim dois tipos de valores para uma transação: valor a preço de agente e valor a preço do mercado. As informações de impostos, subsídios e taxas são derivadas desses dois valores: se existir diferença entre eles, essa diferença corresponde a um imposto ou subsídio que deverá ser somado ou subtraído à receita do consumidor representativo da região.

Na estrutura de gastos do consumidor representativo será vista inicialmente a transação que representa os gastos do setor privado em bens produzidos na região – o ramo doméstico da Armington que forma um bem composto da demanda do setor privado do consumidor:



Os valores dessa transação estão contidos no arquivo em duas matrizes (onde cada linha corresponde a um bem e cada coluna a uma região): uma matriz de valores a preço de agente, VDPA (*Value of Domestic purchases by the Private household, at Agents' prices*) e outra de valores a preço de mercado, VDPM (*Value of Domestic purchases by the Private household, at Market prices*). A regra geral (com exceções) da notação para os nomes das matrizes utiliza o padrão: *valor / tipo de transação / tipo de preço*, sendo todas as matrizes do banco de dados relacionadas na tabela A6.3 do apêndice A6.

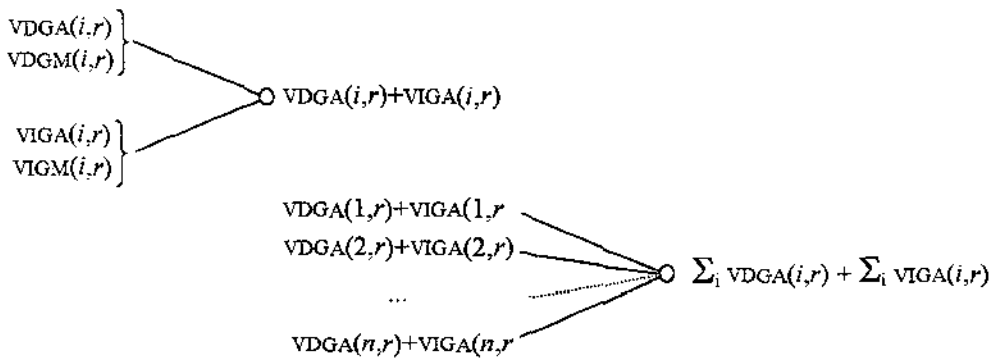
De maneira semelhante, os gastos do setor privado em importados (o outro ramo da Armington) está contido em duas matrizes: VIPA (*Value of Imports by the Private household, at Agents' prices*) e VIPM (*Value of Imports by the Private household, at Market prices*). Colocando os valores para um determinado bem i na estrutura de consumo:



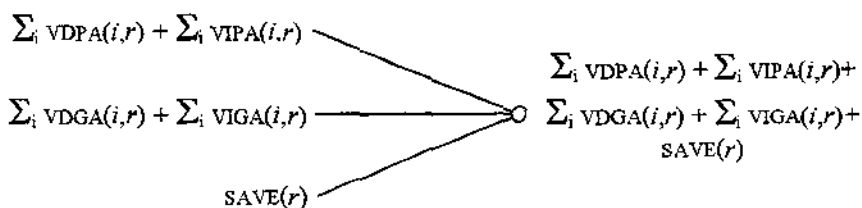
O arquivo contém somente os valores a preço de agente e a preço de mercado das folhas da árvore de consumo (transações do nível inicial da estrutura de consumo), sendo

todos os outros valores da estrutura derivados por agregação dos valores a preço de agente das folhas.

De maneira análoga os gastos do setor público em bens domésticos – conforme o padrão da notação basta substituir o P de *Private* pelo G de *Government* – estão em duas matrizes, VDGA (*Value of Domestic purchases by the Government, at Agents' prices*) e VDGM (*Value of Domestic purchases by the Government, at Market prices*), e os importados em VIGA (*Value of Imports by the Government, at Agents' prices*) e VIGM (*Value of Imports by the Government, at Market prices*):



Para completar a estrutura dos gastos do consumidor basta agregar às demandas dos setores privado e público obtidas acima a demanda por poupança, que está em um vetor SAVE (*Savings - Net Expenditure at Agents Prices*), com um elemento para cada região. A poupança não é tributada, existindo apenas o valor a preço de agente:



Observar que a contribuição à renda do consumidor r devido ao consumo é calculada somente nas folhas da árvore de consumo, e é igual a:

$$IRCS(r) = \sum_i (VDPA(i,r) - VDPM(i,r) + VIPA(i,r) - VIPM(i,r) + VDGA(i,r) - VDGM(i,r) + VIGA(i,r) - VIGM(i,r))$$

Continuando com a descrição das matrizes do arquivo serão apresentadas as informações dos setores produtivos, contidas em matrizes de três níveis, onde cada linha corresponde a um bem, cada coluna a um setor e em profundidade tem-se as regiões.

Na árvore de produção do setor correspondente ao bem j as informações das folhas estão contidas nas matrizes EVFA (*Endowments - Firms' Purchases at Agents' Prices*), VFM (*Endowments - Firms' Purchases at Market Prices*), VDFA (*Intermediates - Firms' Domestic Purchases at Agents' Prices*), VDFM (*Intermediates - Firms' Domestic Purchases at Market Prices*), VIFA (*Intermediates - Firms' Imports at Agents' Prices*) e VIFM (*Intermediates - Firms' Imports at Market Prices*).

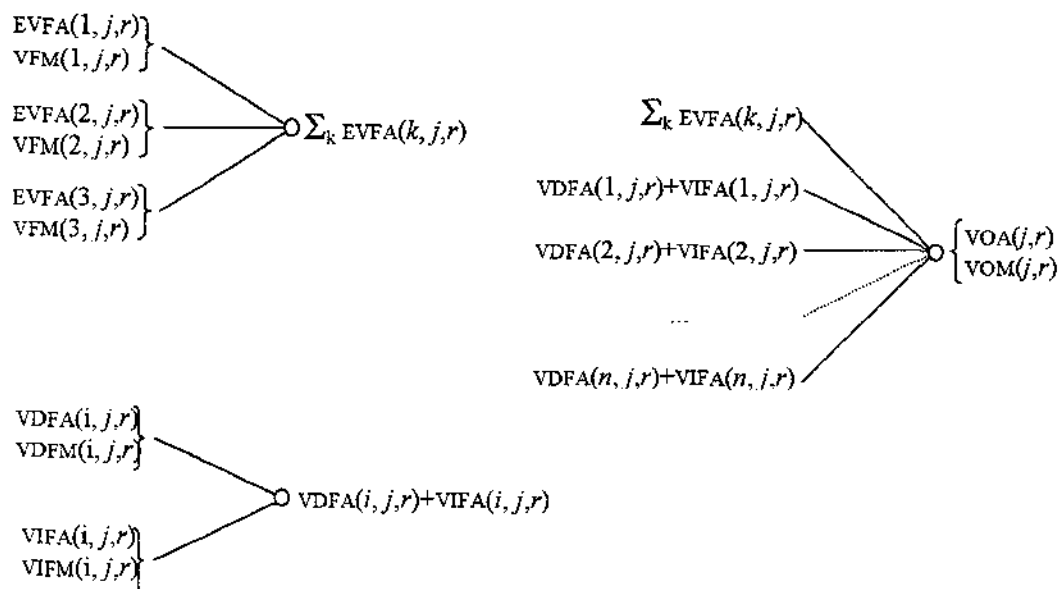
O valor a preço de agente do bem produzido (VOA – *Value of Output at Agent's Prices*), como nos casos anteriores, é obtido por agregação dos valores a preço de agente das folhas da árvore de produção:

$$VOA(j,r) = \sum_k EVFA(k,j,r) + \sum_i VDFA(i,j,r) + \sum_i VIFA(i,j,r)$$

O valor a preço de mercado do bem produzido (VOM – *Value of Output at Market Prices*), visando garantir a consistência dos dados, deve ser obtido por agregação de todos os valores a preço de mercado do bem j demandados na região (demandas intermediárias privada, pública e das firmas, já vistos anteriormente, e outras duas matrizes que serão vistas mais à frente: exportações e serviços de transporte):

$$VOM(j,r) = VDPM(j,r) + VDGM(j,r) + \sum_i VDFM(j,i,r) + \sum_s VXMD(j,r,s) + VST(j,r)$$

Colocando os valores para um determinado setor j na estrutura de produção:



A contribuição à renda do consumidor r relativa à demanda intermediária das firmas, como no caso anterior, é calculada somente nas folhas da árvore de produção:

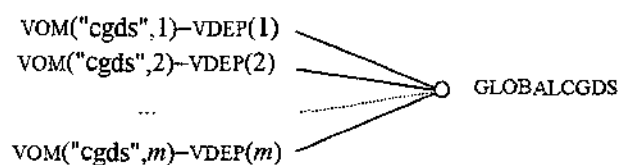
$$IRFS(r) = \sum_j (\sum_k (EVFA(k, j, r) - VFM(k, j, r)) + \sum_i (VDFA(i, j, r) - VDFM(i, j, r)) + \sum_i (VIFA(i, j, r) - VIFM(i, j, r)))$$

A contribuição à renda do consumidor r relativa à produção é calculada pela diferença dos valores a preço de agente e mercado do bem j produzido:

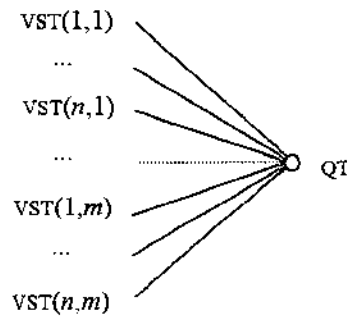
$$IRPS(r) = VOA(j, r) - VOM(j, r)$$

Serão vistos a seguir os dois setores globais, investimentos e serviços de transporte. Em ambos os casos supõe-se que não sejam devidos impostos ao cruzar a fronteira da região.

Os dados de investimento são obtidos subtraindo da produção regional de bens de investimento VOM (*Value of Output at Market Prices*) a depreciação VDEP (*Capital Stock - Value of Depreciation*), um vetor com um elemento para cada região:



O serviços de transporte estão contidos em uma matriz VST (*Trade - Exports for International Transportation, Market Prices*), onde as linhas correspondem às parcelas que cada bem contribui para serviços de transporte e as colunas às regiões:

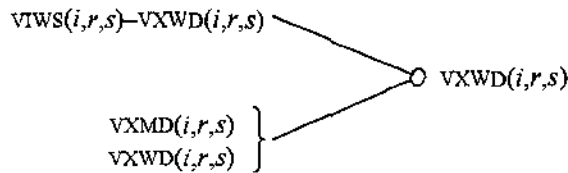


Serão vistas a seguir as matrizes que detalham o comércio internacional. Na descrição do modelo (seção anterior) viu-se que transações entre regiões são feitas em duas etapas, sendo o bem inicialmente exportado para um "mercado mundial" e posteriormente direcionado para a região importadora. A separação fictícia da transação de comércio internacional em duas transações interligadas irá permitir associar impostos de exportação ao consumidor da região exportadora e impostos de importação ao consumidor da região importadora.

Na primeira etapa, os "valores de mercado" são os valores FOB do mercado mundial e os "valores de agente" são os valores de mercado da região exportadora, respectivamente nas matrizes VXWD (*Trade - Bilateral Exports at World Prices - FOB*) e VXMD (*Trade - Bilateral Exports at Market prices*). Na segunda etapa, os "valores de mercado" são os valores CIF do mercado mundial e os "valores de agente" são os valores de mercado da região importadora, respectivamente nas matrizes VIWS (*Trade - Bilateral Imports at World Prices - CIF*) e VIMS (*Trade - Bilateral Imports at Market Prices*). Nessas matrizes as linhas correspondem aos bens, as colunas à região exportadora e em profundidade tem-se a região importadora.

Como anteriormente, as diferenças entre valores de "agente" e "mercado" significam um imposto ou subsídio lançados à renda do consumidor regional, exportador ou importador.

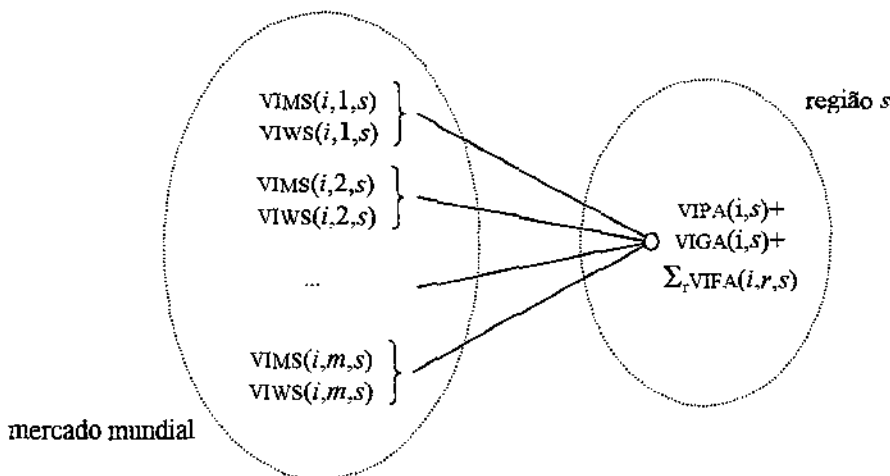
Na primeira etapa da transação, que transfere o bem exportado da região r para o mercado mundial, serão agregados o valor do bem e o custo dos serviços da transportadora global, que é obtido como a diferença entre o valor a preço mundial do bem importado (CIF) e o valor a preço mundial do bem exportado (FOB):



A contribuição à renda do consumidor r relativa às exportações é calculada como:

$$IRXS(r) = \sum_s (\sum_i (VXWD(i,r,s) - VXMD(i,r,s)))$$

Na segunda etapa da transação o bem comercializado de r para s é agregado a todas as outras importações do mesmo bem originadas de diversas regiões exportadoras ($r = 1, \dots, m$) e colocadas no mercado mundial com destino à mesma região s :



A contribuição à renda do consumidor s relativa à importação é calculada como:

$$IRIS(s) = \sum_r (\sum_i (VIWS(i,r,s) - VIMS(i,r,s)))$$

Por coerência com a exportação do bem da região r para a região s a agregação da importações e o cálculo da contribuição à renda acima foram feitos para a região s . Os resultados são análogos para importações de diversas regiões para a região r e a contribuição à renda da região r relativa às importações é:

$$IRIS(r) = \sum_s (\sum_i (VIWS(i,s,r) - VIMS(i,s,r)))$$

Com a descrição do comércio bilateral entre regiões se encerra a descrição do arquivo de matrizes numéricas. Observe-se serem essas as matrizes necessárias para construir um modelo com o arcabouço básico do GTAP, e que existem outras matrizes com informações que não serão aqui apresentadas. Nas versões mais recentes do banco de dados existem ainda outras matrizes, que não estão relacionadas no apêndice A6.3, que possibilitam construir variações do modelo GTAP para análise de questões específicas.

A contribuição total à renda do consumidor r é dada por:

$$IRS(r) = IRCS(r) + IRFS(r) + IRPS(r) + IRXS(r) + IRIS(r)$$

A seguir serão vistos os outros dois arquivos do banco de dados.

O segundo arquivo do tipo HAR contém os conjuntos e subconjuntos de bens e regiões, descritos na tabela a seguir:

H1	REG	regiões
H2	TRAD_COMM	bens comercializáveis
H3	NSAV_COMM	bens que não sejam bens de poupança
H4	DEMD_COMM	bens demandados
H5	PROD_COMM	bens produzidos
H6	ENDW_COMM	fatores primários
H7	ENDWS_COMM	fatores primários de pouca mobilidade
H8	ENDWM_COMM	fatores primários móveis
H9	CGDS_COMM	bens de capital

O terceiro arquivo é do tipo texto e contém diversos parâmetros do modelo, descritos na tabela a seguir:

SUBPAR	parâmetro de substituição na função CDE
INCPAR	parâmetro de expansão na função CDE
ESUBD	elasticidade de substituição entre bens domésticos e importados na estrutura de agregação Armington, para todos os agentes em todas regiões
ESUBM	elasticidade de substituição entre bens importados de diferentes origens na estrutura de agregação Armington, para todos os agentes em todas regiões
ESUBVA	elasticidade de substituição entre capital, trabalho e possivelmente terra, na produção do valor agregado de um setor de produção
ETRAE	elasticidade de transformação para fatores primários de baixa mobilidade
RORFLEX	flexibilidade da taxa de retorno líquida esperada em relação ao estoque de capital, por região
RORDELTA	coeficiente binário (0,1) para determinação da alocação de investimento entre regiões: no primeiro caso a alocação é feita entre regiões mantendo a composição existente de estoques de capital e no segundo é feita acompanhando a mudança nas taxas de retorno esperadas das regiões.

4. Aplicações

*Two roads diverged in a yellow wood,
And sorry I could not travel both...
(Robert Frost)*

Neste capítulo apresenta-se uma aplicação do conjunto de métodos descritos ao arcabouço de modelo e dados do GTAP. Observe-se que para resolver esse modelo, em sua grande maioria, os pesquisadores usam o Gempack ([31], [32]), da escola de modelagem em taxas de crescimento. Outros usam, ainda, o GAMS/MPSGE ([53], [54]), que como o *Pegasus*, pertence à escola de modelagem em níveis. Em qualquer caso, o arcabouço de modelo e os dados são sempre os fornecidos pelo GTAP. Assim, nas simulações deste trabalho os modelos de comércio internacional são construídos de acordo com a proposta do GTAP mas representados e resolvidos usando-se o *Pegasus/Mathematica* com o método de resolução de sistemas não-lineares ACI.

É importante ressaltar que essa aplicação será realizada em uma aproximação ao arcabouço de modelo do GTAP, onde as principais simplificações feitas são a substituição das funções CDE do modelo original por funções CES, a suposição de mobilidade de todos os fatores primários⁸ e uma possibilidade de alocação de investimento, que são alocados mantendo a composição existente dos estoques de capitais.

Essa simplificação ao arcabouço do GTAP é necessária para os modelos de grande porte, pois apesar do *Pegasus/Mathematica* dispor dos procedimentos %CET e %CDE – e portanto ser possível implementar o arcabouço de modelo do GTAP sem simplificações – a

⁸ Nos exemplos a serem aqui apresentados existe somente um fator de baixa mobilidade (land), utilizado em apenas um setor produtivo (food), e portanto, nesse caso, os resultados não serão influenciados por essa simplificação.

avaliação dessas funções ainda não foi implementada no procedimento ACI, o que se pretende que seja feito em uma próxima versão desse procedimento.

4.1 Aplicação dos métodos ao modelo e dados do GTAP

Escolheu-se para esta aplicação o exemplo didático do GTAP apresentado por Hertel e Tsigas [34], que utiliza uma agregação do banco de dados em três regiões e três setores produtivos, a qual será aqui referenciada como o modelo GTAP3x3. Nesse exemplo didático os autores fazem um experimento de simulação (um choque na economia do modelo) que consiste em reduzir a tarifa de importação de um dos setores entre duas regiões; esse mesmo experimento será aqui reproduzido e os resultados obtidos comparados com os apresentados em [34].

Um passo preliminar é a preparação de um arquivo texto de dados a partir dos arquivos do tipo HAR e DAT, descritos na seção anterior, que possa ser lido pelo *Mathematica*⁹; supõe-se portanto que isso já tenha sido feito e exista um arquivo texto GTAP3X3.DAT (reproduzido no apêndice A6) com os dados do modelo GTAP3x3.

A seguir, os comandos que carregam o *Pegasus*:

```
In[1]:=
SetOptions[{$Output, PageWidth -> 90};
SetDirectory["\Pegasus"];
Get["Pegasus.s3"];
Pegasus 1.2 (October 30, 1999)
```

Os dados do GTAP3x3 são lidos:

```
In[4]:=
Get["\Doc\Tese\gtap3x3.dat"];
```

Após a leitura estão disponíveis as regiões, definidas no conjunto s_R , e os bens comercializáveis (*tradeables*), definidos no conjunto s_T , mostrados a seguir:

⁹ É possível (e trivial) a construção no *Mathematica* de uma interface de leitura direta em arquivos HAR desde que a informação do formato binário desse tipo de arquivo seja disponibilizada.


```

In{5}:=
  sR
  sT
Out{5}=
  {usa, eu, row}
Out{6}=
  {food, mnfcs, svces}

```

Existem mais dois conjuntos de bens: os fatores primários, definidos no conjunto sE , e os bens produzidos na economia, definidos no conjunto sP (constituído pelos bens comercializáveis e o bem $cgds$, que representa os bens de capital):

```

In{7}:=
  sE
  sP
Out{7}=
  {land, labor, capital}
Out{8}=
  {food, mnfcs, svces, cgds}

```

O conjunto de regiões e estes dois últimos conjuntos de bens darão origem aos bens do modelo, que serão indexados por região e agente ou uso, obtendo-se assim 129 bens, aos quais se somarão o bem representativo dos serviços de transporte globais (qt) e o bem representativo dos bens de capital globais ($globalcgds$), num total de 131 bens para o modelo GTAP3x3, relacionados em uma tabela na próxima página, e o significado das abreviações utilizadas na indexação é dado a seguir:

$bem_i[r]$	bem_i produzido na região r
$bem_i[va, r]$	valor agregado de fatores primários na produção do bem_i na região r
$bem_i[fd, r]$	demanda intermediária doméstica das firmas na produção do bem_i na região r
$bem_i[bem_i, r]$	quantidade do bem_i utilizada na produção do bem_i (setor $_i$) na região r
$bem_i[im, r]$	quantidade importada (por diversos agentes) do bem_i na região r
$bem_i[cif, r_1, r_2]$	quantidade cif do bem_i exportada da região r_1 para a região r_2
$bem_i[pd, r]$	demanda privada do bem_i na região r
$bem_i[gd, r]$	demanda pública do bem_i na região r
$pd[r]$	bem representativo da demanda privada agregada da região r
$gd[r]$	bem representativo da demanda pública agregada da região r

A geração dos bens do modelo e das atividade de produção (incluindo os bens representativos da demanda privada e pública) e consumo correspondentes serão feitas auto

Tabela de bens do modelo, indexados por região e agente ou uso

land[usa]	land[eu]	land[row]
labor[usa]	labor[eu]	labor[row]
capital[usa]	capital[eu]	capital[row]
food[va, usa]	food[va, eu]	food[va, row]
food[fd, usa]	food[fd, eu]	food[fd, row]
food[im, usa]	food[im, eu]	food[im, row]
food[food, usa]	food[food, eu]	food[food, row]
mnfcs[food, usa]	mnfcs[food, eu]	mnfcs[food, row]
svces[food, usa]	svces[food, eu]	svces[food, row]
food[usa]	food[eu]	food[row]
mnfcs[va, usa]	mnfcs[va, eu]	mnfcs[va, row]
mnfcs[fd, usa]	mnfcs[fd, eu]	mnfcs[fd, row]
mnfcs[im, usa]	mnfcs[im, eu]	mnfcs[im, row]
food[mnfcs, usa]	food[mnfcs, eu]	food[mnfcs, row]
mnfcs[mnfcs, usa]	mnfcs[mnfcs, eu]	mnfcs[mnfcs, row]
svces[mnfcs, usa]	svces[mnfcs, eu]	svces[mnfcs, row]
mnfcs[usa]	mnfcs[eu]	mnfcs[row]
svces[va, usa]	svces[va, eu]	svces[va, row]
svces[fd, usa]	svces[fd, eu]	svces[fd, row]
svces[im, usa]	svces[im, eu]	svces[im, row]
food[svces, usa]	food[svces, eu]	food[svces, row]
mnfcs[svces, usa]	mnfcs[svces, eu]	mnfcs[svces, row]
svces[svces, usa]	svces[svces, eu]	svces[svces, row]
svces[usa]	svces[eu]	svces[row]
food[cgds, usa]	food[cgds, eu]	food[cgds, row]
mnfcs[cgds, usa]	mnfcs[cgds, eu]	mnfcs[cgds, row]
svces[cgds, usa]	svces[cgds, eu]	svces[cgds, row]
cgds[usa]	cgds[eu]	cgds[row]
food[pd, usa]	food[pd, eu]	food[pd, row]
mnfcs[pd, usa]	mnfcs[pd, eu]	mnfcs[pd, row]
svces[pd, usa]	svces[pd, eu]	svces[pd, row]
pd[usa]	pd[eu]	pd[row]
food[gd, usa]	food[gd, eu]	food[gd, row]
mnfcs[gd, usa]	mnfcs[gd, eu]	mnfcs[gd, row]
svces[gd, usa]	svces[gd, eu]	svces[gd, row]
gd[usa]	gd[eu]	gd[row]
food[cif, eu, usa]	food[cif, row, eu]	food[cif, row, row]
food[cif, row, usa]	mnfcs[cif, row, eu]	mnfcs[cif, row, row]
mnfcs[cif, eu, usa]	svces[cif, row, eu]	svces[cif, row, row]
mnfcs[cif, row, usa]	food[cif, eu, row]	
svces[cif, eu, usa]	mnfcs[cif, eu, row]	
svces[cif, row, usa]	svces[cif, eu, row]	
food[cif, usa, eu]		
food[cif, usa, row]		
mnfcs[cif, usa, eu]		qt
mnfcs[cif, usa, row]		globalcgds
svces[cif, usa, eu]		
svces[cif, usa, row]		

maticamente por um conjunto de procedimentos implementados em *Mathematica*, constituindo o pacote GTAPproc (reproduzido no apêndice A7). O comando a seguir carrega esse pacote, disponibilizando seus procedimentos:

```
In[9]:=
Get["\Doc\Tese\GTAPproc.m"];
Cobb-Douglas elasticity used in CES: 0.999
Truncation value limit: 1. 10-6
Non-CRS activity relative weight: 1. 10-11
```

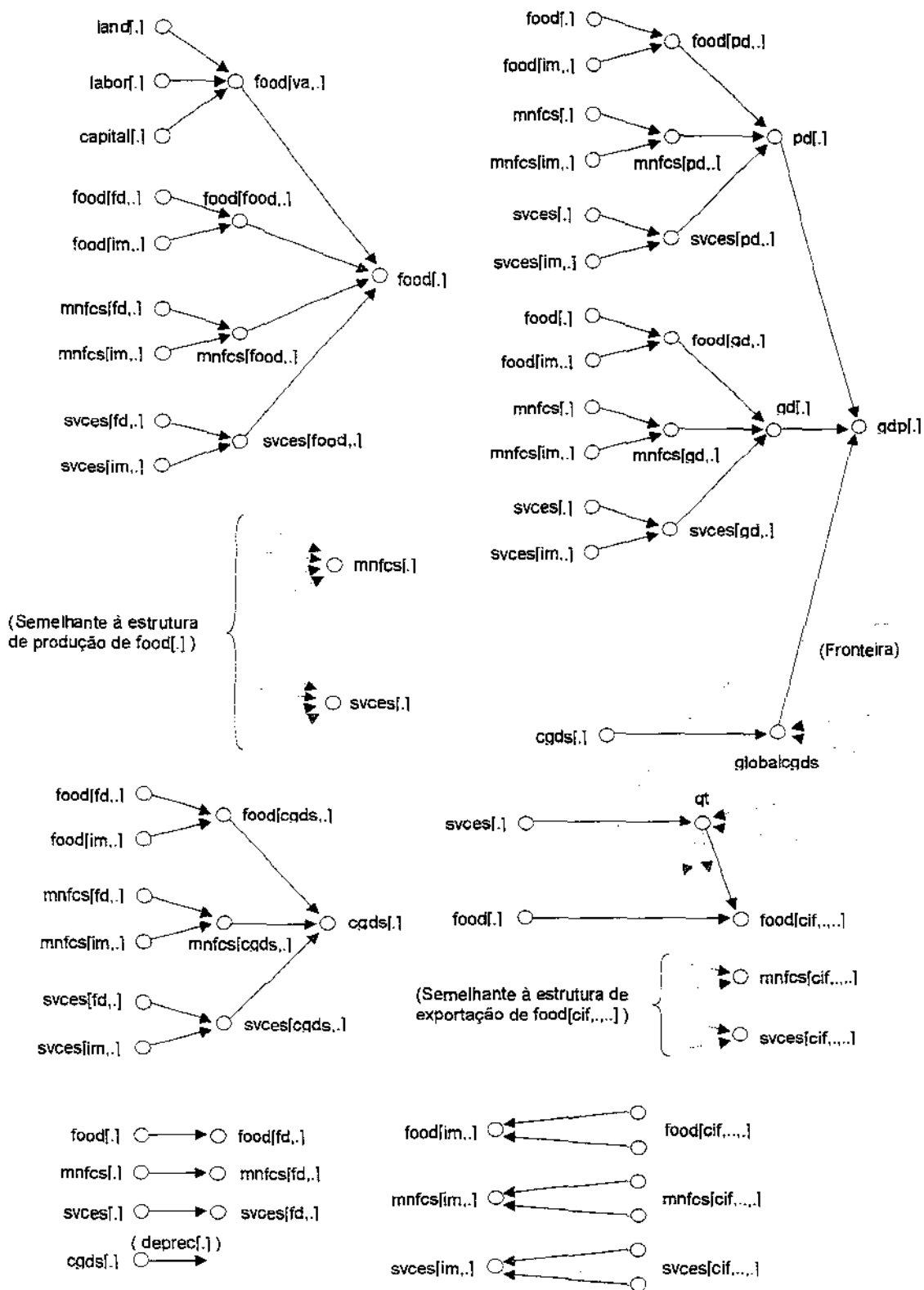
Observe-se no comando anterior que são mostrados os valores de algumas constantes utilizadas internamente pelo pacote.

O comando a seguir executa o procedimento `$DataPrep[]`, onde é feita uma preparação dos dados e verificação de sua consistência, além da agregação do valor da produção a preço de mercado e a preço de agente por região (VOM e VOA):

```
In[10]:=
$DataPrep[]
Endowment Commodities: 3
Tradeable Commodities: 3
Produced Commodities: 4
Regions: 3
Surplus: 3.03317
```

O conjunto de comandos a seguir, onde serão utilizadas funções CES para todas as agregações do modelo, gera os bens e as atividades de produção para todas as regiões, conforme as agregações vistas no capítulo anterior e ilustradas na figura da próxima página:

```
In[11]:=
producao[0]={};
For[i=1,i<=nR,i++,r=sR[[i]]];
  producao[0]=
  Join[producao[0],
    $SectorOutput[r],
    $PrivateDemand[r],
    $PublicDemand[r],
    $Imports[r],
    $Exports[r],
    $Depreciation[r]
  ];
];
producao[0]=
Join[producao[0],
  $GlobalTransport[],
  $GlobalCGDS[]
];
```



Estruturas de produção, consumo, investimento e comércio para uma região.

Os comandos a seguir agregam os bens representativos da demanda privada e pública com a poupança ($pd[r]$, $gd[r]$ e $SAVE[r]$) no consumo, para todas as regiões:

```
In{14}:=
consumo{0}={};
For{i=1,i<=nR,i++, r=sR[[i]]};
consumo{0}=
Join[consumo{0},
$RegionConsumption[r]
];
};
```

A produção e consumo obtidos são utilizados com o procedimento `$FindEquilibrium`, definindo-se como numerário o bem `globalcgds`:

```
In{16}:=
modelo{0}=$FindEquilibrium[
$Production->producao{0},
$Consumption->consumo{0},
$Numeraire->globalcgds,$Report->None,$Solver->ACI
];
$Time[modelo{0}]
$Solution[modelo{0}]
```

A solução obtida é apresentada no apêndice A6.5 e como esperado é composta por valores unitários, com uma precisão de 10^{-5} . Essa solução corresponde à solução do modelo sem nenhuma alteração, o chamado caso base.

O experimento de simulação a ser realizado consiste em reduzir a tarifa de exportação de alimentos da região USA com destino à região EU em 10% às expensas da região USA, ou seja, o preço do bem `food[cif,usa,eu]` na atividade `food[im,eu]` deve ser multiplicado por 0.9, e as alterações à renda atribuídas à região USA.

Nos exemplos aqui apresentados anteriormente a multiplicação de um preço por um fator foi feita na construção de várias atividades, e seria possível aqui repetir esse procedimento, mas seria necessário construir de novo todas as atividades de produção de `producao{0}`. Entretanto, o pacote `GTAPproc` dispõe do procedimento `$Shock` para alterar diretamente o preço de um bem em uma atividade em uma produção já construída, e esse

procedimento é utilizado a seguir, informando-se a produção a ser alterada, o bem, a atividade, o consumidor e o fator a serem utilizados:

```
In{24}:=
  producao[1]=$Shock[producao[0],food[cif,usa,eu],food[im,eu],usa,0.9];
```

A nova produção deve ser utilizada com \$FindEquilibrium:

```
In{26}:=
  modelo[1]=$FindEquilibrium[
    $Production->producao[1],
    $Consumption->consumo[0],
    $Numeraire->globalegds,$Report->None,$Solver->ACTI
  ];
```

Os preços e níveis de atividade da solução obtida são apresentados no apêndice A6.6, ordenados por ordem decrescente de valores absolutos.

Nas tabelas a seguir os valores da variação percentual de preços e níveis de atividades publicados em Hertel e Tsigas [34] nas tabelas 2.20, p. 66, e 2.22, p. 68 (valores da simulação com uma carteira (*portfolio*) fixa de investimento, RORDELTA=0, com o método Gragg 2-4-6), são comparados com seus equivalentes aqui obtidos:

Região EU			
GTAP (Gempack)	<i>Pegasus</i>	GTAP (Gempack)	<i>Pegasus</i>
		%	%
pm(land, eu)	p[land[eu]]	-0.515	-0.424
pm(labor, eu)	p[labor[eu]]	-0.041	-0.038
pm(capital, eu)	p[capital[eu]]	-0.041	-0.037
pm(food, eu)	p[food[eu]]	-0.154	-0.148
pm(mnfcs, eu)	p[mnfcs[eu]]	-0.041	-0.038
pm(svces, eu)	p[svces[eu]]	-0.042	-0.038
pm(cgds, eu)	p[cgds[eu]]	-0.037	-0.034
qo(food, eu)	y[food[eu]]	-0.355	-0.289
qo(mnfcs, eu)	y[mnfcs[eu]]	0.086	0.078
qo(svces, eu)	y[svces[eu]]	0.012	0.007
qo(cgds, eu)	y[cgds[eu]]	-0.004	0.009

Região USA			
GTAP (Gempack)	<i>Pegasus</i>	GTAP (Gempack)	<i>Pegasus</i>
		%	%
pm(land, usa)	p[land[usa]]	1.378	1.369
pm(labor, usa)	p[labor[usa]]	0.141	0.136
pm(capital, usa)	p[capital[usa]]	0.162	0.157
pm(food, usa)	p[food[usa]]	0.181	0.176
pm(mnfc, usa)	p[mnfc[usa]]	0.129	0.125
pm(svcs, usa)	p[svcs[usa]]	0.144	0.139
pm(cgds, usa)	p[cgds[usa]]	0.123	0.119
qo(food, usa)	y[food[usa]]	0.886	0.884
qo(mnfc, usa)	y[mnfc[usa]]	-0.155	-0.151
qo(svcs, usa)	y[svcs[usa]]	-0.001	-0.002
qo(cgds, usa)	y[cgds[usa]]	-0.002	-0.023

Como pode ser observado, os valores são muito próximos em grandeza e sinal, exceto para qo(cgds, eu), uma variação percentual muito pequena, próxima de zero.

Para obtenção da variação equivalente, o *Pegasus* dispõe do procedimento `$IncomeVariation`, que será utilizada a seguir:

```
In[32]:=
$IncomeVariation[modelo[0]]/. $Solution[modelo[1]]
Out[33]=
{usa -> 965.114, eu -> 74.0221, row -> -413.638}
```

Na tabela a seguir os valores da variação equivalente da tabela 2.23, p. 69, obtidos com o mesmo método Grag 2-4-6, são comparados com seus equivalentes aqui obtidos:

Variação equivalente, em \$US milhões			
	USA	EU	ROW
GTAP/Gempack	1004.	62.	-396.
<i>Pegasus</i>	965.	74.	-413.

Observe-se novamente que os valores obtidos com o *Pegasus* são muito próximos dos valores obtidos com o Gempack.

As diferenças entre os resultados obtidos com os dois pacotes, dessa tabela e das anteriores, se devem às simplificações feitas ao arcabouço de modelo do GTAP realizadas na resolução pelo *Pegasus*, e não devem ser vistas sob a ótica que uma é a correta e a outra

incorreta. O ponto de vista que deve ser ressaltado é a proximidade de valores obtidos com métodos de modelagem e resolução tão distintos quanto os da escola de modelagem em níveis do *Pegasus* e da escola de modelagem em taxas de crescimento do Gempack.

4.2 Um modelo GTAP8x9 com países do Mercosul

Para uma aplicação a um modelo de grande porte escolheu-se uma agregação do banco de dados do GTAP em oito regiões e nove setores produtivos, que será referenciada como o modelo GTAP8x9 (o arquivo de definição da agregação – *template* – utilizado com o programa DATA-AGG para obtenção dessa agregação pode ser visto no apêndice A8.1).

As oito regiões consideradas são:

arg	Argentina
bra	Brasil
chi	Chile
nafta	Estados Unidos, Canadá e México
ram	Resto dos países das Américas
ue	União Européia
asia	Países da Ásia
row	Resto dos países do mundo

Os nove setores produtivos são:

AGR	Agricultura
PEC	Pecuária
OPRIM	Pesca e atividade florestal
MIN	Mineração (inclui petróleo e gás)
ALIM	Indústria de alimentação e couro
TRANSP	Fabricação de meios de transporte
MAQ	Máquinas e equipamentos
MNFC	Resto da ind. manufaturados
SERV	Resto da economia

Como anteriormente, supõe-se que exista um arquivo texto GTAP8X9.DAT, com os dados do modelo, preparado a partir dos arquivos do tipo HAR e DAT, que não será aqui reproduzido em razão de sua extensão. O conjunto de comandos que constrói o modelo e executa os procedimentos pode ser visto no apêndice A8.2. A seqüência de comandos é

semelhante à apresentada para o modelo GTAP3x3 até a obtenção da solução para o caso base ($\$FindEquilibrium$ para o modelo [0]), diferindo a partir desse ponto.

Em continuação, na página 166 do apêndice A8.2, existem alguns comandos para o cálculo do saldo da balança comercial do Brasil, e o valor obtido revela um déficit de US\$ 14,704 bilhões (esse resultado reflete a situação de 1995, ano base da versão 4 do banco de dados – em 2000, a balança comercial brasileira registrou déficit de US\$ 691 milhões, sendo esse valor resultado de exportações de US\$ 55,086 bilhões e importações de US\$ 55,777 bilhões, segundo o Ministério do Desenvolvimento).

Em um experimento de simulação, quer-se reduzir o déficit em 25%. Para isso, impõe-se um prêmio sobre os importados, mudando-se, assim, a taxa efetiva de câmbio (Dervis et al. [14], p. 195).

O procedimento será semelhante ao realizado na seção 3.2.5 (onde se desejava fixar um valor mínimo para a taxa salarial), sendo que aqui se condicionará o preço dos bens importados com uma sobretaxa igual a $m[restricao]$. Embora esta restrição afete diretamente ao mesmo tempo as importações e exportações, ela será acomodada no modelo distorcendo-se somente as importações. Obviamente, no equilíbrio, as exportações também sofrerão mudanças, assim como todas as variáveis endógenas do modelo, em menor ou maior grau. O valor de $m[restricao]$ será determinado exogenamente como o par complementar de uma restrição sobre a balança comercial:

$$bal_1 \leq 0.75 bal_0 \quad \text{ou} \quad 0.75 bal_0 - bal_1 \geq 0$$

E a restrição reformulada é

$$restricao = 0.75 bal_0 - bal_1, \text{ com } restricao \geq 0 : m[restricao] \geq 0$$

(a construção dessa restrição é feita também na página 166).

A seguir, na página 167, tem-se o procedimento `$FindEquilibrium` para o modelo[1] – o caso choque.

Antes de continuar com a análise econômica dos resultados, observe-se as estatísticas da execução do procedimento obtidas com a opção `$Trace->True`: a partir das informações de n e nz verifica-se que o sistema de equações não-lineares equivalente tem 3525 equações e o jacobiano tem 21525 elementos não-nulos, que o caracteriza como um sistema de grande porte. De lá também são obtidas as seguintes informações sobre os tempos de execução¹⁰ do modelo[1], editadas para uma melhor visualização:

1) Constructing NLCP ...	1195.89	Second
2) Compiling f	1335.51	Second
3) avaliacao f.....:	0.11	segundos
4) construcao jacobiano:	7.36	segundos
5) fatoracao matriz....:	5.28	segundos
6) tempo das iteracoes.:	5.43	segundos
7) tempo total ACI.....:	19.22	segundos
8) Total time elapsed...	2567.33	Second/42.7888 Minute

As linhas 1,2 e 8 acima apresentam informações de tempos medidas no ambiente *Mathematica*, e as linhas 3 a 7 tempos medidos no código ACI, implementado na linguagem C. Observe-se que se por um lado os tempos de execução do procedimento ACI são muito razoáveis, os tempos do *Mathematica* são extremamente demorados; podem ser explicados lembrando que o *Mathematica* é um ambiente de programação simbólica, onde cada comando é executado repetidamente para obter simplificações simbólicas, até que a expressão obtida na execução atual seja igual à execução anterior. No entanto, se a construção do modelo matemático fosse feita em C – e isso é possível para modelos específicos, como o do GTAP, embora perdendo os recursos e a generalidade do

¹⁰ Não foi possível obter estatísticas de tamanho e tempos de execução no Gempack, pois o modelo GTAP8x9 ultrapassou o limites permitidos na versão demonstração disponível. (Em uma nova versão dos procedimentos GTAPproc (ainda em *Mathematica*), implementada após a apresentação desta dissertação, o tempo de execução do GTAP8x9 foi reduzido para 12 minutos. Para o modelo GTAP3x3, nessa nova versão, o tempo total de execução do *Pegasus/ACI* é de 15,17 segundos, e o do Gempack é de 12,30 segundos.)

Pegasus/Mathematica –, uma estimativa grosseira do tempo necessário para a construção do problema seria igual ao tempo de resolução do sistema, ou seja, em torno de 40 segundos para o tempo total de construção e resolução do problema matemático, o que é razoável tendo em vista o porte do modelo e o microcomputador utilizado, um AMD K6-III, com 128 Mb de memória.

Retornando-se à análise do problema econômico, na página 168 e seguinte são feitos diversos cálculos auxiliares, necessários para essa análise.

No primeiro resultado observa-se que o valor de $m[\text{restricao}]$ é 1.207452, ou seja, a solução obtida nessa simulação indica que o prêmio sobre o preço dos importados para reduzir o déficit da balança comercial em 25% é de 20.74%.

Tem-se também os cálculos dos dados de poupança e investimento antes e depois do choque, que são dados nas tabelas a seguir:

Poupança e investimento antes do choque			
	poupança (S_0)	investimento (I_0)	$S_0 - I_0$
arg	8435.36	8579.77	-144.414
bra	55282.2	65076.5	-9794.25
chi	9987.57	9703.35	284.213
nafta	537569.	627976.	-90407.6
ram	18305.1	30431.5	-12126.4
ue	598232.	559048.	39184.4
asia	1.35623E+6	1.28213E+6	74103.
row	209994.	211093.	-1098.86
total	2.79404E+6	2.79404E+6	0.000001316

Poupança e investimento após o choque			
	poupança (S_1)	investimento (I_1)	$S_1 - I_1$
arg	8333.88	8573.91	-240.029
bra	56360.5	65037.1	-8676.64
chi	9929.07	9696.77	232.296
nafta	536505.	627551.	-91045.5
ram	18237.9	30410.9	-12172.9
ue	597114.	558669.	38445.3
asia	1.35406E+6	1.28126E+6	72800.4
row	209607.	210950.	-1342.67
total	2.79215E+6	2.79215E+6	0.000539735

Percebe-se que o ajuste macroeconômico para se obter o novo saldo na balança comercial dá-se basicamente através da poupança doméstica, sendo que o investimento mantém-se praticamente inalterado. (Para uma discussão sobre as chamadas "regras de

fechamento" que inclui as relações entre poupança, investimento e balança comercial, ver Hertel et al. [61], p. 266.)

Para os fatores primários tem-se:

Preços dos fatores primários após o choque			
	land	labor	capital
arg	0.984279	0.988662	0.989036
bra	1.03101	1.04702	1.04657
chi	0.994352	0.994477	0.994459
nafta	1.00052	0.998035	0.99804
ram	0.99587	0.996383	0.996458
ue	1.00246	0.998147	0.998154
asia	0.999257	0.998373	0.998375
row	1.00042	0.998144	0.998098

Quantidades utilizadas dos fatores primários após o choque			
	land	labor	capital
arg	8369.22	127023.	119219.
bra	9240.5	289529.	333400.
chi	1245.29	20564.4	33737.6
nafta	35758.4	4.59133*10 ⁶	2.7201*10 ⁶
ram	10884.	120651.	178333.
ue	23200.5	5.10616*10 ⁶	2.52361*10 ⁶
asia	153653.	3.7285*10 ⁶	3.00926*10 ⁶
row	49768.4	1.49814*10 ⁶	1.21802*10 ⁶

A partir das tabelas dos fatores primários calcula-se o índice de preços dos fatores primários global (média ponderada dos preços pelas quantidades), obtendo-se o valor 0.99925 para esse índice, e os índices de preços dos fatores primários regionais, que divididos por $pfacgm$ fornecem os índices de preços dos fatores primários regionais normalizados (indicados por $pfacgrn$) e sua variação percentual, na tabela a seguir:

Índices de preços dos fatores primários regionais		
	$pfacgrn$	% variação
arg	0.989435	-1,06
bra	1.04733	4,73
chi	0.995209	-0,48
nafta	0.998797	-0,12
ram	0.997155	-0,28
ue	0.998911	-0,11
asia	0.999143	-0,09
row	0.998913	-0,11

Com o aumento da "taxa efetiva de câmbio", representada pelo prêmio de 20.74% sobre os importados, há a redução do déficit na balança comercial em 25%, como exigido

pela restrição imposta. Com um alívio das contas externas, ocorre a apreciação em 4,73% dos preços relativos dos fatores primários domésticos em relação à cesta dos fatores primários de todas as regiões (o novo numerário, nesse caso) (ver comunicação privada (*e-mail*) de 24/11/1999 de T. Hertel aos participantes do GTAPLIST, reproduzida no apêndice A8.3).

A seguir, a variação equivalente de renda das regiões, onde se pode ver que todas as regiões perdem com esse choque:

Variação equivalente	
arg	-662.665
bra	-634.989
chi	-152.395
nafta	-1910.68
ran	-456.832
ue	-2091.44
asia	-2557.73
row	-825.425

Antes de finalizar, é necessário cautela com os resultados obtidos, tendo em vista que as informações da versão 4 do banco de dados para os países que não pertencem à OECD precisam ser corrigidas no que se refere à proteção do setor agrícola, como informa T. Hertel em comunicação privada (*e-mail*) de 24/11/1999 aos participantes do GTAPLIST, reproduzida a seguir nas partes pertinentes:

"... the weakest parts of the version 4 database – the agricultural protection data for non-OECD countries. When constructing the version 4 database, the latest information available to us was that compiled for the Uruguay Round studies – dating back to the late 1980's. These protection rates were adjusted in a systematic way to produce the version 4 agr protection for non-OECD economies – but there is no new information content on policies. (...) I think anyone doing serious analysis of agricultural protection in the non-OECD economies cannot avoid collecting supplementary data and using the ALTERTAX feature of RunGTAP to change the base protection rates to bring them in line with observed policies. It is not an adequate defense to simply assert that "these come from GTAP" and leave it at that. Ultimate

responsibility must rest with the individual conducting the study, as policies are continually changing and GTAP will never be fully up-to-date in all dimensions."

Dessa maneira, como a correção recomendada dos dados por Hertel não foi realizada por não se dispor dos dados de proteção atual, os resultados obtidos devem ser considerados no contexto de um exercício didático para demonstração da aplicação dos métodos aqui apresentados.

5. Conclusão

Não seria adequado comparar o *Pegasus* com pacotes comerciais como o Gempack ou o GAMS. Obviamente, do ponto de vista de um usuário empenhado em construir um modelo CEG, as muitas comodidades oferecidas por tais pacotes comerciais não poderiam ser desprezadas, mesmo levando em consideração seus altos custos. A opção por um ou outro pacote quase sempre resulta da familiaridade do usuário; o custo de escalar a "curva de aprendizagem" parece ser o fator determinante nessa escolha. O *Pegasus*, embora um trabalho em desenvolvimento continuado, seria capaz de satisfazer algumas das mais exigentes expectativas do usuário. Entretanto, talvez a motivação maior para seu desenvolvimento tenha sido simplesmente a satisfação intelectual de estudar as instigantes questões teóricas e empíricas envolvidas, a busca pelo conhecimento enfim.

Entre as contribuições principais do trabalho, podem-se citar:

- Implementação de um algoritmo para resolução de sistemas de equações não-lineares de grande porte, caracterizado pelo uso de alocação dinâmica de memória, representação esparsa de matrizes, armazenamento das equações do modelo em forma simbólica, o cálculo do jacobiano do sistema não-linear por diferenças finitas e a fatoração da matriz jacobiana através do pacote Harwell/MA28.
- Um conjunto de procedimentos *Mathematica* para a manipulação dos dados e construção de um modelo CEG segundo o construto do GTAP (*Global Trade Analysis Project*).
- Validação do ciclo completo de modelagem – obtenção das informações no banco de dados, construção, calibragem e solução - usando um modelo CEG do GTAP

apresentado por Hertel e Tsigas [27], tendo sido obtidos resultados muito próximos aos relatados naquele trabalho.

- Um exercício de simulação de políticas cambiais para o Brasil, usando um modelo CEG de grande porte com 8 regiões e 9 setores produtivos para o Mercosul.

Em suma, espera-se que tenha sido demonstrado neste trabalho o domínio de um conjunto abrangente de métodos para a construção, calibragem e resolução de modelos computáveis de equilíbrio geral de grande porte.

Para se chegar a esse resultado existiram algumas dificuldades que tiveram de ser superadas ao longo do percurso. Por exemplo, a necessidade de escalamento dos dados. Os procedimentos sem escalamento testados com sucesso no modelo 3x3 não resolviam modelos maiores que incluíam o Brasil como uma região. O que ocorria era que no modelo 3x3 os valores dos fluxos de renda eram da mesma grandeza e isso possibilitava a fatoração da matriz jacobiana mal-escalada; já com os dados do Brasil, pequenos em relação às outras regiões, a fatoração era impossível. Outra dificuldade surgiu em função de uma falha não-relatada (*bug*) da versão 3 do *Mathematica*: quando a função implementada em C retornava como resultado um vetor que continha um ou mais zeros, o *Mathematica* “travava”. Foram necessárias muitas horas de computador para se concluir que somente modelos com resultado sem zeros eram resolvidos com sucesso. Muitas horas também foram dedicadas a estudar o arcabouço de modelo e os dados do GTAP, consistentes com o pacote Gempack, que trabalha com o conceito da descrição de variáveis por variações percentuais ao invés de níveis.

Por último – mas nem por isso a menor dificuldade, tendo em vista ser minha formação básica em Ciência da Computação – meu despreparo para trilhar os mesmos

caminhos de Scarf, Mathiesen, Dixon, Robinson, Taylor e outros. Apesar dos esforços de meu orientador, acredito que continuarei sendo um profissional de métodos quantitativos que dispõe de um dicionário "Economia em mil palavras" e apenas consegue entender as necessidades de um economista.

E para o futuro esse dicionário deverá ser muito usado, pois existem várias contribuições que poderiam ser acrescentadas ao trabalho:

- Implementação de um procedimento com o objetivo específico de construção do modelo GTAP em linguagem C.
- Desenvolvimento de uma rotina que possa buscar as informações de valores de fluxos de renda diretamente de uma Matriz de Contabilidade Social em uma planilha *Excel*.
- Construção de procedimentos para modelar e resolver modelos CEG como problemas de complementaridade mistos (MCP), que combinam equações com desigualdades fracas.

6. Referências Bibliográficas

- [1] Adams, P.A., P.B. Dixon, D. McDonald, G.A. Meagher and B.R. Parmenter (1993) 'Forecasts for the Australian Economy Using the MONASH Model', *Paper presented to International Symposium on Economic Modelling, University of Piraeus, Greece.*
- [2] Adelman, I. e S. Robinson, *Income Distribution Policy in Developing Countries*, Stanford University Press, Stanford, California, 1978.
- [3] Arica, J., "Cálculo de Equilíbrio Econômico usando um Algoritmo de Ponto Fixo", tese de mestrado, Depto. Matemática Aplicada, UNICAMP. 1987.
- [4] Arrow, K. e G. Debreu, "Existence of equilibrium for a competitive economy", *Econometrica*, 22, pp. 265–290. 1954.
- [5] Bordoni, Orlando F. J. G., "PEGASUS: o Problema de Equilíbrio Geral Aplicado resolvido por um Sistema Unificado de Suporte Computacional", tese de mestrado, Depto. Matemática Aplicada, UNICAMP, Campinas, Brasil. 1989.
- [6] Broadie, M., 1985, "An Introduction to the Octahedral Algorithm for the Computation of Economic Equilibria", *Mathematical Programming Study*, Vol. 23, 121-143.
- [7] Broyden, C. G., "A class of methods for solving nonlinear simultaneous equations", *Math. Comput.* 19, pp 577-593 . 1965.
- [8] Brooke, A., Kendrick, D. e Meeraus, A., *GAMS: A User's Guide*, The Scientific Press, Redwood City. 1988.
- [9] Castro, S. A. "The structure of some general equilibrium models for Brazil", *Anais do IX Encontro Brasileiro de Econometria, Sociedade Brasileira de Econometria.* 1987.
- [10] Dahlquist, Germund, e Ake Björck, *Numerical Methods*, (traduzido por Ned Anderson) Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1974.
- [11] Dantzig, G. B. e Cottle, R.W., "Linear and nonlinear programming", em "Mathematical Theory of Optimization", *Encyclopaedia Britannica*, Vol XIII, 15th Edition, H.H. Benton, Publisher, 1974.
- [12] Duff, I.S., 'MA28 - A Set of FORTRAN Subroutines for Sparse Unsymmetric Linear Equations', Harwell Report R.8730 (HMSO London). 1977.

- [13] Duff, I. S., Erisman, A. M., Reid, J. K.. *Direct Methods for Sparse Matrices*. New York: Oxford University Press. 1986.
- [14] Dervis, K., J. de Melo, and S. Robinson, *General Equilibrium Models of Development Policy*, Cambridge University Press, Cambridge, England, 1982.
- [15] Dixon, P.B., B.R. Parmenter, J. Sutton and D.P. Vincent, *ORANI: A Multisectoral Model of the Australian Economy*, North-Holland, Amsterdam. 1982.
- [16] Dixon, P.B., B. R. Parmenter, A. A. Powell e P. J. Wilcoxon. *Notes and Problems in Applied General Equilibrium Economics*, Amsterdam: North-Holland. 1992.
- [17] Dixon, P.B., B. R. Parmenter, J. Sutton e D. P. Vincent. *Orani: A multisectoral model of the Australian economy*, Amsterdam: North-Holland. 1982.
- [18] Fang, Shu-Cheng, Puthenpura, S., *Linear Optimization and Extensions: Theory and Algorithms*, Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1993.
- [19] Feltestein, A., "A General Equilibrium Approach to the Analysis of Trade Restrictions with an Application to Argentina", *IMF Staff Papers*, vol 27 nro 4 pp 749-84. 1980.
- [20] Feltenstein, A., 1980, A general equilibrium approach to the analysis of trade restrictions with an application to Argentina. International Monetary Fund Staff Papers 27, 749-84
- [21] Feltenstein, Andrew and Shah, Anwar (1995), 'General Equilibrium Effects of Investment Incentives in Mexico', 46(2) Journal of Development Economics, 253-269
- [22] Ferreira Filho, J. B. S., *MEGABRÁS – Modelo de Equilíbrio Geral Aplicado à Agricultura Brasileira*, tese de doutorado, FEA-USP. 1995.
- [23] Ferris, M.C., Pang, J.S. "Engineering And Economic Applications of Complementarity Problems". *SIAM Review* 39, pp. 669-713. 1997.
- [24] Fisher, I. *Mathematical Investigations in the Theory of Value and Prices*" (1892), reedição: Yale University Press, 1961
- [25] Guilhoto, J. J. M., *Um Modelo Computável de Equilíbrio Geral para Planejamento e Análise de Políticas Agrícolas (PAPA) na Economia Brasileira*, tese de livre docência, ESALQ-USP. 1995.

- [26] Ginsburgh, V. e Robinson, S. "Equilibrium and prices in multisector models". in Syrquin, M., Taylor, L. e Westphal, I.e., eds., *Economic structure and performance*. New York, Academic Press. 1984.
- [27] Gomes-Ruggiero, Marcia A. , . *Métodos Quase-Newton para Resolução de Sistemas Não Lineares Esparsos de Grande Porte*, tese de doutorado, Depto. Matemática Aplicada, UNICAMP, Campinas, Brasil. 1990.
- [28] Gropp, Reint, E. (1995), 'Corporate Taxation and Capital Structure Choice in Germany: A General Equilibrium Model', 52 *Finanzarchiv*, 196-121
- [29] Haddad, E.A. Domingues, E.P., *EFES – Um modelo aplicado de equilíbrio geral para a economia brasileira: projeções setoriais para 1999-2004*. São Paulo: Universidade de São Paulo. (mimeo). 2000.
- [30] Harrigan, F., "Software Reviews: Software for Solving Numerical General Equilibrium Models.", *The Economic Journal*, vol.103, pp1088-1104. 1993.
- [31] Harrison, W. J. and Pearson, K. R. "Computing Solutions for Large General Equilibrium Models using Gempack" CoPS/Impact Preliminary Working Paper No. IP-64, June 1994.
- [32] Harrison, W. J., Pearson, K. R. and Powell, A. A. "Multiregional and Intertemporal AGE Modelling via Gempack" CoPS/Impact Preliminary Working Paper No. IP-66, November 1995.
- [33] Hertel, T. H., Ianchovichina, E., McDonald, B. J., "Multi-Region General Equilibrium Modeling". in Francois, J. F. e Reinert, K. A., editors, *Applied Methods For Trade Policy Analysis*. Cambridge U. Press. 1997.
- [34] Hertel, T. H., Tsigas, M. E. "Structure of GTAP". in Hertel, T. W., editor, *Global Trade Analysis Modelling and applications*. Cambridge U. Press. 1997.
- [35] Johansen, L. (1960). *A Multi-Sectoral Study Of Economic Growth*. North Holland/American Elsevier, Second Enlarge Edition, 1974.
- [36] Leontief, Wassily. "Quantitative Input-Output Relations in the Economic System of the United States", *Review of Economics and Statistics* 18, no. 3, August, 1936.
- [37] Lemke, C. E., "Bimatrix Equilibrium Points and Mathematical Programming", *Management Science* , vol. 11, nro. 7, May, 1965.

- [38] Lysy, F. J., Taylor, L., "The general equilibrium income distribution model". in Taylor, L. Bacha, E., Cardoso, E., Lysy, F. J., eds., *Models of growth and distribution for Brazil*. London, Oxford University Press. 1980.
- [39] Manne, A. S., "Multi-sector models for development planning: A survey". *Journal of Development Economics* 1 nro. 1:43-69. 1974.
- [40] Manne, A. S., "On the formulation and solution of economic equilibrium models". *Mathematical Programming Studies* 23 1-21. North Holland. 1985.
- [41] Martínez, J. M., A quasi-Newton method with modification of one column per iteration, *Computing*, 33, pp 353-362. 1984.
- [42] Mathiesen, L., "Computacional Experience in Solving Equilibrium Models by a Sequence of Linear Complementarity Problems", *Operations Research*, vol. 33, nro. 6, Nov-Dec. 1985.
- [43] Miller, Ronald E., Blair, Peter D.. *Input-Output Analysis: Foundations and extensions*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc. 1985.
- [44] Murtaugh, B.A., e Saunders, M.A.. *minos 5.0 user's guide. Technical Report Technical Report 83-20*, Systems Optimization Laboratory, Stanford University, 1983.
- [45] Murty, K. G., *Linear Complementarity, Linear and Nonlinear Programming*, Heldermann, Berlim. 1988.
- [46] Najberg, S., F. J. Z. Rigolon, S. P. Vieira, "Modelo de Equilíbrio Geral Computável como Instrumento de Política Econômica: Uma Análise de Câmbio X Tarifas". *Textos para Discussão n. 30*. Rio de Janeiro: BNDES. 1995.
- [47] Powell, M. J. D., "A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations". in Rabinowitz, P., ed., *Numerical Methods for Non-Linear Algebraic Equations*. London: Gordon & Breach. 1970.
- [48] Proença, E. P., "Análise de Equilíbrio Geral Aplicada a Economias Distorcidas", tese de mestrado, Depto. Matemática Aplicada, UNICAMP, Campinas, Brasil. 1991.
- [49] Quesnay, François; *Tableau Économique*, 1758.
- [50] Robinson, S. "Multisectoral models". in Chenery, H., Snirivasan, T.N., *Handbook of Development Economics*. Vol II Elsevier Science Publishers. 1989.
- [51] Rotondo, W., "Cálculo de Equilíbrio Econômico por Complementaridade", tese de mestrado, Depto. Matemática Aplicada, UNICAMP. 1988.

- [52] Rutherford, T. F., "A Modelling System for Applied General Equilibrium Analysis", Cowles Foundation Discussion Paper no. 836. 1987.
- [53] Rutherford, T.F., "General Equilibrium Modeling with *MPSGE*", The University of Western Ontario. 1989.
- [54] Rutherford, T. F. "Applied General Equilibrium Modeling with *MPSGE* as a *GAMS* Subsystem: An Overview of the Modeling Framework and Syntax", Department of Economics University of Colorado November, 1995
- [55] Samuelson, P. A., "Spatial price equilibrium and linear programming", *The American Economic Review*, XLII, june. 1952.
- [56] Scaramucci, José A., "Análise Aplicada de equilíbrio econômico com *Pegasus e Mathematica* Tese de Livre Docência, Depto. Matemática Aplicada, UNICAMP, Campinas, Brasil. 1997.
- [57] Scarf, Herbert E. "On the Computation of Equilibrium Prices." In *Ten Economic Studies in the Tradition of Irving Fisher*, John Wiley, 1967.
- [58] Scarf, Herbert E.; *The computation of economic equilibria*, com a colaboração de Terje Hansen.; New Haven, Yale University Press, 1973. 249 p.
- [59] Scarf, H.E. "A Computation of General Equilibrium." In *General Equilibrium*, eds. John Eatwell, Murray Milgate, Peter Newman, The New Palgrave: A Dictionary of Economics, London: The Macmillan Press Ltd, 1989, pp. 84-97
- [60] Shoven, J. e Whalley, J., "On the Computation of Competitive Equilibria on International Markets with Tariffs", *Journal of International Economics*, vol 4 pp 341-354. 1974.
- [61] Shoven, J. e Whalley, J. *Applying General Equilibrium*, Cambridge Surveys of Economic Literature, Cambridge University Press. 1992.
- [62] Taylor, L., "Theoretical Foundations and Technical Implications". in Blitzer, C., Clark, P. e Taylor, L., eds., *Economy -Wide models and Development Planning*. Oxford, Oxford University Press. 1975.
- [63] Whalley, J., "The U. K. Tax System 1978-1970: Some Fixed Point Indicators of It's Economic Impact", *Econometrica*, vol 45 nro 8 pp 1837-58. 1977.
- [64] Walras, Léon; *Éléments d'Économie Politique Pure*, Lausanne: L. Corbaz. 1874, 1877.

- [65] Wirth, N., *Algorithms + Data Structures = Programs*, Englewood Cliffs: Prentice-Hall, Inc. 1976.
- [66] Wolfram, S. J., *Mathematica, a System for Doing Mathematics by Computer*, California, Redwood City: Addison-Wesley. 1991.
- [67] York, Harold L. (1993), 'An Applied General Equilibrium Model of International Tax Competition among the Group of Seven Countries', 15(56) *Journal of Policy Modeling*, 653-672.
- [68] Zullo, Jurandir, Júnior. "Cálculo de Equilíbrios Econômicos por Otimização", tese de mestrado, Depto. Matemática Aplicada, UNICAMP. 1990.

7. Apêndices

A1 – O armazenamento simbólico de expressões analíticas

Os pacotes para resolução de sistemas de equações não-lineares geralmente são compostos por dois conjuntos de procedimentos ou sub-rotinas: um conjunto, fornecido pelo autor do pacote, que implementa um ou mais algoritmos de resolução, e outro conjunto, fornecido pelo usuário, que implementa a função vetorial equivalente ao sistema e, eventualmente, o jacobiano dessa função. Para um determinado sistema de equações que se deseja resolver, geralmente é necessária a elaboração e compilação desse segundo conjunto de procedimentos na linguagem de programação em que o pacote é implementado, gerando-se então com os dois conjuntos um módulo executável exclusivo para esse sistema.

Outra possibilidade é implementar no pacote a capacidade de avaliar as expressões analíticas que compõem as equações de um sistema qualquer, que poderiam ser lidas pelo pacote de um arquivo texto, por exemplo. Para isso são necessários um *banco de dados* na memória de trabalho do pacote para armazenar em forma simbólica as expressões, e dois procedimentos: um procedimento *montador* que leia, interprete e monte as expressões do arquivo texto no banco de dados, de forma adequada para que um procedimento *avaliador* possa efetuar a avaliação das expressões de maneira eficiente sempre que necessário.

As noções básicas para se implementar a capacidade de avaliar expressões em um pacote serão apresentadas a seguir através de um exemplo didático muito simples. As técnicas e conceitos aqui apresentados de forma simplificada podem ser vistos em detalhes

em Wirth [65]. Observe-se que embora uma implementação eficiente do armazenamento simbólico e avaliação de expressões utilize um banco de dados com mais informações, envolvendo alocação dinâmica de memória, apontadores, listas ligadas e pesquisa binária, e os procedimentos montador e avaliador sejam mais complexos que o indicado nesse exemplo, o método é basicamente o mesmo deste exemplo.

Por convenção, o posicionamento dos operadores em uma expressão tem o formato *infix*, onde os operadores são posicionados entre os operandos: por exemplo em $a+b$ o operador "+" é posicionado entre os operandos "a" e "b", e a regra de avaliação associada é "percorra a expressão até chegar ao segundo operando e então efetue a operação indicada entre os dois operandos".

O formato mais adequado para armazenamento em computador é o formato *posfix*, onde o operador é posicionado após os operandos: $ab+$. A regra de avaliação associada é "percorra a expressão até chegar a um operador e então efetue a operação indicada com o número de operandos anteriores necessários à operação".

Uma das vantagens desse formato é dispensar o uso de parênteses para indicar a ordem em que as operações devem ser realizadas: por exemplo, a expressão $(a+b)*(c+d)$ teria a forma $ab+cd+*$ no formato *posfix*, e sua avaliação seria feita com o auxílio de variáveis temporárias t_1 , t_2 e t_3 com a regra de avaliação enunciada acima para esse formato: $ab+cd+* \rightarrow t_1cd+* \rightarrow t_1t_2* \rightarrow t_3$.

O procedimento montador deve ler uma expressão *infix* reconhecendo operandos (variáveis e constantes numéricas) e operadores, armazenando no banco de dados os operandos e uma lista simbólica correspondente ao formato *posfix* da expressão.

A lista simbólica é formada por *endereços dos operandos* no banco de dados e por *códigos dos operadores*, estes últimos obtidos de uma tabela de operadores, onde funções como LOG, SIN, COS, SQRT são consideradas um tipo especial de operador de apenas um operando.

Considere-se que os códigos dos operadores sejam dados pela tabela a seguir:

<i>código</i>	1	2	3	4	5	6	7	8	9	17	...
<i>operador</i>	+	-	*	/	=	:=	SQRT	...
<i>núm operandos</i>	2	2	2	2	1	1	1	...

Considere-se um banco de dados simplificado, com endereço, nome, valor e apontador para a lista simbólica. Considere-se que a expressão a ser analisada seja:

$$y := 2,5 * (a + \text{SQRT}(x)) \quad (\text{infix fornecido})$$

$$y \ 2,5 \ a \ x \ \text{SQRT} \ + \ * \ := \quad (\text{posfix obtido pelo montador})$$

A seguir, as informações colocadas no banco de dados pelo procedimento montador ao interpretar essa expressão. Observe-se que nem todos operandos possuem valor inicialmente, e somente *y* possui lista simbólica, determinada pelo operador ":="

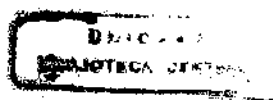
<i>endereço</i>	1	2	3	4	5	6	7
<i>nome</i>	"2,5"	<i>a</i>	<i>x</i>	<i>y</i>
<i>valor</i>	2,5	-	-	-
<i>lista simbólica</i>	-	-	-	<i>lista4</i>

Supondo que sejam usados valores negativos para identificar os operadores, a lista simbólica de endereços de operandos e códigos de operadores construída será:

	<i>y</i>	"2,5"	<i>a</i>	<i>x</i>	SQRT	+	*	:=	FIM
	↑	↑	↑	↑	↑	↑	↑	↑	↑
<i>lista4</i>	4	1	2	3	-17	-1	-3	-9	0

Para finalizar, o procedimento avaliador utilizará uma pilha auxiliar de valores do tipo LIFO (*Last In First Out*) para calcular o valor de *y* por meio do algoritmo a seguir¹¹:

¹¹ Supõe-se que de alguma maneira as variáveis *a* e *x* já tenham recebido valores



Passo 1) $i = 1$
Passo 2) $k = lista4(i)$
 Se $k > 0$ então:
 acrescentar *valor*(k) à pilha auxiliar
 Senão: Se $k < 0$ então:
 fazer $n = num_operandos(-k)$
 executar operação $-k$ com n valores da pilha auxiliar
 retirar n valores da pilha auxiliar
 acrescentar resultado da operação à pilha auxiliar
 Senão: terminar algoritmo
Passo 3) $i = i + 1$
 Voltar ao passo 2)

A2 – O método Johansen/Gempack

O Gempack é um sistema computacional para a implementação e resolução de modelos de equilíbrio econômico geral e parcial, podendo tratar um conjunto abrangente de comportamentos econômicos. O sistema pode ser instalado em diversos tipos de computadores e oferece também um conjunto de programas utilitários para a manipulação do banco de dados econômicos e dos resultados das simulações.

No Gempack a implementação do modelo econômico é feita a partir de uma representação algébrica das equações do modelo. Essa linguagem algébrica não permite explicitamente desigualdades no sistema a ser resolvido, embora várias espécies de comportamento de otimização, cuja formalização seja complicada ou não conhecida analiticamente, possam ser representadas diretamente através das condições de primeira ordem do problema. As equações podem ser dadas em sua forma não-linear ou na forma linearizada – que é o padrão do pacote –, ou até mesmo em uma mistura das duas formas. Uma característica única do pacote é a condensação algébrica automatizada de equações, que torna possível a resolução de modelos muito maiores do que normalmente seria praticável.

O método de resolução utilizado no pacote será descrito a seguir com base em Harrison e Pearson [31] e Dahlquist e Björk [10].

Na representação matemática do modelo econômico supõe-se que as interações entre os agentes da economia podem ser descritas pelo sistema de equações:

$$g_i(y_1, \dots, y_{m+n}) = 0, \quad \text{para } i = 1, \dots, m,$$

onde g_1, \dots, g_m são funções das $m+n$ variáveis y_1, \dots, y_{m+n} . Para resolver esse sistema, n das variáveis precisam ser especificadas exogenamente (fixadas como constantes) deixando as outras m endógenas. Usando-se x_1, \dots, x_n para indicar o conjunto de variáveis exógenas e z_1, \dots, z_m para indicar as endógenas, as equações do sistema podem ser rescritas como

$$g_i(z_1, \dots, z_m, x_1, \dots, x_n) = 0, \quad \text{para } i = 1, \dots, m. \quad (22)$$

Observe-se que assim as variáveis z_1, \dots, z_m serão determinadas *implicitamente* por $g_i(z_1, \dots, z_m, x_1, \dots, x_n)$ como função dos valores das variáveis x_1, \dots, x_n . Na prática, embora as funções g_1, \dots, g_m sejam conhecidas (no sentido que suas fórmulas são dadas), a forma explícita de z_i em função de x_1, \dots, x_n não é conhecida; existe apenas a garantia de sua existência. Utilizando-se a representação vetorial (indicada em negrito), com $\mathbf{z} = \{z_1, \dots, z_m\}$, $\mathbf{x} = \{x_1, \dots, x_n\}$ e $\mathbf{g}(\mathbf{z}, \mathbf{x}) = \{g_1(\mathbf{z}, \mathbf{x}), \dots, g_m(\mathbf{z}, \mathbf{x})\}$, tem-se

$$\mathbf{g}(\mathbf{z}, \mathbf{x}) = \mathbf{0}, \quad (23)$$

onde \mathbf{g} é uma função vetorial do \mathbf{R}^{m+n} no \mathbf{R}^m . Assume-se que as primeiras derivadas parciais $\partial g_i / \partial z_j$ e $\partial g_i / \partial x_k$ existem e são funções contínuas em cada ponto no domínio pertinente.

Os modelos computáveis de equilíbrio econômico geralmente são utilizados em *problemas de simulação*. Em um problema de simulação se deseja, a partir de uma solução de equilíbrio conhecida do modelo (por exemplo $\mathbf{z} = \mathbf{z}_E$ e $\mathbf{x} = \mathbf{x}_E$), conhecer os novos valores \mathbf{z}_S das variáveis endógenas quando os valores das variáveis exógenas mudam de \mathbf{x}_E para \mathbf{x}_S , ou seja,

$$\left. \begin{array}{l} \text{dado que } \mathbf{z} = \mathbf{z}_E \text{ quando } \mathbf{x} = \mathbf{x}_E \text{ é uma solução de (23),} \\ \text{achar o valor } \mathbf{z}_S \text{ de } \mathbf{z} \text{ quando } \mathbf{x} = \mathbf{x}_S. \end{array} \right\} \quad (24)$$

Para a resolução desse problema considere-se a introdução de uma variável escalar v (um número real) e que as variáveis exógenas \mathbf{x} sejam dadas como uma função dessa nova variável v por

$$\mathbf{x} = \mathbf{x}_E + v(\mathbf{x}_S - \mathbf{x}_E). \quad (25)$$

Note-se que $\mathbf{x} = \mathbf{x}_E$ quando $v = 0$ e $\mathbf{x} = \mathbf{x}_S$ quando $v = 1$, e que \mathbf{x} se move ao longo de uma linha reta (em n dimensões) ligando \mathbf{x}_E a \mathbf{x}_S quando v varia de 0 a 1.

Como z é uma função (implícita) de \mathbf{x} , e \mathbf{x} é claramente uma função diferenciável de v , então z também é uma função diferenciável de v , e pela regra da cadeia tem-se

$$\frac{dz_i}{dv} = \sum_{j=1}^n \frac{\partial z_i}{\partial x_j} \frac{dx_j}{dv}$$

Por hipótese as derivadas parciais $\partial g_i / \partial z_j$ e $\partial g_i / \partial x_k$ existem e são funções contínuas de z e \mathbf{x} , e viu-se acima que cada z_i e x_k também são funções diferenciáveis de v . Então a regra da cadeia pode ser novamente usada para diferenciar as equações $g_i(z_1, \dots, z_m, x_1, \dots, x_n) = 0$ em relação a v , obtendo-se

$$\sum_{j=1}^m \frac{\partial g_i}{\partial z_j} \frac{dz_j}{dv} + \sum_{k=1}^n \frac{\partial g_i}{\partial x_k} \frac{dx_k}{dv} = 0, \quad \text{para } i=1, \dots, m \quad (26)$$

Da definição de \mathbf{x} em função de v tem-se que

$$\frac{dx_k}{dv} = x_{S_k} - x_{E_k}, \quad \text{para } k=1, \dots, n$$

Substituindo dx_k/dv em (26) e expressando-se esse sistema na forma matricial, tem-se:

$$\begin{pmatrix} \frac{\partial g_1}{\partial z_1} & \dots & \frac{\partial g_1}{\partial z_m} \\ \vdots & \dots & \vdots \\ \frac{\partial g_m}{\partial z_1} & \dots & \frac{\partial g_m}{\partial z_m} \end{pmatrix} \begin{pmatrix} \frac{dz_1}{dv} \\ \vdots \\ \frac{dz_m}{dv} \end{pmatrix} = - \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \dots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \dots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \dots & \frac{\partial g_m}{\partial x_n} \end{pmatrix} \begin{pmatrix} x_{S_1} - x_{E_1} \\ \vdots \\ x_{S_n} - x_{E_n} \end{pmatrix}$$

Ou seja,

$$\mathbf{A}(\mathbf{z}, \mathbf{x}) \frac{d\mathbf{z}}{dv} = \mathbf{b}(\mathbf{z}, \mathbf{x}),$$

onde:

$$\mathbf{x} = \mathbf{x}_E + v (\mathbf{x}_S - \mathbf{x}_E),$$

$\mathbf{A}(\mathbf{z}, \mathbf{x})$ é a matriz $m \times m$ cujo elemento (i,j) é a derivada $\partial g_i / \partial z_j$ avaliada em (\mathbf{z}, \mathbf{x}) ,

$d\mathbf{z}/dv$ é o vetor $m \times 1$ cujo j -ésimo elemento é dz_j/dv e

$\mathbf{b}(\mathbf{z}, \mathbf{x})$ é o vetor $m \times 1$ obtido multiplicando-se a matriz $m \times n$ cujo elemento (i,k) é a derivada $\partial g_i / \partial x_k$ avaliada em (\mathbf{z}, \mathbf{x}) pelo vetor $n \times 1$ cujo k -ésimo elemento é $x_{S_k} - x_{E_k}$.

Observe-se que em decorrência da definição de \mathbf{x} em função de v tem-se

$$\mathbf{A}(\mathbf{z}, \mathbf{x}) \rightarrow \mathbf{A}(\mathbf{z}, \mathbf{x}_E + v (\mathbf{x}_S - \mathbf{x}_E)) \rightarrow \mathbf{A}(\mathbf{z}, v),$$

$$\mathbf{b}(\mathbf{z}, \mathbf{x}) \rightarrow \mathbf{b}(\mathbf{z}, \mathbf{x}_E + v (\mathbf{x}_S - \mathbf{x}_E)) \rightarrow \mathbf{b}(\mathbf{z}, v).$$

E então, para todo v pertinente,

$$\mathbf{A}(\mathbf{z}, v) \frac{d\mathbf{z}}{dv} = \mathbf{b}(\mathbf{z}, v)$$

Ou,

$$\frac{d\mathbf{z}}{dv} = \mathbf{A}^{-1}(\mathbf{z}, v) \mathbf{b}(\mathbf{z}, v), \quad (27)$$

Na prática, parece ser o caso de a matriz $\mathbf{A}(\mathbf{z}, v)$ ter inversa para os valores pertinentes de \mathbf{z} e v em um fechamento válido num modelo bem especificado. Não se conhecem resultados teóricos que garantam a existência da inversa dessa matriz; entretanto, os únicos exemplos conhecidos em que $\mathbf{A}(\mathbf{z}, v)$ não tem inversa são casos em que $\mathbf{g}(\mathbf{z}, \mathbf{x}) = \mathbf{0}$ não tem uma solução única para \mathbf{z} em termos de \mathbf{x} em uma dada vizinhança. No Gempack a inversão da matriz \mathbf{A} é feita pelo pacote de fatorização de matrizes esparsas Harwell-MA28.

O sistema (27) é um sistema de equações diferenciais ordinárias de primeira ordem cuja solução é dada por $\mathbf{z}(v)$, ou seja, \mathbf{z} em função de v através de $\mathbf{g}(\mathbf{z}, \mathbf{x}) = \mathbf{0}$ (implicitamente) e de $\mathbf{x}(v) = \mathbf{x}_E + v (\mathbf{x}_S - \mathbf{x}_E)$.

De (1.3) tem-se que a condição inicial é $z(0) = z_E$, pois

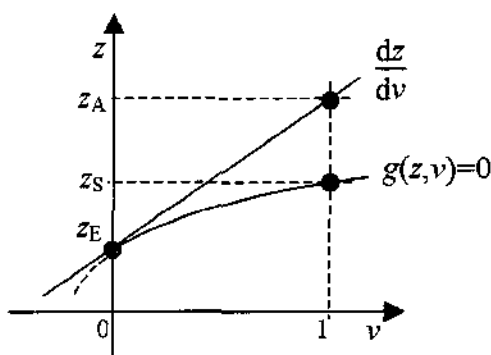
$$z(x(0)) = z(x_E) = z_E$$

Assim, o problema de simulação (24) foi convertido no seguinte problema:

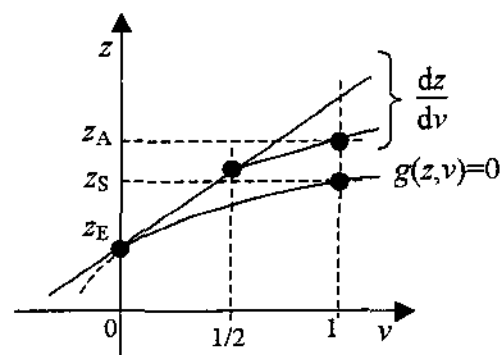
$$\left. \begin{array}{l} \text{Dado que } \frac{dz}{dv} = A^{-1}(z, v) b(z, v), \\ \text{com a condição inicial } z(0) = z_E, \\ \text{achar } z_S, \text{ o valor de } z(v) \text{ quando } v = 1 \end{array} \right\} \quad (28)$$

O problema (2.3) é um exemplo de um "Problema de Valor Inicial para Equações Diferenciais Ordinárias"; esses problemas são bem conhecidos, ocorrendo em diversas áreas e são extensivamente estudados. Caso fosse possível obter a forma explícita de $z(v)$, a obtenção de z_S seria muito simples; entretanto, como isso não é possível, será necessário obter uma aproximação numérica para $z(v)$, calculando-se dz/dv por diferenças finitas.

Nas figuras abaixo são ilustrados os métodos Johansen e Euler para a obtenção de z_A , a aproximação numérica de z_S , em um exemplo com apenas uma dimensão. A diferença básica entre os métodos é que no de Euler o intervalo $(0,1)$ é dividido em subintervalos de tamanho h ($h=1/2$ na figura), sendo dz/dv reavaliada em cada subintervalo.



a) Método Johansen



b) Método Euler

Observe-se que em a) tem-se $z_A \gg z_S$, devendo esse método ser utilizado somente quando se deseja apenas ter uma idéia geral da direção das mudanças em uma simulação; em b) é possível atingir o nível de precisão desejado diminuindo-se o valor de h .

No método de Euler, fazendo-se $v_k = h k$, e substituindo-se dz/dv pelo quociente de diferenças $\frac{z_{k+1} - z_k}{h}$, tem-se:

$$\frac{z_{k+1} - z_k}{h} \approx \mathbf{A}^{-1}(\mathbf{z}, v_k) \mathbf{b}(\mathbf{z}, v_k), \quad \text{com } \mathbf{z}_0 = \mathbf{z}_E$$

obtendo-se assim a fórmula recursiva para o cálculo de \mathbf{z} :

$$\mathbf{z}_{k+1} = \mathbf{z}_k + h \mathbf{A}^{-1}(\mathbf{z}, v_k) \mathbf{b}(\mathbf{z}, v_k), \quad \text{com } \mathbf{z}_0 = \mathbf{z}_E$$

Entretanto, a convergência do método de Euler para a solução verdadeira é lenta e cara em termos computacionais quando $h \rightarrow 0$. Na prática é utilizada a *extrapolação de Richardson*, onde os valores para \mathbf{z}_s obtidos em duas ou mais repetições do método de Euler com um número relativamente pequeno de subintervalos (por exemplo: 2, 4, 6), escolhidos de maneira que h seja decrescente em cada repetição, são extrapolados para $h = 0$ (ver [10]).

A precisão pode ser aumentada utilizando-se o *método do ponto médio*, uma modificação do método de Euler, que utiliza para o cálculo de dz/dv o quociente de diferenças centrado no ponto médio:

$$\frac{dz}{dv} \approx \frac{z_{k+1} - z_{k-1}}{2h}, \quad \text{com } \mathbf{z}_0 = \mathbf{z}_E \text{ e } \mathbf{z}_1 \text{ calculado como no método de Euler.}$$

Esse método introduz uma pequena oscilação no valor calculado de \mathbf{z}_k que aumenta com o valor de k . No *método do ponto médio modificado* ou *método de Gragg*, essa oscilação é amortecida com uma fórmula especial de cálculo de \mathbf{z}_k aplicada nos valores pares de k (ver [10]).

Para concluir reproduz-se as recomendações de Harrison e Pearson [31]:

- "1. O método de Gragg é geralmente mais preciso que o de Euler.
2. Quando em dúvida, a extrapolação deve ser aplicada.

3. *Extrapolar de três diferentes repetições é melhor do que de duas. (Por exemplo, extrapolar da solução Gragg de 2, 4 e 6 subintervalos é melhor do que da solução de apenas 4 e 6 subintervalos.)*

A3 – O algoritmo SLCP, para resolução de $f(\mathbf{z}) \geq 0 : \mathbf{z} \geq 0$

Para apresentar o método de resolução é necessário fazer uma redefinição dos nomes das variáveis e funções que compõem das equações de excesso de oferta e lucro negativo do sistema (12), que ficará com um total de $n = n_b + n_s$ equações.

Assim, seja o vetor $\mathbf{z} = \{z_1, \dots, z_n\}$, com n elementos, tal que:

$$\begin{cases} z_i = p_i, & \text{para } i = 1, \dots, n_b \\ z_{n_b+j} = y_j, & \text{para } j = 1, \dots, n_s \end{cases}$$

Substituindo os vetores \mathbf{p} e \mathbf{y} pelo vetor \mathbf{z} em (8) e (11), tem-se:

$$\begin{cases} ES_i(\mathbf{z}) = ES_i(\mathbf{p}), & \text{para } i = 1, \dots, n_b \\ NP_j(\mathbf{z}) = NP_j(\mathbf{p}), & \text{para } j = 1, \dots, n_s \end{cases}$$

E então, seja a função $\mathbf{f}(\mathbf{z}) = \{f_1(\mathbf{z}), \dots, f_n(\mathbf{z})\}$, com n elementos, tal que:

$$\begin{cases} f_i(\mathbf{z}) = ES_i(\mathbf{z}), & \text{para } i = 1, \dots, n_b \\ f_{n_b+j}(\mathbf{z}) = NP_j(\mathbf{z}), & \text{para } j = 1, \dots, n_s \end{cases}$$

Dessa maneira, os sistemas (8) e (11) são representados pelo sistema:

$$\left. \begin{array}{l} f_i(\mathbf{z}) \geq 0 \\ z_i \geq 0 \\ z_i f_i(\mathbf{z}) = 0 \end{array} \right\} \text{ para } i = 1, \dots, n$$

Ou, usando a notação vetorial, o problema será:

$$\left. \begin{array}{l} \mathbf{f}(\mathbf{z}) \geq \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ \mathbf{z}^T \mathbf{f}(\mathbf{z}) = 0 \end{array} \right\} \text{ ou } \mathbf{f}(\mathbf{z}) \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0} \quad (12'')$$

A resolução do sistema (12'') é feita através de um algoritmo baseado no método de Newton, conhecida como algoritmo SLCP (Seqüência de Problemas de Complemen-

taridade Linear) [42], em um processo iterativo onde cada iteração é composta por dois passos básicos:

1) A partir de uma aproximação existente da solução, obtém-se uma nova aproximação da solução através da linearização do sistema no ponto conhecido.

2) Se a nova aproximação satisfizer um dado critério de proximidade da solução, será considerada a solução do problema e encerra-se o processo iterativo; caso contrário a nova aproximação deverá ser utilizada como ponto de partida para uma nova iteração.

Em um método de Newton, o primeiro passo é realizado resolvendo-se o sistema linearizado. Entretanto, no caso de um problema de complementaridade não-linear, a linearização de $f(\mathbf{z})$ resulta em um problema de complementaridade *linear*, onde a matriz jacobiana de $f(\mathbf{z})$ tem posto incompleto no caso mais geral, e esse problema deve ser resolvido por um método que leve esse fato em consideração, como por exemplo o método de Lemke (ver apêndice A4), que se assemelha ao método simplex de programação linear. Assim, a resolução do problema de complementaridade *não-linear* é feita através da resolução de uma *seqüência* de problemas de complementaridade *linear*.

Para a linearização do problema (18) considere-se a aproximação de Taylor de primeira ordem de $f(\mathbf{z})$ em torno de um ponto \mathbf{z}_0 dado:

$$\mathbf{f}(\mathbf{z}) \cong \mathbf{f}(\mathbf{z}_0) + \mathbf{J}(\mathbf{z}_0) (\mathbf{z} - \mathbf{z}_0),$$

onde \mathbf{z}_0 é uma aproximação conhecida da solução e $\mathbf{J}(\mathbf{z}_0)$ é a matriz $n \times n$ composta pelas derivadas parciais de $f(\mathbf{z})$ calculadas em \mathbf{z}_0 – a matriz jacobiana de $f(\mathbf{z})$ em \mathbf{z}_0 :

$$\mathbf{J}(\mathbf{z}_0) = \begin{bmatrix} \left. \frac{\partial f_1(\mathbf{z})}{\partial z_1} \right|_{\mathbf{z}_0} & \cdots & \left. \frac{\partial f_1(\mathbf{z})}{\partial z_n} \right|_{\mathbf{z}_0} \\ \cdots & \cdots & \cdots \\ \left. \frac{\partial f_n(\mathbf{z})}{\partial z_1} \right|_{\mathbf{z}_0} & \cdots & \left. \frac{\partial f_n(\mathbf{z})}{\partial z_n} \right|_{\mathbf{z}_0} \end{bmatrix}$$

Rearranjando a expressão (19) para agrupar os termos constantes:

$$\mathbf{f}(\mathbf{z}) \cong \mathbf{J}(\mathbf{z}_0) \mathbf{z} + (\mathbf{f}(\mathbf{z}_0) - \mathbf{J}(\mathbf{z}_0) \mathbf{z}_0)$$

Fazendo-se $\mathbf{q} = (\mathbf{f}(\mathbf{z}_0) - \mathbf{J}(\mathbf{z}_0) \mathbf{z}_0)$ e chamando $\mathbf{J}(\mathbf{z}_0)$ de \mathbf{M} para ressaltar o fato de serem respectivamente um vetor e uma matriz de elementos constantes, tem-se:

$$\mathbf{f}(\mathbf{z}) \cong \mathbf{M} \mathbf{z} + \mathbf{q} \quad (29)$$

E substituindo-se (19) em (18''):

$$\left. \begin{array}{l} \mathbf{M} \mathbf{z} + \mathbf{q} \geq \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ \mathbf{z}^T (\mathbf{M} \mathbf{z} + \mathbf{q}) = 0 \end{array} \right\} \text{ ou } \mathbf{M} \mathbf{z} + \mathbf{q} \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0} \quad (30)$$

O sistema (30) é um problema de complementaridade *linear*, e deve ser resolvido pelo algoritmo LCP, uma implementação do método de Lemke [53]. É possível que, mesmo existindo uma solução para o sistema (30), o método de Lemke não tenha sucesso na busca de uma solução, e nesse caso o processo iterativo deve ser terminado.

No segundo passo, após a aplicação com sucesso do método de Lemke, o vetor \mathbf{z} conterà a nova aproximação da solução, e a condição de complementaridade é utilizada como critério de proximidade da solução, verificando-se se $\mathbf{z}^T \mathbf{f}(\mathbf{z}) < \varepsilon$, onde ε é uma constante positiva tão próxima de zero quanto se queira. Caso esse critério não seja satisfeito, retorna-se ao primeiro passo utilizando \mathbf{z} no lugar de \mathbf{z}_0 . Caso o critério seja satisfeito, o vetor \mathbf{z} é a solução do problema e encerra-se o processo iterativo.

Assim, a resolução do problema de complementaridade *não-linear* $\mathbf{f}(\mathbf{z}) \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0}$ é feita através da resolução de uma *seqüência* de problemas de complementaridade *linear* da forma $\mathbf{M}_k \mathbf{z} + \mathbf{q}_k \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0}$, onde o índice k é acrescentado à \mathbf{M} e \mathbf{q} para indicar que foram obtidos na iteração k do processo iterativo, como resultado da linearização de $\mathbf{f}(\mathbf{z})$ em \mathbf{z}_{k-1} , sendo o processo iniciado com \mathbf{z}_0 , uma aproximação da solução conhecida.

O momento mais adequado para eliminar a equação correspondente ao numerário é ao calcular \mathbf{M}_k e \mathbf{q}_k , possibilitando a escolha de outro bem como numerário. A cada escolha de numerário corresponde um PCNL diferente. Como o algoritmo de Lemke pode não convergir mesmo quando existe uma solução, pode ocorrer eventualmente que um PCNL determinado por um numerário possa ser resolvido pelo algoritmo SLCP enquanto que outro PCNL possa não ser. (Note-se que com a eliminação da equação do numerário, \mathbf{M}_k e \mathbf{q}_k terão dimensão $n-1$.)

Algoritmo SLCP:

Dados a função $\mathbf{f}(\mathbf{z}): \mathbf{R}^n \rightarrow \mathbf{R}^n$, a aproximação inicial da solução \mathbf{z}_0 , a aproximação ε desejada para a solução e o bem a ser utilizado como numerário, fazer:

Passo 0 – Inicialização:

$$k = 0$$

Passo 1 – Obtenção da nova aproximação da solução:

$$k = k + 1$$

Passo 1.1 – Linearização de $\mathbf{f}(\mathbf{z})$ no ponto \mathbf{z}_{k-1} , obtendo \mathbf{M}_k e \mathbf{q}_k com dimensão $n-1$, eliminando a equação correspondente ao numerário, considerando unitário o preço desse bem.

Passo 1.2 – Resolução de $\mathbf{M}_k \mathbf{z} + \mathbf{q}_k \geq \mathbf{0} : \mathbf{z} \geq \mathbf{0}$ usando o método de Lemke

Passo 1.3 – Se não foi possível encontrar solução, então:

Se existirem outros bens a serem testados como numerário, então:

Utilizar outro bem como numerário;

Voltar para o passo 1.1.

Senão (caso todos os bens já tenham sido testados):

Encerrar o processo, não é possível encontrar solução.

Senão (caso tenha sido encontrada solução):

Inserir o valor 1 como preço do numerário no vetor \mathbf{z} .

Passo 2 – Teste de proximidade da solução encontrada na iteração k :

Se $\mathbf{z}^T \mathbf{f}(\mathbf{z}) < \varepsilon$, então:

Encerrar o processo, a solução é z ;

Senão:

Fazer $z_k = z$;

Voltar para o passo 1.

A4 – O algoritmo LCP, para resolução de $\mathbf{M}(z)+\mathbf{q} \geq \mathbf{0} : z \geq \mathbf{0}$

Diversos problemas que surgem em diferentes campos tais como programação matemática, teoria dos jogos, economia, mecânica e geometria podem ser propostos de forma unificada da seguinte maneira:

"Dados uma matriz \mathbf{M} de dimensão $n \times n$ e um vetor \mathbf{q} n -dimensional, e considerando os vetores \mathbf{z} e \mathbf{w} tal que $\mathbf{w} = \mathbf{M} \mathbf{z} + \mathbf{q}$, achar a solução para o sistema:

$$\left. \begin{array}{l} \mathbf{w} \geq \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ \mathbf{w}^T \mathbf{z} = 0 \end{array} \right\} \quad (30)$$

A condição $\mathbf{z}^T \mathbf{w} = 0$ é conhecida como restrição de complementaridade, pois para qualquer solução do sistema (30), se a variável z_j do vetor \mathbf{z} é positiva, então a variável w_j do vetor \mathbf{w} deve ser zero, e vice-versa, para $j = 1, \dots, n$. O par de variáveis (z_j, w_j) é chamado par complementar e cada variável no par é o complemento da outra. Assim, o problema acima é conhecido como um problema de complementaridade.

Considerando-se no lugar da matriz \mathbf{M} uma função $\mathbf{f}: \mathbf{R}^n \rightarrow \mathbf{R}^n$ tem-se a forma mais geral, um problema de complementaridade não-linear. Karamardian [24] apresenta diversos teoremas de existência de solução cobrindo os casos em que \mathbf{f} é não-linear e não-diferenciável, não-linear porém diferenciável e uma transformação afim. Neste último caso tem-se um problema de complementaridade linear e se pode provar a existência de solução para diversas classes de matrizes.

Ainda não foi desenvolvido nenhum algoritmo que resolva o sistema (30) para \mathbf{M} e \mathbf{q} quaisquer. Entretanto, para certas classes de matrizes, existem dois algoritmos que convergem para uma solução (ou concluem que ela não existe): o método do pivô principal,

desenvolvido por Dantzig e Cottle, e o algoritmo do pivô complementar, desenvolvido por Lemke [10].

Será analisado a seguir o algoritmo LCP (uma implementação do método de Lemke [10]), adaptado de Murty [34]. Para isso, colocaremos o sistema (30) de maneira equivalente, mais adequada ao nosso objetivo:

$$\left. \begin{array}{l} \mathbf{I} \mathbf{w} - \mathbf{M} \mathbf{z} = \mathbf{q} \\ \mathbf{w} \geq \mathbf{0} \\ \mathbf{z} \geq \mathbf{0} \\ \mathbf{w}^T \mathbf{z} = 0 \end{array} \right\} \quad (31)$$

O algoritmo tem certa semelhança com o método simplex de programação linear, trabalhando com um tablô obtido a partir de (31), formado pela matriz identidade (associada a \mathbf{w}), a matriz $-\mathbf{M}$ (associada a \mathbf{z}) e uma coluna cujos elementos são todos -1 , (o vetor $-\mathbf{e}_n$, que é associado a uma variável artificial z_0 usada para inicialização do algoritmo), num total de $2n+1$ colunas. O vetor \mathbf{q} é associado às linhas do tablô.

\mathbf{w}	\mathbf{z}	z_0	
\mathbf{I}	$-\mathbf{M}$	$-\mathbf{e}_n$	\mathbf{q}

Ou:

w_1, \dots, w_n	z_1, \dots, z_n	z_0	
1, 0, 0, ..., 0	$-m_{1,1}, \dots, -m_{1,n}$	-1	q_1
0, 1, 0, ..., 0	$-m_{2,1}, \dots, -m_{2,n}$	-1	q_2
...
0, ..., 0, 0, 1	$-m_{n,1}, \dots, -m_{n,n}$	-1	q_n

Em cada iteração do algoritmo trabalha-se com uma *base*, uma submatriz quadrada não-singular de ordem n formada pelas colunas do tablô. As variáveis associadas às colunas da base são chamadas *variáveis básicas*. As n variáveis básicas (selecionadas do conjunto

de $2n+1$ variáveis $\{w_1, \dots, w_n, z_1, \dots, z_n, z_0\}$) constituem um *vetor básico*. O vetor básico que tem exatamente uma variável de cada par complementar (z_j, w_j) é chamado *vetor complementar básico*, associado a uma *base complementar*. Se o vetor satisfizer essa condição para $n-1$ variáveis apenas (e conseqüentemente tem-se um par complementar em que as duas variáveis que o compõem são não-básicas, e além disso, z_0 deve ser básica), é chamado *vetor quase-complementar básico*, associado a uma *base quase-complementar*.

O passo fundamental de cada iteração é o *pivoteamento* da base. Para isso, são escolhidas uma variável básica, que sairá da base, e uma variável não-básica que entrará na base. Isso determina também o elemento pivô, com o qual é feita a atualização do tablô. Para solução do sistema correspondente a essa nova base, considera-se zero as variáveis não-básicas e resolve-se o sistema. Se os valores das variáveis básicas da solução obtida são não-negativos, a base é chamada *factível*. O algoritmo trabalha somente com bases factíveis complementares ou quase-complementares.

Para inicializar o algoritmo, considere-se que se $q \geq 0$, então tem-se a solução trivial $(w, z) = (q; 0)$ e termina-se o algoritmo. Senão, tem-se na matriz I uma base complementar não-factível, e entra-se com a coluna associada à variável artificial z_0 para substituir uma coluna que dever sair, obtendo uma base quase-complementar factível. Para isso, identifica-se a linha r tal que $q_r = \text{mínimo}\{q_i : 1 \leq i \leq n\}$ e usa-se como pivô o r -ésimo elemento do vetor coluna associado a z_0 . Após o pivoteamento, tem-se uma base associada ao vetor básico quase complementar $(w_1, \dots, w_{r-1}, z_0, w_{r+1}, \dots, w_n)$. Como o pivô é -1 e o vetor q inicial negativo, o vetor q atualizado (solução) será não-negativo e tem-se uma base quase complementar factível.

Na segunda iteração, como também nas seguintes, existe uma maneira única de continuar com o algoritmo, através da regra do pivô complementar, que consiste em escolher o complemento da variável que saiu na iteração anterior para ser a próxima variável a entrar na base. Isso irá garantir também a complementaridade da nova base.

A variável que deve sair será determinada através do teste da razão mínima. Seja a_{is} um elemento da coluna s do tablô que deve entrar na base e q_i um elemento do vetor q atualizado, $i = 1, \dots, n$. Determina-se a linha r tal que $\frac{q_r}{a_{rs}} = \text{mínimo}\left\{\frac{q_i}{a_{is}} : 1 \leq i \leq n\right\}$ e a r -ésima variável deverá sair da base. Isso garante que a nova base será viável. O pivô será o elemento a_{rs} .

Existem exatamente duas maneiras possíveis para término do algoritmo. Uma possibilidade é obter-se uma base complementar factível quando a variável z_0 é escolhida pelo teste da razão mínima para sair da base. A solução do tablô para essa base final corresponde à solução do problema de complementaridade. Outra possibilidade acontece quando a coluna do pivô se torna não-positiva e nesse caso o algoritmo termina em raia. Quando isso acontece, o algoritmo é incapaz de encontrar uma solução. É possível que o problema tenha solução, mas se tiver, o algoritmo não consegue encontrá-la. Entretanto, se a matriz \mathbf{M} for semi-definida positiva, pode-se provar que o término em raia só ocorre quando o problema não tem solução.

A5 – O algoritmo ACI, para resolução de $f(z) = 0$ ¹²

Nos problemas em que por construção do modelo econômico pode-se garantir que na solução se tem $z > 0$, o problema de complementaridade $f(z) \geq 0 : z \geq 0$ se reduz a $f(z) = 0$, com $z \geq 0$, e nesse caso o problema pode ser resolvido por métodos gerais que se apliquem a um sistema de equações não-lineares, como por exemplo os métodos secantes¹³.

Os métodos secantes, introduzidos inicialmente por Broyden [7], pertencem ao conjunto de métodos derivados do método de Newton que, resumidamente, é um procedimento iterativo que a partir de uma aproximação conhecida da solução, obtém uma nova aproximação corrigindo a aproximação existente utilizando informações da função e suas derivadas, calculadas na aproximação existente.

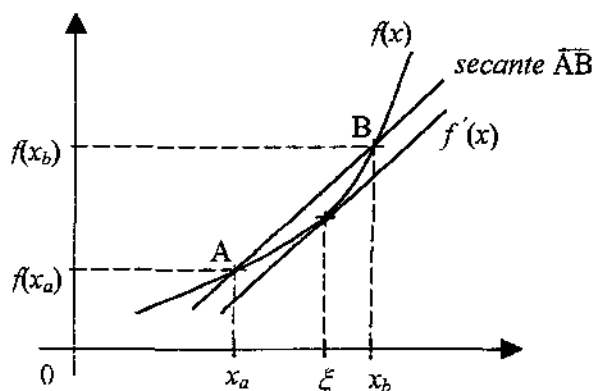
O método de Newton é largamente usado na resolução de sistemas não-lineares, e sob certas hipóteses o método converge para a solução com uma taxa quadrática. Entretanto, muitas vezes se deseja evitar ou minimizar o cálculo de derivadas, e existem diversas variações do método de Newton com esse objetivo, sendo essas variações conhecidas como métodos *quase-newtonianos* ou, para usar terminologia já consagrada, "quase-Newton".

Um resultado que será utilizado é o teorema do valor médio, que tem o seguinte enunciado: se $f(x)$ é contínua em $[x_a, x_b]$ e diferenciável em (x_a, x_b) , então existe um ponto ξ em (x_a, x_b) tal que:

$$f'(\xi) = \frac{f(x_b) - f(x_a)}{(x_b - x_a)}, \quad \text{para } x_a < \xi < x_b.$$

¹² Este apêndice é baseado em Gomes-Ruggiero [27].

Em uma interpretação geométrica desse teorema ilustrada na figura a seguir, são mostradas $f'(\xi)$ (a tangente ao gráfico de $f(x)$ no ponto ξ) e a secante \overline{AB} , cujas inclinações são iguais:



Esse resultado é o fundamento básico dos métodos secantes que tem em comum, no cálculo da correção de direção da nova iteração, substituir a direção indicada pelas derivadas calculadas no ponto atual por uma direção aproximada, dada pela direção da secante que passa pelo ponto atual e o anterior. Geralmente, a partir da aproximação inicial, na primeira iteração se usa o método de Newton para obter uma segunda aproximação, e nas iterações seguintes se aplica o método secante. Para possibilitar o uso de figuras ilustrativas o método será apresentado inicialmente no \mathbf{R}^1 , e posteriormente estendido para o \mathbf{R}^n .

Como introdução será visto a seguir o método de Newton, que também pode ser derivado do teorema do valor médio. Para isso, considere-se a função $f(x)$, e os pontos x^* e x_k , com $f(x^*) = 0$ e x_k pertencente à vizinhança de x^* . Aplicando-se o resultado do teorema para o ponto x_k , tem-se:

$$f(x) - f(x_k) = f'(\xi) (x - x_k),$$

¹³ A metodologia será apresentada para $f(x) = 0$, tradicionalmente utilizada na literatura para representar sistemas não-lineares.

para x qualquer na vizinhança de x_k .

Aproximando $f'(\xi)$ por $f'(x_k)$ tem-se um modelo linear local em torno do ponto x_k para $f(x)$:

$$L_k(x) = f(x_k) + f'(x_k)(x - x_k)$$

Como se deseja um ponto x_{k+1} tal que $L_k(x_{k+1}) = 0$, então

$$0 = f(x_k) + f'(x_k)(x_{k+1} - x_k)$$

Supondo-se $f'(x_k)$ diferente de zero é possível isolar x_{k+1} , obtendo-se assim a fórmula iterativa de Newton para uma nova aproximação de x^* :

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k)$$

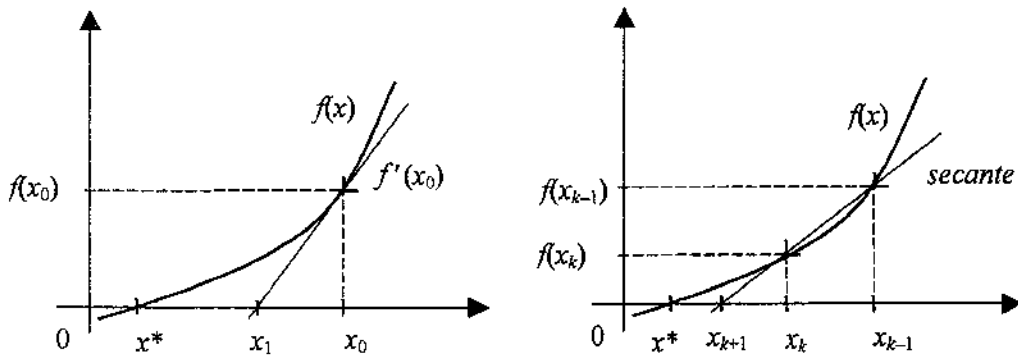
Com objetivos didáticos, será visto a seguir um método secante no \mathbf{R}^1 , onde a fórmula iterativa é semelhante à de Newton:

$$x_{k+1} = x_k - t_{k+1}^{-1} f(x_k),$$

onde t_{k+1} é a correção de direção na iteração k . Observe-se que a obtenção de t_{k+1} deve ser feita antes da aplicação da fórmula iterativa acima, sendo que na primeira iteração se usa a direção da tangente (como no método de Newton, com $t_1 = f'(x_0)$), e nas iterações seguintes se usa a direção da secante que passa pelos dois últimos pontos obtidos no procedimento iterativo:

$$t_{k+1} = \frac{f(x_k) - f(x_{k-1})}{(x_k - x_{k-1})}$$

As figuras a seguir ilustram o caso da iteração inicial e o caso das iterações seguintes:



Da definição da correção de direção tem-se que t_{k+1} deve satisfazer a equação:

$$t_{k+1} (x_k - x_{k-1}) = f(x_k) - f(x_{k-1})$$

É usual fazer $s_k = (x_k - x_{k-1})$ e $y_k = f(x_k) - f(x_{k-1})$, obtendo-se:

$$t_{k+1} s_k = y_k$$

No formato acima, essa equação é conhecida como a *equação secante*. Deve ser observado que no \mathbf{R}^1 a equação secante é composta apenas por escalares, e nesse caso, se $s_k \neq 0$, existe um único t_{k+1} que satisfaz a equação.

A seguir será visto o caso mais geral, onde \mathbf{x}_k , \mathbf{x}^* e $\mathbf{f}(\mathbf{x})$ pertencem ao \mathbf{R}^n .

Inicialmente será visto o método de Newton, onde a sequência $\{\mathbf{x}_k\}$ gerada pelo procedimento iterativo é tal que a aproximação \mathbf{x}_{k+1} é um zero de um modelo local linear para $\mathbf{f}(\mathbf{x}_k)$ construído em torno de \mathbf{x}_k :

$$\mathbf{L}_k(\mathbf{x}) = \mathbf{f}(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k) (\mathbf{x} - \mathbf{x}_k),$$

onde $\mathbf{J}(\mathbf{x}_k)$ é a matriz jacobiana de \mathbf{f} em \mathbf{x}_k .

De $\mathbf{L}_k(\mathbf{x}) = 0$ vem:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{J}(\mathbf{x}_k)^{-1} \mathbf{f}(\mathbf{x}_k)$$

Supondo-se que $\mathbf{J}(\mathbf{x}_k)$ seja não-singular, a equação acima é a fórmula iterativa de Newton para uma nova aproximação de \mathbf{x}^* . Entretanto, uma iteração do método de Newton

requer a avaliação da matriz jacobiana em \mathbf{x}_k e a resolução do sistema linear $\mathbf{J}(\mathbf{x}_k)^{-1} \mathbf{f}(\mathbf{x}_k)$, que geralmente são computacionalmente muito caros, não compensando a taxa quadrática de convergência.

Serão vistos a seguir diversos métodos secantes que visam evitar a avaliação da matriz jacobiana, desenvolvidos a partir da classe de métodos proposta por Broyden [7] em 1965 com esse objetivo central, onde as iterações são realizadas de acordo com a fórmula:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_{k+1}^{-1} \mathbf{f}(\mathbf{x}_k)$$

que, em uma extensão para o \mathbf{R}^n do método secante para o \mathbf{R}^1 visto acima, a correção de direção agora é feita por uma matriz \mathbf{B}_{k+1} , sendo que na iteração inicial $\mathbf{B}_1 = \mathbf{J}(\mathbf{x}_0)$ e nas iterações seguintes as matrizes \mathbf{B}_{k+1} devem satisfazer:

$$\mathbf{B}_{k+1} (\mathbf{x}_k - \mathbf{x}_{k-1}) = \mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{k-1})$$

Ou, na forma da *equação secante*, fazendo $\mathbf{s}_k = (\mathbf{x}_k - \mathbf{x}_{k-1})$ e $\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{k-1})$:

$$\mathbf{B}_{k+1} \mathbf{s}_k = \mathbf{y}_k$$

Entretanto, diferentemente do caso unidimensional, existem infinitas matrizes \mathbf{B}_{k+1} que satisfazem a equação secante acima, sendo necessárias condições adicionais para determinar uma única matriz \mathbf{B}_{k+1} . Assim, os métodos secantes diferem entre si pelas condições adicionais impostas sobre \mathbf{B}_{k+1} para possibilitar a resolução do sistema, tais como preservar alguma estrutura especial da matriz jacobiana (como simetria e esparsidade), obedecer algum princípio de variação mínima de \mathbf{B}_k , ou outras condições.

A condição adicional proposta por Broyden é que a matriz \mathbf{B}_{k+1} seja construída de modo que a troca no modelo afim seja mínima, impondo-se que \mathbf{B}_{k+1} não seja diferente de \mathbf{B}_k no espaço ortogonal a \mathbf{s}_k :

$$\mathbf{B}_{k+1} \mathbf{t} = \mathbf{B}_k \mathbf{t}, \text{ para } \forall \mathbf{t} \in \mathbf{R}^n, \text{ com } \mathbf{t}^\top \mathbf{s}_k = 0$$

Dessa condição deriva a "atualização secante de Broyden" para \mathbf{B}_k (ver [18] para detalhes):

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k}{\mathbf{s}_k^T \mathbf{s}_k} \mathbf{s}_k^T$$

O fato de \mathbf{B}_{k+1} ser obtida adicionando-se uma matriz de posto 1 à \mathbf{B}_k dificulta a implementação deste método no caso esparso, pois \mathbf{B}_{k+1} pode resultar não esparsa. Ainda que \mathbf{B}_{k-1} resulte esparsa, a estrutura de \mathbf{B}_k não será preservada, impedindo que se explore a esparsidade no método. Visando manter a esparsidade, as implementações práticas do método de Broyden utilizam a fórmula de Sherman–Morrison–Woodbury para o cálculo de \mathbf{B}_{k+1}^{-1} , onde definindo-se um vetor auxiliar \mathbf{w}_k tal que:

$$\mathbf{w}_k = \frac{\mathbf{s}_k - \mathbf{B}_k^{-1} \mathbf{y}_k}{\mathbf{s}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k},$$

a expressão para \mathbf{B}_{k+1}^{-1} fica:

$$\mathbf{B}_{k+1}^{-1} = (\mathbf{I} + \mathbf{w}_k \mathbf{s}_k^T) \mathbf{B}_k^{-1}$$

Assim, a cada iteração k devem ser armazenados os vetores \mathbf{w}_k e \mathbf{s}_k , e por essa razão o número de iterações consecutivas é limitado pela disponibilidade de memória da máquina. Ao se esgotar a memória disponível é feita uma iteração Newton, com $\mathbf{B}_{k-1} = \mathbf{J}(\mathbf{x}_k)$, reiniciando-se o processo.

A seguir será visto com mais detalhe o método de atualização de uma coluna por iteração (ACI), proposto por Martínez em 1983, que obtém a matriz \mathbf{B}_{k+1} através de uma atualização de posto um de uma coluna de \mathbf{B}_k , de modo que \mathbf{B}_{k+1} satisfaça a equação secante.

Conhecidos \mathbf{x}_k e \mathbf{B}_k , com $\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_{k-1})$ e $\mathbf{s}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$, se escolhe o índice j_k da coluna de \mathbf{B}_k que será modificada:

$$|s_{j_k}| > \alpha \|s_k\|_\infty, \quad \alpha > 0$$

Como só a coluna j_k é modificada, pode-se escrever:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{u}_k \mathbf{e}_{j_k}^T$$

onde o vetor \mathbf{u}_k deve ser tal que a equação secante seja satisfeita:

$$\begin{aligned} \mathbf{B}_{k+1} \mathbf{s}_k = \mathbf{y}_k &\Leftrightarrow (\mathbf{B}_k + \mathbf{u}_k \mathbf{e}_{j_k}^T) \mathbf{s}_k = \mathbf{y}_k \\ &\Leftrightarrow \mathbf{u}_k \mathbf{e}_{j_k}^T \mathbf{s}_k = \mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k \\ &\Leftrightarrow \mathbf{u}_k = \frac{\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k}{\mathbf{e}_{j_k}^T \mathbf{s}_k} \end{aligned}$$

Dai, substituindo \mathbf{u}_k na expressão iterativa, tem-se:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k - \mathbf{B}_k \mathbf{s}_k}{\mathbf{e}_{j_k}^T \mathbf{s}_k} \mathbf{e}_{j_k}^T$$

Usando a fórmula de Sherman-Morrison-Woodbury¹⁴, pode-se afirmar que a inversa de $\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{u}_k \mathbf{e}_{j_k}^T$ existe, se \mathbf{B}_k for inversível e se $(1 + \mathbf{e}_{j_k}^T \mathbf{B}_k^{-1} \mathbf{u}_k) \neq 0$, e é dada por:

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} - \frac{\mathbf{B}_k^{-1} \mathbf{u}_k \mathbf{e}_{j_k}^T \mathbf{B}_k^{-1}}{1 + \mathbf{e}_{j_k}^T \mathbf{B}_k^{-1} \mathbf{u}_k}$$

Substituindo \mathbf{u}_k na expressão acima, tem-se:

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\mathbf{s}_k - \mathbf{B}_k^{-1} \mathbf{y}_k) \mathbf{e}_{j_k}^T \mathbf{B}_k^{-1}}{(\mathbf{e}_{j_k}^T \mathbf{B}_k^{-1} \mathbf{y}_k)}$$

Definindo-se \mathbf{w}_k tal que:

$$\mathbf{w}_k = \frac{(\mathbf{s}_k - \mathbf{B}_k^{-1} \mathbf{y}_k)}{(\mathbf{e}_{j_k}^T \mathbf{B}_k^{-1} \mathbf{y}_k)},$$

a expressão para \mathbf{B}_{k+1}^{-1} fica:

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \mathbf{w}_k \mathbf{e}_{j_k}^T \mathbf{B}_k^{-1}$$

E então:

$$\mathbf{B}_{k+1}^{-1} = (\mathbf{I} + \mathbf{w}_k \mathbf{e}_{j_k}^T) \mathbf{B}_k^{-1}$$

Pode-se observar que a cada iteração k efetuada é preciso armazenar apenas um vetor \mathbf{w}_k e um índice j_k adicionais. É sempre possível reiniciar o processo iterativo fazendo $\mathbf{B}_{k+1} = \mathbf{J}(\mathbf{x}_k)$.

Na próxima página, o algoritmo ACI (Atualização de uma Coluna por Iteração) em uma adaptação de Gomes-Ruggiero [27].

Na página seguinte, a implementação desse algoritmo em linguagem C.

¹⁴ Ver Fang e Puthenpura[18] para detalhes.

Algoritmo ACI

Dados a aproximação inicial \mathbf{x}_0 , a quantidade m de vetores \mathbf{w}_k adicionais (capacidade de armazenamento de vetores disponível), o parâmetro β (usado no cálculo para restringir a amplitude do passo \mathbf{s}_k) e uma precisão ε , executar:

- Passo 0: $k = 0, n = 0$
- Passo 1: obter $\mathbf{B}_k = \mathbf{J}(\mathbf{x}_k)$
- Passo 2: obter as matrizes \mathbf{L}, \mathbf{U} e \mathbf{P} tais que $\mathbf{P} \mathbf{B}_k = \mathbf{L} \mathbf{U}$
- Passo 3: resolver $\mathbf{L} \mathbf{U} \hat{\mathbf{s}}_k = \mathbf{P}(-\mathbf{f}(\mathbf{x}_k))$
- Passo 4: obter $\theta = \min\left\{1, \frac{\beta}{\|\mathbf{s}_k\|_\infty}\right\}$ e fazer $\mathbf{s}_k = \theta \hat{\mathbf{s}}_k$
- Passo 5: fazer $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$
se $\max_j \{ |f_j(\mathbf{x}_{k+1})| \} < \varepsilon$, então fim, \mathbf{x}_{k+1} é a solução
- Passo 6: fazer $q = \text{mod}(k, m)$
se $k \neq n$ e $q = 0$, então fazer $k = k+1$
 $n = k$
voltar ao passo 1.
- Passo 7: caso contrário, efetuar o cálculo de \mathbf{w}_k :
- 7.1 - obter $j_k = \arg \max_j \{ |e_j^T \mathbf{s}_k| \}$
- 7.2 - cálculo de $\mathbf{t} = -\mathbf{B}_k^{-1} \mathbf{f}(\mathbf{x}_k)$:
- 7.2.1 - resolver $\mathbf{L} \mathbf{U} \mathbf{t} = -\mathbf{P} \mathbf{f}(\mathbf{x}_k)$
- 7.2.2 - para $i = n, \dots, (k-1)$ fazer:
 $\mathbf{t} \leftarrow (\mathbf{I} + \mathbf{w}_i \mathbf{e}_{j_i}^T) \mathbf{t}$
- 7.3 - cálculo de $\mathbf{v}_k = \mathbf{B}_k^{-1} \mathbf{y}_k$:
- fazer $\mathbf{v}_k = \hat{\mathbf{s}}_k - \mathbf{t}$
- 7.4 - fazer $\mathbf{w}_k = \frac{\mathbf{s}_k - \mathbf{v}_k}{\mathbf{e}_{j_k}^T \mathbf{v}_k}$
- Passo 8: completar o cálculo do produto $-\mathbf{B}_{k+1} \mathbf{f}(\mathbf{x}_{k+1})$:
- fazer $\hat{\mathbf{s}}_k = \mathbf{t} + \mathbf{w}_k \mathbf{e}_{j_k}^T \mathbf{t}$
- Passo 9: $k = k+1$
voltar ao passo 4.

Implementação do algoritmo ACI

Nessa implementação do algoritmo, a fatoração da matriz jacobiana é feita através de uma versão em C do conjunto de subrotinas Harwell/MA28 [48], originalmente em Fortran, e que não é listada devido a sua extensão.

```
void aci( z0x, lz0, irnx, lirnx, icnx, licnx, ax, lax, bx, lbx,
        epsilon, limiter, corfacx, infox)
    double *z0x,*ax,*bx,epsilon;
    int *irnx,*icnx,limiter,infox,corfacx;
    long lz0,lirnx,licnx,lax,lbx;
{
    /* Local variables */
    int licn, lirn, n, iflag, llarge, iconcp, irank;
    int idisp[2], irncp, nsrch, ndrop, mtype, numnz, num;
    int lp, mp, nz, nzx, minicn, minirn, istart;
    int *ikeep, *iw, *icn, *irn, *izeros;
    int corfac;
    int i, j, jx, iz;
    int k, kmax, *jkp, *p, jk, jka, jkf, jkfa, njk, ik;
    long lbig, grow, lblock, abort1, abort2, abort3;
    long lzeros, lx;
    double *W[100], *f, *sk, *s, *xn, *t, *wx, fkmax, skmax, teta, aux, eps, temp;
    double themax, big, tol, resid, u;
    double *a, *b, *w, *x, *h;

    extern double evalexpr();
    extern void Mathmens();
    extern void NoMem();
    extern int ma28ad_();
    extern int ma28cd_();
    extern int stats();

    if(infox){
        _ftime(&acttime);
        til= acttime.time + acttime.millitm/1000.;
    }

    /* Parameter adjustments */
    --bx;
    --icnx;
    --irnx;
    --ax;

    /* Inicializacao das estruturas da MA28 */
    logcomp=0;
    tol = 0.0;           u = 0.1;           mtype = 1;
    lblock = 1;         grow = 1;           lbig = 0;
    abort1 = 1;         abort2 = 1;        abort3 = 0;
    nsrch = 32768;      istart = 0;
    lp = 6;             mp = 6;
    iflag=-15;         lzeros=0;
    n = lz0;           lx=0;
    nz=0;
    for(i=1;i<=n;i++) nz+=expr[i].ndep;
    corfac=corfacx;
    licn = nz*corfac;
    lirn = nz*2;
    icn=malloc(licn*sizeof(int));   if(!icn) {NoMem();goto error;};
    irn=malloc(lirn*sizeof(int));   if(!irn) {NoMem();goto error;};
    ikeep=malloc(n*5*sizeof(int));  if(!ikeep) {NoMem();goto error;};
    iw=malloc(n*8*sizeof(int));     if(!iw) {NoMem();goto error;};
    a=malloc(licn*sizeof(double));  if(!a) {NoMem();goto error;};
    b=malloc(n*sizeof(double));     if(!b) {NoMem();goto error;};
    w=malloc(n*sizeof(double));     if(!w) {NoMem();goto error;};
}
```

```

x=malloc(n*sizeof(double));      if(!x) {NoMem();goto errol;};
h=malloc(n*sizeof(double));      if(!h) {NoMem();goto errol;};
izeros=malloc(lz0*sizeof(int));  if(!izeros) {NoMem();goto errol;};
if(infox){
    _ftime(&acttime);
    ti2= acttime.time + acttime.millitm/1000.;
}
aux=0.;
eps=pow(10.,-6.);
for(i=0;i<n;i++) x[i]=z0x[i];

/* Avaliacao de f(x0) e h */
for(i=1;i<=n;i++){
    h[i-1] = eps*dr_aps(x[i-1]);
    if (h[i-1] == 0.0) h[i-1] = eps;
    expr[i].val=evalexpr(i,x);
    aux=aux+expr[i].val*x[i-1];
}
if(infox){
    _ftime(&acttime);
    tf2= acttime.time + acttime.millitm/1000.;
    sprintf(mens,"Print[\"avaliacao f.....: %10.4f segundos\"];",tf2-ti2);
    Mathmens(mens);
    ti2=tf2;
}

/* Obtencao do jacobiano por diferencas finitas */
iz=0;
for(i=1;i<=n;i++){
    b[i-1]=expr[i].val;
    for(j=0;j<expr[i].ndep;j++){
        j= expr[i].dep[j] - 1;
        temp = x[j];
        x[j] = x[j] + h[j];
        aux = (evalexpr(i,x) - expr[i].val) / h[j];
        x[j] = temp;
        if(aux != 0.0){
            irn[i2] = i;
            icn[i2] = j+1;
            a[i2] = aux;
            b[i-1] = b[i-1] - aux * temp;
            iz++;
        }
    }
};
nz=iz;
if(infox){
    _ftime(&acttime);
    tf2= acttime.time + acttime.millitm/1000.;
    sprintf(mens,"Print[\"construcao jacobiano: %10.4f segundos\"];",tf2-ti2);
    Mathmens(mens);
}

/* Decomposicao LU para Matriz Esparsa */
if(infox){
    _ftime(&acttime);
    ti2= acttime.time + acttime.millitm/1000.;
}
ma28ad_(&n, &nz, a, &icn, irn, &irn, icn, &u, &keep, iw, w, &iflag, &ip,
        &mp, &lblock, &grow, &abort1, &abort2, &abort3, &lbig, &tol, &themax,
        &big, &irank, idisp, &irncp, &icncp, &minirn, &minicn, &ndrop,
        &nsrch, &istart, &numnz, &num, &large);

if(infox){
    _ftime(&acttime);
    tf2= acttime.time + acttime.millitm/1000.;
    sprintf(mens,"Print[\"fatoracao matriz....: %10.4f segundos\"j];",tf2-ti2);
    Mathmens(mens);
}

if (iflag != 0) {

```

```

    sprintf(mens,"Print[\\\"*** Error return from LU factorization routine\\\"]");
    Mathmens(mens);
    if (irank != n) {
        sprintf(mens,"Print[\\\"*** Matrix is not full rank\\\"]"); Mathmens(mens);
    }
    goto errol;
}
if(infox)stats(n,nz,mtype,tol,ndrop,u,lblock,grow,w[0],lbig,
    abort1,abort2,abort3,nrchr,irank,themax,
    idisp,licn,icncp,minicn,lirn,irncp,minirn,
    num,large,resid,iflag,corfac,big);

/* Inicializacao das estruturas do ACI */
k=0;
kmax=limiter;
fkmax=1.;
njk=-1;
jk=0;
jka=-1;
jkf=0;
jka=-1;
s=malloc(n*sizeof(double));          if(!s) {NoMem();goto errol;};
sk=malloc(n*sizeof(double));         if(!sk) {NoMem();goto errol;};
xn=malloc(n*sizeof(double));         if(!xn) {NoMem();goto errol;};
t=malloc(n*sizeof(double));          if(!t) {NoMem();goto errol;};
wx=malloc(n*sizeof(double));         if(!wx) {NoMem();goto errol;};
f=malloc(n*sizeof(double));          if(!f) {NoMem();goto errol;};
jkp=malloc(kmax*sizeof(int));        if(!jkg) {NoMem();goto errol;};
p=malloc(n*sizeof(int));              if(!p) {NoMem();goto errol;};
for(i=0; i<n; i++) p[i]=0;
for(i=0; i<n; i++) { f[i]=expr[i+1].val; sk[i] = -f[i]; }
jkf=0; for(i=1; i<n; i++) if(dr_abs(f[i])>dr_abs(f[jkf])) jkf=i;
fkmax=dr_abs(f[jkf]);
if(infox){
    _ftime(&acttime);
    ti2= acttime.time + acttime.millitm/1000.;
}

/* algoritmo */
while(fkmax > epsilon && k < kmax){

    /* cálculo do passo sk */
    if(k==0){
        /* Newton */
        ma28cd_(&n, a, &licn, icn, ikeep, sk, w, &mtype, idisp, &resid);
    }
    else {
        /* ACI */
        ma28cd_(&n, a, &licn, icn, ikeep, t, w, &mtype, idisp, &resid);
        for(ik=0; ik<=njk; ik++){
            for(i=0; i<n; i++) wx[i]=t[i]+W[ik][i]*t[jkp[ik]];
            for(i=0; i<n; i++) t[i]=wx[i];
        };
        for(i=0; i<n; i++) wx[i]=(s[i]-sk[i]+t[i])/(sk[jk]-t[jk]);
        for(i=0; i<n; i++) sk[i]=t[i]+wx[i]*t[jk];
        if(p[jk]==0){
            p[jk]=++njk;
            jkp[njk]=jk;
            W[p[jk]]=malloc(n*sizeof(double)); if(!W[p[jk]]) {NoMem();goto errol;};
            for(i=0; i<n; i++) W[p[jk]][i]=wx[i];
        }
        else {
            aux=W[p[jk]][jk];
            for(i=0; i<n; i++) W[p[jk]][i]=W[p[jk]][i] + (1.+aux)*wx[i];
        }
    };

    /* cálculo do novo ponto */
    teta=1.; aux=1.;

    if(jk==jka){

```

```

    Volta:
    if(teta>0.8){
        teta=teta-0.1;
        for(i=0;i<n;i++) xn[i]=x[i]+teta*sk[i];
        if(evalexpr(jkf+1,xn)*f[jkf]>0.)goto Volta;
        else teta=teta+0.05;
    }
}
*/
for(i=0;i<n;i++) s[i]=teta*sk[i];
for(i=0;i<n;i++) x[i]=x[i]+s[i];
for(i=0;i<n;i++) {f[i]=evalexpr(i+1,x); t[i]=-f[i];}
jkfa=jkf;
jkf=0; for(i=1;i<n;i++) if(dr_abs(f[i])>dr_abs(f[jkf])) jkf=i; fkmax=dr_abs(f[jkf]);
// 7.1 - Obtencao de jk
jka=jk;
jk=0; for(i=1;i<n;i++) if(dr_abs(sk[i])>dr_abs(sk[jk])) jk=i; skmax=dr_abs(sk[jk]);
if(infox){
    sprintf(mens,
        "Print[\" %d, jk= %d, teta= %f, fkmax= %f\",", k,jk+1,teta,f[jkf]);
    Mathmens(mens);
}
k++;
}
if(k==kmax && fkmax>epsilon){
    sprintf(mens,"Print[\"Warning return from ACI: iteration limit\"]");
    Mathmens(mens);
}
}

/*      PASSAR PARA O MATHEMATICA: */
/*      TIPO DA SOLUCAO */
/*      VETOR x (SOLUCAO - SOMENTE NAO ZEROS) */
/*      VETOR izeros (INDICE DE ZEROS) */
lzeros = 0;
for (i = 0; i < lz0; i++) {
    if(x[i]!=0.) x[i-lzeros] = x[i];
    else izeros[lzeros++] = i+1;
}
lx=lz0-lzeros;

errol:
if(infox){
    ftime(&acttime);
    tf2= acttime.time + acttime.millitm/1000.;
    sprintf(mens,"Print[\"tempo das iteracoes.: %10.4f segundos\",",tf2-ti2);
    Mathmens(mens);
    sprintf(mens,"Print[\"tempo total.....: %10.4f segundos\",",tf2-ti1);
    Mathmens(mens);
}
MLPutFunction( stdlink, "List", 4L);
MLPutInteger( stdlink, iflag);
MLPutRealList( stdlink, x, lx);
MLPutIntegerList( stdlink, izeros, lzeros);
MLPutIntegerList( stdlink, izeros, 0L);

if(s)free( s );
if(sk)free( sk );
if(xn)free( xn );
if(t)free( t );
if(f)free( f );
if(jkp)free( jkp );
if(p)free( p );
if(wx)free( wx );
if(njk>-1){
    for(ik=0;ik<=njk;ik++) if(W[ik])free(W[ik]);
}
if(izeros)free(izeros);
if(lcn)free(lcn);
if(irn)free(irn);
if(ikkeep)free(ikkeep);
if(iw)free(iw);

```



```
    if(x) free(x);
    if(a) free(a);
    if(b) free(b);
    if(w) free(w);
    if(h) free(h);

    return;
} /* aci */
```

A6 – Tabelas do GTAP

A6.1 – Tabela de Regiões

aus	Australia
nzl	New Zealand
jpn	Japan
kor	Korea
idn	Indonesia
mys	Malaysia
phl	Philippines
sgp	Singapore
tha	Thailand
vnm	Viet Nam
chn	China
hkg	Hong Kong
twn	Taiwan
ind	India
lka	Sri Lanka
ras	Rest of South Asia
can	Canada
usa	United States of America
mex	Mexico
cam	Central America and the Caribbean
ven	Venezuela
col	Colombia
rap	Rest of the Andean Pact
arg	Argentina
bra	Brazil
chl	Chile
ury	Uruguay
rsm	Rest of South America
gbr	United Kingdom
deu	Germany
dnk	Denmark
swe	Sweden
fin	Finland
reu	Rest of European Union
eft	EFTA
cea	Central European Associates
fsu	Former Soviet Union
tur	Turkey
rme	Rest of Middle East
mar	Morocco
rnf	Rest of North Africa
saf	South African Customs Union
rsa	Rest of southern Africa
rss	Rest of sub-Saharan Africa
row	Rest of World

A6.2 – Tabela de Setores

pdr	Paddy rice
wht	Wheat
gro	Cereal grains nec
v f	Vegetables, fruit, nuts
osd	Oil seeds
c b	Sugar cane, sugar beet
pfb	Plant-based fibers
ocr	Crops nec
ctl	Bovine cattle, sheep and goats, horses
oap	Animal products nec
rmk	Raw milk
wol	Wool, silk-worm cocoons
for	Forestry
fsh	Fishing
col	Coal
oil	Oil
gas	Gas
omn	Minerals nec
cmt	Bovine cattle, sheep and goat, horse meat prods
omt	Meat products nec
vol	Vegetable oils and fats
mil	Dairy products
pcr	Processed rice
sgr	Sugar
ofd	Food products nec
b t	Beverages and tobacco products
tex	Textiles
wap	Wearing apparel
lea	Leather products
lum	Wood products
ppp	Paper products, publishing
p c	Petroleum, coal products
crp	Chemical, rubber, plastic products
nmn	Mineral products nec
i s	Ferrous metals
nfm	Metals nec
fmp	Metal products
mvh	Motor vehicles and parts
otn	Transport equipment nec
ele	Electronic equipment
ome	Machinery and equipment nec
omf	Manufactures nec
ely	Electricity
gdt	Gas manufacture, distribution
wtr	Water
cns	Construction
t t	Trade, transport
osp	Financial, business, recreational services
osg	Public admin and defence, education, health
dwe	Dwellings

A6.3 – Tabela de Matrizes Numéricas do GTAP

EVOA(i,r)	value of commodity i output in region r.
EVFA(i,j,r)	producer expenditure on i by industry j, in region r, valued at agent's prices
SAVE(r)	expenditure on NET savings in region r valued at agent's prices
VDFA(i,j,r)	purchases of domestic i r for use in j in region r
VIFA(i,j,r)	purchases of imported i r for use in j in region r
VDFA(i,r)	private household expenditure on domestic i in r
VIFA(i,r)	private household expenditure on imported i
VDGA(i,r)	government household expenditure on domestic i in r
VIGA(i,r)	government household expenditure on imported i
VKB(r)	value of beginning-of-period capital stock, in region r
VDEP(r)	value of capital depreciation, in r (depreciation rate is exogenous and therefore does not appear in update)
VXMD(i,r,s)	exports of commodity i from region r to destination s valued at market prices (tradeables only)
VST(i,r)	exports of commodity i from region r for international transportation valued at market prices (tradeables only)
VFM(i,j,r)	producer expenditure on i by industry j, in region r, valued at market prices
VIFM(i,j,r)	purchases of imports i for use in j in region r
VDEM(i,j,r)	purchases of domestic i r for use in j in region r
VIPM(i,r)	private household expenditure on i in r
VDEPM(i,r)	private household expenditure on domestic i in r
VIGM(i,r)	gov't household expenditure on i in r
VDGM(i,r)	government household expenditure on domestic i in r
VIMS(i,r,s)	imports of commodity i from region r to s, valued at domestic market prices
VXWD(i,r,s)	exports of commodity i from region r to destination s valued fob (tradeables only)
VIWS(i,r,s)	imports of commodity i from region r to s, valued cif (tradeables only)

A6.4 – Arquivo de dados GTAP3x3.DAT

```

BeginPackage["gtap$dat`"];
Off[General::spell1];
sE={land,labor,capital};
sT={food,mnfcs,svces};
sP={food,mnfcs,svces,cgds};
sR={usa,eu,row};
subpar={{0.866678,0.897081,0.967472},
        {0.239291,0.317116,0.669294},
        {9.999990 10^-03,9.999990 10^-03,9.999990 10^-03}};
incpar={{0.133962,0.160812,0.331707},
        {0.793119,0.789978,0.838876},
        {1.14379,1.26177,1.29555}};
esubd={2.39901,2.79556,1.94365};
esubm={4.63905,6.08810,3.91673};
esubva={0.789314,1.21992,1.38946,0.000000};
etrae=-1.00000;
rorflex={10.0000,10.0000,10.0000};
rordelta=0.000000;
evfa={{ { 17125.5, 0.000000, 0.000000, 0.000000},
        { 96441.2, 792128., 2.606860*10^+06, 0.000000},
        { 97376.8, 382360., 1.408053*10^+06, 0.000000}},
      {{ 26493.4, 0.000000, 0.000000, 0.000000},
        { 322275., 983005., 2.883581*10^+06, 0.000000},
        { 161469., 507072., 1.504840*10^+06, 0.000000}},
      {{ 189538., 0.000000, 0.000000, 0.000000},
        { 450327., 1.205943*10^+06, 3.356066*10^+06, 0.000000},
        { 345572., 1.290375*10^+06, 2.456803*10^+06, 0.000000}}};
evoa={{ 17125.5, 26493.4, 189538.},
      { 3.495430*10^+06, 4.188861*10^+06, 5.012336*10^+06},
      { 1.887790*10^+06, 2.173381*10^+06, 4.092750*10^+06}};
save={ 94962.9, 483721., 1.416088*10^+06};
vdep={ 642439., 820761., 1.263439*10^+06};
vdfa={{ { 205558., 6410.94, 74399.2, 13.1041},
        { 84772.3, 817396., 624365., 237332.},
        { 146429., 570221., 1.362529*10^+06, 457696.}},
      {{ 416001., 31753.4, 98144.8, 1614.86},
        { 131379., 1.272887*10^+06, 719176., 420478.},
        { 133418., 478624., 1.020430*10^+06, 812942.}},
      {{ 756612., 91078.0, 164493., 13338.1},
        { 214411., 2.260118*10^+06, 1.382614*10^+06, 731243.},
        { 251116., 1.189158*10^+06, 2.107787*10^+06, 1.516343*10^+06}}};
vdfm={{ { 205558., 6410.94, 74399.2, 13.1041},
        { 84772.3, 817396., 624365., 237332.},
        { 146429., 570221., 1.362529*10^+06, 457696.}},
      {{ 422648., 32096.1, 91253.3, 1631.18},
        { 129259., 1.254759*10^+06, 702373., 416786.},
        { 132375., 474769., 999881., 804980.}},
      {{ 758052., 90475.7, 160515., 13209.4},
        { 212897., 2.248894*10^+06, 1.358590*10^+06, 728828.},
        { 250486., 1.186568*10^+06, 2.090145*10^+06, 1.515264*10^+06}}};
vdga= { { 10993.3, 67.5141, 9161.28},
        { 191368., 1074.55, 104126.},
        { 870490., 1.263114*10^+06, 1.216695*10^+06}};

```

```

vdgm= {{ 10993.3, 67.5141, 9161.28},
        { 191368., 1074.55, 104124.},
        { 870490., 1.261331*10^+06, 1.216650*10^+06}};
vdpa= {{ 318045., 597677., 1.262444*10^+06},
        { 525012., 681164., 970310.},
        { 2.988009*10^+06, 2.637620*10^+06, 3.597929*10^+06}};
vdpm= {{ 318045., 568163., 1.238019*10^+06},
        { 525012., 660948., 942632.},
        { 2.988009*10^+06, 2.580938*10^+06, 3.583712*10^+06}};
vfm= {{{ 17125.5, 0.000000, 0.000000, 0.000000},
        { 96441.2, 792128., 2.606860*10^+06, 0.000000},
        { 97376.8, 382360., 1.408053*10^+06, 0.000000}},
      {{ 26493.4, 0.000000, 0.000000, 0.000000},
        { 322275., 983005., 2.883581*10^+06, 0.000000},
        { 161469., 507072., 1.504840*10^+06, 0.000000}},
      {{ 189538., 0.000000, 0.000000, 0.000000},
        { 450327., 1.205943*10^+06, 3.356066*10^+06, 0.000000},
        { 345572., 1.290375*10^+06, 2.456803*10^+06, 0.000000}}};
vifa={{ { 13631.7, 373.050, 3906.71, 1.40877},
        { 8208.62, 207796., 63590.6, 79531.2},
        { 1002.46, 12859.8, 18537.9, 5366.01}},
      {{ 42092.3, 7912.63, 3554.66, 136.945},
        { 16193.9, 266868., 76663.3, 106557.},
        { 3772.01, 29626.7, 103271., 1410.07}},
      {{ 105026., 35622.2, 19106.6, 4069.24},
        { 40676.0, 792411., 265739., 319312.},
        { 11724.9, 44417.6, 156316., 14027.5}}};
vifm={{ { 13631.7, 373.050, 3906.71, 1.40877},
        { 8208.62, 207796., 63590.6, 79531.2},
        { 1002.46, 12859.8, 18537.9, 5366.01}},
      {{ 42092.3, 7912.63, 3554.66, 136.945},
        { 16193.9, 266868., 76663.3, 106557.},
        { 3772.01, 29626.7, 103271., 1410.07}},
      {{ 105418., 35232.2, 18738.5, 3199.82},
        { 40417.8, 780211., 260163., 314557.},
        { 11713.9, 44284.4, 155695., 13989.8}}};
viga= {{ 429.605, 2.45493, 359.833},
        { 31457.0, 213.698, 31809.7},
        { 3530.50, 1004.62, 13704.8}};
vigm= {{ 429.605, 2.45493, 359.833},
        { 31457.0, 213.698, 31794.7},
        { 3530.50, 1004.62, 13704.7}};
vims={{ { 1.100000*10^-09, 6859.38, 26046.6},
        { 2.000000*10^-09, 97204.2, 472786.},
        { 6.000000*10^-10, 15146.8, 54940.3}},
      {{ 13074.4, 1.100000*10^-09, 66762.6},
        { 105716., 2.000000*10^-09, 483935.},
        { 51672.4, 6.000000*10^-10, 113339.}},
      {{ 71049.4, 45601.9, 118099.},
        { 336508., 451205., 914030.},
        { 83485.3, 100467., 160888.}}};
vipa= {{ 14563.5, 27296.3, 74027.8},
        { 179406., 143088., 280046.},
        { 28790.4, 27237.4, 105782.}};
vipm= {{ 14563.5, 26138.1, 71801.8},
        { 179406., 123155., 274600.},
        { 28790.4, 25926.6, 105452.}};

```

```

viws={{ { 1.100000*10^-09, 6233.81, 23308.4},
        { 2.000000*10^-09, 88663.3, 430573.},
        { 6.000000*10^-10, 15146.8, 54940.3}},
        {{ 9551.25, 1.100000*10^-09, 47160.8},
        { 99786.7, 2.000000*10^-09, 448896.},
        { 51672.4, 6.000000*10^-10, 113339.}},
        {{ 42915.8, 39178.3, 86077.3},
        { 306211., 403618., 826391.},
        { 83377.4, 100461., 160872.}}};
vkb={ 1.606098*10^+07, 2.051902*10^+07, 3.158598*10^+07};
vst= {{ 0.000000, 0.000000, 0.000000},
      { 0.000000, 0.000000, 0.000000},
      { 17582.3, 119137., 72844.1}}};
vxmd={{ { 1.100000*10^-09, 5644.06, 20775.4},
        { 2.000000*10^-09, 80850.3, 394642.},
        { 6.000000*10^-10, 15201.5, 54887.1}},
        {{ 8563.99, 1.100000*10^-09, 41459.8},
        { 92695.5, 2.000000*10^-09, 401265.},
        { 51672.4, 6.000000*10^-10, 113154.}},
        {{ 39950.3, 46810.9, 77794.2},
        { 283716., 366257., 765480.},
        { 83377.4, 101020., 160494.}}};
vxwd={{ { 1.100000*10^-09, 5698.90, 20907.6},
        { 2.000000*10^-09, 82358.7, 403067.},
        { 6.000000*10^-10, 15146.8, 54940.3}},
        {{ 8527.68, 1.100000*10^-09, 42098.9},
        { 92695.5, 2.000000*10^-09, 415598.},
        { 51672.4, 6.000000*10^-10, 113339.}},
        {{ 38756.6, 35945.1, 78327.4},
        { 283716., 373002., 768303.},
        { 83377.4, 100461., 160872.}}};
EndPackage[];

```

A6.5 – Solução de equilíbrio do modelo[0] do GTAP3x3

p[food[usa]]->1.	y[food[fd,usa]]->1.
p[food[fd,usa]]->1.	y[mnfcs[fd,usa]]->1.
p[mnfcs[usa]]->1.	y[svces[fd,usa]]->1.
p[mnfcs[fd,usa]]->1.	y[food[va,usa]]->1.
p[svces[usa]]->1.	y[food[food,usa]]->1.
p[svces[fd,usa]]->1.	y[mnfcs[food,usa]]->1.
p[land[usa]]->1.	y[svces[food,usa]]->1.
p[labor[usa]]->1.	y[food[usa]]->1.
p[capital[usa]]->1.	y[mnfcs[va,usa]]->1.
p[food[va,usa]]->1.	y[food[mnfcs,usa]]->1.
p[food[im,usa]]->1.	y[mnfcs[mnfcs,usa]]->1.
p[food[food,usa]]->1.	y[svces[mnfcs,usa]]->1.
p[mnfcs[im,usa]]->1.	y[mnfcs[usa]]->1.
p[mnfcs[food,usa]]->1.	y[svces[va,usa]]->1.
p[svces[im,usa]]->1.	y[food[svces,usa]]->1.
p[svces[food,usa]]->1.	y[mnfcs[svces,usa]]->1.
p[mnfcs[va,usa]]->1.	y[svces[svces,usa]]->1.
p[food[mnfcs,usa]]->1.	y[svces[usa]]->1.
p[mnfcs[mnfcs,usa]]->1.	y[food[cgds,usa]]->1.
p[svces[mnfcs,usa]]->1.	y[mnfcs[cgds,usa]]->1.
p[svces[va,usa]]->1.	y[svces[cgds,usa]]->1.
p[food[svces,usa]]->1.	y[cgds[usa]]->1.
p[mnfcs[svces,usa]]->1.	y[food[pd,usa]]->1.
p[svces[svces,usa]]->1.	y[mnfcs[pd,usa]]->1.
p[food[cgds,usa]]->1.	y[svces[pd,usa]]->1.
p[mnfcs[cgds,usa]]->1.	y[pd[usa]]->1.
p[svces[cgds,usa]]->1.	y[food[gd,usa]]->1.
p[gd[usa]]->1.	y[mnfcs[gd,usa]]->1.
p[food[pd,usa]]->1.	y[svces[gd,usa]]->1.
p[mnfcs[pd,usa]]->1.	y[gd[usa]]->1.
p[svces[pd,usa]]->1.	y[food[im,usa]]->1.
p[pd[usa]]->1.	y[mnfcs[im,usa]]->0.999999
p[food[gd,usa]]->1.	y[svces[im,usa]]->1.
p[mnfcs[gd,usa]]->1.	y[food[cif,usa,eu]]->1.
p[svces[gd,usa]]->1.	y[food[cif,usa,row]]->0.999999
p[gd[usa]]->1.	y[mnfcs[cif,usa,eu]]->0.999999
p[food[cif,eu,usa]]->1.	y[mnfcs[cif,usa,row]]->1.
p[food[cif,row,usa]]->1.	y[svces[cif,usa,eu]]->0.999997
p[mnfcs[cif,eu,usa]]->1.	y[svces[cif,usa,row]]->0.999998
p[mnfcs[cif,row,usa]]->1.	y[food[fd,eu]]->1.
p[svces[cif,eu,usa]]->1.	y[mnfcs[fd,eu]]->1.
p[svces[cif,row,usa]]->1.	y[svces[fd,eu]]->1.
p[qt]]->1.	y[food[va,eu]]->1.
p[food[cif,usa,eu]]->1.	y[food[food,eu]]->1.
p[food[cif,usa,row]]->1.	y[mnfcs[food,eu]]->1.
p[mnfcs[cif,usa,eu]]->1.	y[svces[food,eu]]->1.
p[mnfcs[cif,usa,row]]->1.	y[food[eu]]->1.
p[svces[cif,usa,eu]]->1.	y[mnfcs[va,eu]]->1.
p[svces[cif,usa,row]]->1.	y[food[mnfcs,eu]]->1.
p[food[eu]]->1.	y[mnfcs[mnfcs,eu]]->1.

p[food[fd,eu]]->1.	y[svces[mnfcs,eu]]->1.
p[mnfcs[eu]]->1.	y[mnfcs[eu]]->1.
p[mnfcs[fd,eu]]->1.	y[svces[va,eu]]->1.
p[svces[eu]]->1.	y[food[svces,eu]]->1.
p[svces[fd,eu]]->1.	y[mnfcs[svces,eu]]->1.
p[land[eu]]->1.	y[svces[svces,eu]]->1.
p[labor[eu]]->1.	y[svces[eu]]->1.
p[capital[eu]]->1.	y[food[cgds,eu]]->1.
p[food[va,eu]]->1.	y[mnfcs[cgds,eu]]->1.
p[food[im,eu]]->1.	y[svces[cgds,eu]]->1.
p[food[food,eu]]->1.	y[cgds[eu]]->1.
p[mnfcs[im,eu]]->1.	y[food[pd,eu]]->1.
p[mnfcs[food,eu]]->1.	y[mnfcs[pd,eu]]->1.
p[svces[im,eu]]->1.	y[svces[pd,eu]]->1.
p[svces[food,eu]]->1.	y[pd[eu]]->1.
p[mnfcs[va,eu]]->1.	y[food[gd,eu]]->1.
p[food[mnfcs,eu]]->1.	y[mnfcs[gd,eu]]->1.
p[mnfcs[mnfcs,eu]]->1.	y[svces[gd,eu]]->1.
p[svces[mnfcs,eu]]->1.	y[gd[eu]]->1.
p[svces[va,eu]]->1.	y[food[im,eu]]->1.
p[food[svces,eu]]->1.	y[mnfcs[im,eu]]->1.
p[mnfcs[svces,eu]]->1.	y[svces[im,eu]]->0.999997
p[svces[svces,eu]]->1.	y[food[cif,eu,usa]]->1.
p[food[cgds,eu]]->1.	y[food[cif,eu,row]]->1.
p[mnfcs[cgds,eu]]->1.	y[mnfcs[cif,eu,usa]]->0.999999
p[svces[cgds,eu]]->1.	y[mnfcs[cif,eu,row]]->1.
p[cgds[eu]]->1.	y[svces[cif,eu,usa]]->1.
p[food[pd,eu]]->1.	y[svces[cif,eu,row]]->0.999999
p[mnfcs[pd,eu]]->1.	y[food[fd,row]]->1.
p[svces[pd,eu]]->1.	y[mnfcs[fd,row]]->1.
p[pd[eu]]->1.	y[svces[fd,row]]->1.
p[food[gd,eu]]->1.	y[food[va,row]]->1.
p[mnfcs[gd,eu]]->1.	y[food[food,row]]->1.
p[svces[gd,eu]]->1.	y[mnfcs[food,row]]->1.
p[gd[eu]]->1.	y[svces[food,row]]->1.
p[food[cif,row,eu]]->1.	y[food[row]]->1.
p[mnfcs[cif,row,eu]]->1.	y[mnfcs[va,row]]->1.
p[svces[cif,row,eu]]->1.	y[food[mnfcs,row]]->1.
p[food[cif,eu,row]]->1.	y[mnfcs[mnfcs,row]]->1.
p[mnfcs[cif,eu,row]]->1.	y[svces[mnfcs,row]]->1.
p[svces[cif,eu,row]]->1.	y[mnfcs[row]]->1.
p[food[row]]->1.	y[svces[va,row]]->1.
p[food[fd,row]]->1.	y[food[svces,row]]->1.
p[mnfcs[row]]->1.	y[mnfcs[svces,row]]->1.
p[mnfcs[fd,row]]->1.	y[svces[svces,row]]->1.
p[svces[row]]->1.	y[svces[row]]->1.
p[svces[fd,row]]->1.	y[food[cgds,row]]->1.
p[land[row]]->1.	y[mnfcs[cgds,row]]->1.
p[labor[row]]->1.	y[svces[cgds,row]]->1.
p[capital[row]]->1.	y[cgds[row]]->1.
p[food[va,row]]->1.	y[food[pd,row]]->1.
p[food[im,row]]->1.	y[mnfcs[pd,row]]->1.
p[food[food,row]]->1.	y[svces[pd,row]]->1.
p[mnfcs[im,row]]->1.	y[pd[row]]->1.

p[mnfcs[food,row]]->1.	y[food[gd,row]]->1.
p[svces[im,row]]->1.	y[mnfcs[gd,row]]->1.
p[svces[food,row]]->1.	y[svces[gd,row]]->1.
p[mnfcs[va,row]]->1.	y[gd[row]]->1.
p[food[mnfcs,row]]->1.	y[food[im,row]]->0.999999
p[mnfcs[mnfcs,row]]->1.	y[mnfcs[im,row]]->1.
p[svces[mnfcs,row]]->1.	y[svces[im,row]]->0.999999
p[svces[va,row]]->1.	y[food[cif,row,usa]]->1.
p[food[svces,row]]->1.	y[food[cif,row,eu]]->1.
p[mnfcs[svces,row]]->1.	y[food[cif,row,row]]->1.
p[svces[svces,row]]->1.	y[mnfcs[cif,row,usa]]->0.999999
p[food[cgds,row]]->1.	y[mnfcs[cif,row,eu]]->1.
p[mnfcs[cgds,row]]->1.	y[mnfcs[cif,row,row]]->1.
p[svces[cgds,row]]->1.	y[svces[cif,row,usa]]->1.
p[cgds[row]]->1.	y[svces[cif,row,eu]]->0.999998
p[food[pd,row]]->1.	y[svces[cif,row,row]]->0.999999
p[mnfcs[pd,row]]->1.	y[qt]->0.999995
p[svces[pd,row]]->1.	y[globalcgds]->1.
p[pd[row]]->1.	u[usa]->1.
p[food[gd,row]]->1.	u[eu]->1.
p[mnfcs[gd,row]]->1.	u[row]->1.
p[svces[gd,row]]->1.	
p[gd[row]]->1.	
p[food[cif,row,row]]->1.	
p[mnfcs[cif,row,row]]->1.	
p[svces[cif,row,row]]->1.	
p[globalcgds]->1	

A6.6 – Solução de equilíbrio do modelo[1] do GTAP3x3

p[food[im,eu]] -> 0.98014	y[food[cif,usa,eu]] -> 1.53454
p[land[usa]] -> 1.01369	y[food[cif,row,eu]] -> 0.948919
p[food[mnfcs,eu]] -> 0.994785	y[food[im,eu]] -> 1.04056
p[land[eu]] -> 0.995762	y[food[cif,eu,usa]] -> 1.01299
p[food[food,eu]] -> 0.996793	y[food[va,usa]] -> 1.00884
p[food[cgds,eu]] -> 0.99705	y[food[food,usa]] -> 1.00884
p[food[va,usa]] -> 1.00245	y[mnfcs[food,usa]] -> 1.00884
p[food[pd,eu]] -> 0.997696	y[svces[food,usa]] -> 1.00884
p[food[gd,eu]] -> 0.997857	y[food[usa]] -> 1.00884
p[food[svces,eu]] -> 0.99786	y[food[im,usa]] -> 1.00861
p[food[usa]] -> 1.00176	y[food[cif,usa,row]] -> 0.992173
p[food[fd,usa]] -> 1.00176	y[food[cif,row,usa]] -> 1.00746
p[food[gd,usa]] -> 1.00168	y[mnfcs[cif,usa,eu]] -> 0.993145
p[food[pd,usa]] -> 1.00167	y[mnfcs[cif,usa,row]] -> 0.993654
p[land[row]] -> 0.998348	y[food[fd,eu]] -> 0.993846
p[food[svces,usa]] -> 1.00165	y[food[fd,usa]] -> 1.00599
p[food[mnfcs,usa]] -> 1.00164	y[food[cif,eu,row]] -> 1.00585
p[food[food,usa]] -> 1.00163	y[svces[cif,usa,eu]] -> 0.994885
p[food[cif,usa,row]] -> 1.00158	y[svces[cif,usa,row]] -> 0.995007
p[capital[usa]] -> 1.00157	y[mnfcs[cif,eu,usa]] -> 1.00446
p[food[cif,usa,eu]] -> 1.00156	y[svces[cif,eu,usa]] -> 1.00387
p[food[cgds,usa]] -> 1.00155	y[food[va,eu]] -> 0.997107
p[food[eu]] -> 0.998516	y[food[food,eu]] -> 0.997107

p[food[fd,eu]] -> 0.998516	y[mnfcs[food,eu]] -> 0.997107
p[mnfcs[va,usa]] -> 1.00143	y[svces[food,eu]] -> 0.997107
p[svces[va,usa]] -> 1.00143	y[food[eu]] -> 0.997107
p[svces[usa]] -> 1.00139	y[mnfcs[cif,eu,row]] -> 1.00284
p[svces[fd,usa]] -> 1.00139	y[svces[im,usa]] -> 1.00272
p[svces[cif,usa,eu]] -> 1.00139	y[mnfcs[im,usa]] -> 1.00262
p[svces[cif,usa,row]] -> 1.00139	y[svces[cif,row,usa]] -> 1.00241
p[svces[food,usa]] -> 1.00138	y[mnfcs[cif,row,usa]] -> 1.00224
p[svces[gd,usa]] -> 1.00138	y[svces[cif,eu,row]] -> 1.00194
p[svces[svces,usa]] -> 1.00137	y[food[gd,eu]] -> 1.00162
p[svces[cgds,usa]] -> 1.00137	y[qt] -> 1.00157
p[svces[pd,usa]] -> 1.00137	y[mnfcs[va,usa]] -> 0.998489
p[food[cif,eu,row]] -> 0.99863	y[food[mnfcs,usa]] -> 0.998489
p[food[cif,eu,usa]] -> 0.998634	y[mnfcs[mnfcs,usa]] -> 0.998489
p[labor[usa]] -> 1.00136	y[svces[mnfcs,usa]] -> 0.998489
p[svces[mnfcs,usa]] -> 1.00136	y[mnfcs[usa]] -> 0.998489
p[pd[usa]] -> 1.00132	y[svces[im,eu]] -> 0.99864
p[gd[usa]] -> 1.00132	y[food[pd,eu]] -> 1.00134
p[mnfcs[usa]] -> 1.00125	y[mnfcs[im,eu]] -> 0.9989
p[mnfcs[fd,usa]] -> 1.00125	y[food[im,row]] -> 0.99894
p[cgds[usa]] -> 1.00119	y[food[va,row]] -> 0.998962
p[mnfcs[cif,usa,eu]] -> 1.00115	y[food[food,row]] -> 0.998962
p[mnfcs[cif,usa,row]] -> 1.00115	y[mnfcs[food,row]] -> 0.998962
p[mnfcs[food,usa]] -> 1.00113	y[svces[food,row]] -> 0.998962
p[mnfcs[svces,usa]] -> 1.00113	y[food[row]] -> 0.998962
p[mnfcs[gd,usa]] -> 1.00106	y[mnfcs[fd,usa]] -> 0.999082
p[mnfcs[mnfcs,usa]] -> 1.00098	y[mnfcs[va,eu]] -> 1.00078
p[mnfcs[cgds,usa]] -> 1.00092	y[food[mnfcs,eu]] -> 1.00078
p[mnfcs[pd,usa]] -> 1.00092	y[mnfcs[mnfcs,eu]] -> 1.00078
p[pd[eu]] -> 0.999349	y[svces[mnfcs,eu]] -> 1.00078
p[food[va,eu]] -> 0.999425	y[mnfcs[eu]] -> 1.00078
p[food[im,usa]] -> 0.999567	y[mnfcs[cif,row,row]] -> 1.00065
p[svces[im,eu]] -> 1.00043	y[food[fd,row]] -> 0.999365
p[svces[cif,eu,usa]] -> 0.999615	y[mnfcs[fd,eu]] -> 1.00053
p[svces[eu]] -> 0.999615	y[mnfcs[pd,usa]] -> 1.00048
p[svces[fd,eu]] -> 0.999615	y[svces[cif,row,row]] -> 1.00048
p[svces[gd,eu]] -> 0.999615	y[svces[im,row]] -> 0.999577
p[gd[eu]] -> 0.999615	y[mnfcs[gd,usa]] -> 1.00041
p[svces[cif,eu,row]] -> 0.999615	y[food[cif,row,row]] -> 1.00035
p[svces[cgds,eu]] -> 0.999616	y[svces[cif,row,eu]] -> 1.00035
p[mnfcs[eu]] -> 0.999617	y[mnfcs[va,row]] -> 1.00033
p[mnfcs[fd,eu]] -> 0.999617	y[food[mnfcs,row]] -> 1.00033
p[svces[pd,eu]] -> 0.999623	y[mnfcs[mnfcs,row]] -> 1.00033
p[labor[eu]] -> 0.999624	y[svces[mnfcs,row]] -> 1.00033
p[mnfcs[va,eu]] -> 0.999626	y[mnfcs[row]] -> 1.00033
p[svces[va,eu]] -> 0.999626	y[mnfcs[gd,eu]] -> 0.999759
p[capital[eu]] -> 0.999629	y[food[cgds,usa]] -> 0.999772
p[mnfcs[cif,eu,usa]] -> 0.999637	y[mnfcs[cgds,usa]] -> 0.999772
p[svces[food,eu]] -> 0.999637	y[svces[cgds,usa]] -> 0.999772
p[mnfcs[cif,eu,row]] -> 0.999638	y[cgds[usa]] -> 0.999772
p[cgds[eu]] -> 0.99966	y[food[gd,usa]] -> 0.999799
p[svces[mnfcs,eu]] -> 0.999662	y[mnfcs[fd,row]] -> 1.00019
p[food[va,row]] -> 0.999663	y[pd[usa]] -> 1.00016
p[mnfcs[svces,eu]] -> 0.999674	y[gd[usa]] -> 1.00016

p[mnfcs[food,eu]] -> 0.999682	y[mnfcs[cif,row,eu]] -> 1.00016
p[svces[svces,eu]] -> 0.999689	y[mnfcs[im,row]] -> 0.999846
p[mnfcs[gd,eu]] -> 0.999715	y[mnfcs[pd,eu]] -> 0.999852
p[mnfcs[mnfcs,eu]] -> 0.999719	y[svces[gd,eu]] -> 0.999858
p[mnfcs[pd,eu]] -> 0.999719	y[gd[eu]] -> 0.999858
p[mnfcs[cgds,eu]] -> 0.999736	y[svces[fd,eu]] -> 1.00014
p[svces[im,row]] -> 1.00022	y[svces[pd,usa]] -> 1.00012
p[mnfcs[im,eu]] -> 1.0002	y[pd[eu]] -> 1.00012
p[food[row]] -> 0.999805	y[food[pd,usa]] -> 0.999887
p[food[fd,row]] -> 0.999805	y[food[gd,row]] -> 1.00011
p[food[cif,row,row]] -> 0.999813	y[globalcgds] -> 0.999893
p[food[cif,row,usa]] -> 0.999814	y[mnfcs[gd,row]] -> 0.999895
p[food[cif,row,eu]] -> 0.999814	y[svces[gd,usa]] -> 1.0001
p[food[gd,row]] -> 0.999816	y[food[cgds,eu]] -> 1.00009
p[food[pd,row]] -> 0.999822	y[mnfcs[cgds,eu]] -> 1.00009
p[food[svces,row]] -> 0.999837	y[svces[cgds,eu]] -> 1.00009
p[food[food,row]] -> 0.999843	y[cgds[eu]] -> 1.00009
p[mnfcs[im,row]] -> 1.00013	y[svces[fd,usa]] -> 1.00008
p[food[cgds,row]] -> 0.999877	y[svces[pd,eu]] -> 0.999922
p[food[im,row]] -> 1.00012	y[svces[va,eu]] -> 1.00007
p[food[mnfcs,row]] -> 0.999892	y[food[svces,eu]] -> 1.00007
p[qt] -> 0.999893	y[mnfcs[svces,eu]] -> 1.00007
p[svces[im,usa]] -> 0.999907	y[svces[svces,eu]] -> 1.00007
p[mnfcs[im,usa]] -> 0.999937	y[svces[eu]] -> 1.00007
p[mnfcs[mnfcs,row]] -> 1.00004	y[mnfcs[pd,row]] -> 0.999935
p[mnfcs[cgds,row]] -> 1.00004	y[gd[row]] -> 0.999936
p[mnfcs[gd,row]] -> 1.00004	y[food[cgds,row]] -> 0.999939
p[labor[row]] -> 0.999966	y[mnfcs[cgds,row]] -> 0.999939
p[pd[row]] -> 0.999966	y[svces[cgds,row]] -> 0.999939
p[mnfcs[food,row]] -> 1.00003	y[cgds[row]] -> 0.999939
p[mnfcs[svces,row]] -> 1.00003	y[svces[gd,row]] -> 0.999939
p[mnfcs[pd,row]] -> 1.00003	y[svces[pd,row]] -> 0.999952
p[svces[va,row]] -> 0.999976	y[svces[fd,row]] -> 1.00004
p[mnfcs[va,row]] -> 0.999978	y[svces[va,row]] -> 1.00004
p[svces[cif,row,usa]] -> 0.999988	y[food[svces,row]] -> 1.00004
p[svces[cif,row,eu]] -> 0.999988	y[mnfcs[svces,row]] -> 1.00004
p[svces[row]] -> 0.999988	y[svces[svces,row]] -> 1.00004
p[svces[fd,row]] -> 0.999988	y[svces[row]] -> 1.00004
p[svces[cif,row,row]] -> 0.999988	y[pd[row]] -> 0.999964
p[capital[row]] -> 0.999989	y[food[pd,row]] -> 1.00003
p[mnfcs[row]] -> 1.00001	y[svces[va,usa]] -> 0.999978
p[mnfcs[fd,row]] -> 1.00001	y[food[svces,usa]] -> 0.999978
p[cgds[row]] -> 1.00001	y[mnfcs[svces,usa]] -> 0.999978
p[svces[cgds,row]] -> 0.99999	y[svces[svces,usa]] -> 0.999978
p[svces[gd,row]] -> 0.99999	y[svces[usa]] -> 0.999978
p[svces[pd,row]] -> 0.999994	
p[gd[row]] -> 0.999994	
p[svces[mnfcs,row]] -> 0.999996	
p[mnfcs[cif,row,eu]] -> 0.999998	u[usa] -> 1.00018
p[svces[food,row]] -> 0.999998	u[eu] -> 1.00001
p[mnfcs[cif,row,row]] -> 0.999998	u[row] -> 0.999955
p[mnfcs[cif,row,usa]] -> 0.999999	
p[svces[svces,row]] -> 1	
p[globalcgds] -> 1	

A6.7 – Escalamento dos dados do modelo GTAP3x3

Distribuição de frequência dos valores absolutos dos elementos do jacobiano

valor ≤ a	sem escalamento		com escalamento	
	quant.	%	quant.	%
1E-05	0	0,0	0	0,0
1E-04	3	0,3	0	0,0
1E-03	5	0,4	0	0,0
1E-02	31	2,7	0	0,0
1E-01	88	7,8	2	0,2
1E+00	430	38,0	0	0,0
1E+01	0	0,0	4	0,4
1E+02	9	0,8	22	1,9
1E+03	18	1,6	33	2,9
1E+04	64	5,6	124	10,9
1E+05	151	13,3	278	24,5
1E+06	211	18,6	419	37,0
1E+07	122	10,8	250	22,1
1E+08	1	0,1	1	0,1
>1E+08	0	0,0	0	0,0
	1133	100,0	1133	100,0

A7 – O pacote GTAPproc

```
(*  
Title: GTAPproc  
Author: Orlando Frederico Jose Godoy Bordoni  
Collaborator: Jose Antonio Scaramucci  
Version: 1.0  
Date: December 30, 2000  
Ementa: The GTAPproc package is an auxilliary procedure collection for  
        applied economic equilibrium analysis of GTAP models  
*)
```

```
(*BeginPackage["GTAPproc"];*)
```

```
$SectorOutput::usage=" ";  
$PrivateDemand::usage=" ";  
$PublicDemand::usage=" ";  
$RegionConsumption::usage=" ";  
$Depreciation::usage=" ";  
$Imports::usage=" ";  
$Exports::usage=" ";  
$GlobalTransport::usage=" ";  
$GlobalCGDS::usage=" ";  
$Shock::usage=" ";
```

```
(*Begin["`Private`"];*)
```

```
SetAttributes[p, Listable];  
cobbk=0.999;  
Print["Cobb-Douglas elasticity used in CES: ",cobbk];  
corte=N[10^-6];  
Print["Truncation value limit: ",corte];  
coefix= N[10^-11];  
Print["Non-CRS activity relative weight: ",coefix];
```

```
ClearAll[indice];  
indice[set_,elem_]:=Position[set,elem][[1,1]];
```

```
ClearAll[$DataPrep];  
$DataAgg[]:=  
Module[{rendas,despesas,surplus,ofer$prod,ofer$imp,dem$firm,dem$priv,  
        dem$gov,dem$exp,dem$transp,dem$inv,dem$depr,balanco},  
  subpar=Transpose[subpar];  
  subpar=(1-subpar);  
  incpar=Transpose[incpar];  
  esubp=(Times@@#&)/@((subpar^incpar)^(1/(Plus@@#&)/@incpar));  
  evfa=Transpose/@evfa;  
  evoa=Transpose[evoa];  
  vdfa=Transpose/@vdfa;  
  vdfm=Transpose/@vdfm;  
  vdga=Transpose[vdga];  
  vdgm=Transpose[vdgm];  
  vdpa=Transpose[vdpa];  
  vdpm=Transpose[vdpm];  
  vfm=Transpose/@vfm;
```

```

vifa=Transpose/@vifa;
vifm=Transpose/@vifm;
viga=Transpose[viga];
vigm=Transpose[vigm];
vims=Transpose/@vims;
vipa=Transpose[vipa];
vipm=Transpose[vipm];
viws=Transpose/@viws;
vst=Transpose[vst];
vxmd=Transpose/@vxmd;
vxwd=Transpose/@vxwd;

nE=Length[sE];Print["Endowment Commodities: ",nE];
nT=Length[sT];Print["Tradeable Commodities: ",nT];
nP=Length[sP];Print["Produced Commodities: ",nP];
nR=Length[sR];Print["Regions: ",nR];

voa=Apply[Plus,evfa,{2}] + Apply[Plus,vdfa,{2}] + Apply[Plus,vifa,{2}];
vom=Apply[Plus,Transpose/@vdfm,{2}] + vdpm + vdgm +
  Apply[Plus,Transpose[vxmd],{1}] + vst;
vom=Transpose[Join[Transpose[vom],Drop[Transpose[voa],nT]]];
rendas=(
  (*dotações - depreciação*)
  +Apply[Plus,Apply[Plus,evfa,{1}],{1}]-vdep
  (*imp/sub na produção de bens*)
  +Apply[Plus,vom-voa,{1}]
  (*imp/sub no valor agregado*)
  +Apply[Plus,Apply[Plus,evfa-vfm,{1}],{1}]
  (*imp/sub na importação das firmas*)
  +Apply[Plus,Apply[Plus,vifa-vifm,{1}],{1}]
  (*imp/sub na demanda interm. das firmas*)
  +Apply[Plus,Apply[Plus,vdfa-vdfm,{1}],{1}]
  (*imp/sub na importação privada*)
  +Apply[Plus,vipa-vipm,{1}]
  (*imp/sub no consumo privado*)
  +Apply[Plus,vdpa-vdpm,{1}]
  (*imp/sub na importação pública*)
  +Apply[Plus,viga-vigm,{1}]
  (*imp/sub no consumo público*)
  +Apply[Plus,vdga-vdgm,{1}]
  (*imp/sub na exportação de bens*)
  +Apply[Plus,Apply[Plus,vxwd-vxmd,{2}],{1}]
  (*imp/sub na importação de bens*)
  +Apply[Plus,Apply[Plus,vims-viws,{2}],{1}]);

despesas=(Apply[Plus,vipa+vdpa+viga+vdga,{1}]+save);
surplus=rendas-despesas;
Print["Surplus: ",Plus@@surplus];

(* balanço mundial de quantidades: *)
(* oferta: *)
ofer$prod = Apply[Plus,vom,{1}];
ofer$imp = Apply[Plus,Apply[Plus,vims,{1}],{1}];
(* demanda: *)
dem$firm = Apply[Plus,Apply[Plus,vdfm+vifm,{1}],{1}];
dem$priv = Apply[Plus,vdpm+vipm,{1}];
dem$gov = Apply[Plus,vdgm+vigm,{1}];

```

```

dem$exp = Apply[Plus,Apply[Plus,Transpose[vxmd],{1}],{1}];
dem$transp = Apply[Plus,vst,{1}];
dem$inv = (Flatten[Drop[Transpose[vom],nT]]-vdep);
dem$depr = vdep;

(* balanço: *)
balanco=ofer$prod+ofer$imp-dem$firm-dem$priv-
dem$gov-dem$exp-dem$transp-dem$inv-dem$depr;
(*
NumberForm[Transpose[Prepend[Transpose[
{ofer$prod,ofer$imp,-dem$firm,-dem$priv,
-dem$gov,-dem$exp,-dem$transp,
-dem$inv,-dem$depr,balanco}],
{"ofer$prod","ofer$imp","dem$firm",
"dem$priv","dem$gov","dem$exp","dem$transp",
"dem$inv","dem$depr","balanco"}]],{10,3}]
Print[balanco];
*)
];

ClearAll[$ativ];
$ativ[bf_List,br_,qaf_List,qmf_List,qar_,qmr_,sig_,ir_]:=
Module[{i,n,qx,px,qr,pr,sX,peso,coefs,dp,saco,conta,brx},

sX=bf; px=Chop[qaf,corte]; qx=Chop[qmf,corte]; pr=qar; qr=qmr;
n=Length[qx]; i=1;
If[n>0,
While[i<=n, (* Eliminar ramos com valor igual a zero *)
If[qx[[i]]==0.,
If[Head[sX[[i]]]!=land,
Print["Warning: Coefficient ",
sX[[i]]," in activity ",br," is zero"]];
sX=Drop[sX,{i}]; px=Drop[px,{i}]; qx=Drop[qx,{i}];
n--;
i++;
];
];
];
saco={};
If[qmr>0.,
i=1; n=Length[qx];
conta=0;
While[i<=n, (* Transformar em atividade de coef. *)
peso=qx/Plus@@qx; (* fixos (não-CRS) os ramos com peso *)
If[peso[[i]]<coefix, (* relativo menor que uma constante *)
brx=br; (* predefinida (coefix) *)
brx=ToExpression[ToString[Head[brx]]<>
"$"<>ToString[++conta]]@@brx;
dp={};
If[Abs[px[[i]]/qx[[i]]-1.]>corte,
AppendTo[dp,p[sX[[i]]]->px[[i]]/qx[[i]] p[sX[[i]]] ] ];
If[Abs[pr/qr-1.]>corte,
AppendTo[dp,p[br]->pr/qr p[br] ] ];
coefs={sX[[i]]->-qx[[i]],br->peso[[i]]*qr};
(*
ncrs1 -> {{a -> -1, b -> 1} -> cons},

```

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE


```

ncrs2 -> {{{(a -> -1, b -> 1) -> cons}, {p[a] -> 0.9 p[a]}}
*)
If[Length[dp]==0, AppendTo[saco,brx->{coefs->sR[[ir]]}];
Print["Attention: Leontieff activity ",
brx," created for ",sX[[i]]];
AppendTo[saco,brx->{{coefs->sR[[ir]]},dp}];
Print["Attention: Leontieff activity ",
brx," created for ",sX[[i]]];
];
sX=Drop[sX,{i}]; px=Drop[px,{i}]; qx=Drop[qx,{i}];
pr=(1.-peso[[i]]) pr; qr=(1.-peso[[i]]) qr;
n--;
';
i++;
];
];
If[n>0,
px=px/qx; pr=pr/qr;
If[sig==0. || n==1,
coefs=Join[Thread[sX->-qx], {br->qr}];
';
coefs=$CES[{sX->br}, Thread[{px, qx}],
{pr, qr}, {sig}, $OutputLevel->qr];
];
AppendTo[sX,br]; AppendTo[px,pr];
n++; i=1;
While[i<=n, If[Abs[px[[i]]]-1.]<corde,
sX=Drop[sX,{i}]; px=Drop[px,{i}]; n--, i++];
If[n==0, AppendTo[saco, br-> coefs],
AppendTo[saco, br-> {coefs,
Thread[p[sX]->px p[sX]]->sR[[ir]]}]];
];
];
];
If[Length[saco]==0,
Print["Warning: All coefficients in activity ",br," are zero"];
];
Return[saco];
];

ClearAll[$SectorOutput];
$SectorOutput[r_]:=Module[{i,c1,ic1,ic2,ir,qx,px,qr,pr,n,
sX,nX,sEx,nEx,sTx,sqx,arv,saco},
ir=indice[sR,r];
saco={};

(* Firms Domestic Intermediate Demand *)
svdfm=Apply[Plus,Transpose/@vdfm,{2}][[ir]];
For[ic=1,ic<=nT,ic++,
coefs={};
AppendTo[coefs, sT[[ic]][r]->-svdfm[[ic]]];
AppendTo[coefs, sT[[ic]][fd,r]->svdfm[[ic]]];
AppendTo[saco, sT[[ic]][fd,r]->coefs];
];

```

```

For[ic1=1,ic1<=nP,ic1++,
  cl=sP[[ic1]]; sqx={}; sTx={};

  (* Firms Primary Factor Demand *)
  If[cl!=Last[sP], (* Pular CGDS *)
    arv=#[r]&/@sE;
    qx=vfm[[ir,ic1]];px=evfa[[ir,ic1]];
    pr=Plus@@evfa[[ir,ic1]]; qr=Plus@@evfa[[ir,ic1]];
    AppendTo[sqx,pr]; AppendTo[sTx,cl[va,r]];
    saco=Join[saco,
      Flatten[$ativ[arv,cl[va,r],px,qx,pr,qr,esubva[[ic1]],ir]]];
  ];

  (* Firms Imports and Domestic Armington Aggregation *)
  For[ic2=1,ic2<=nT,ic2++,
    arv={sT[[ic2]][fd,r],sT[[ic2]][im,r]};
    qx={vdfm[[ir,ic1,ic2]],vifm[[ir,ic1,ic2]]};
    px={vdfa[[ir,ic1,ic2]],vifa[[ir,ic1,ic2]]};
    AppendTo[sqx,Plus@@px]; AppendTo[sTx,sT[[ic2]][cl,r]];
    qr=Plus@@px; pr=Plus@@px;
    saco=Join[saco,Flatten[$ativ[arv,sT[[ic2]][cl,r],
      px,qx,pr,qr,esubd[[ic2]],ir]]];
  ];

  (* Firms Inputs Leontief Aggregation *)
  pr=vov[[ir,ic1]];qr=vom[[ir,ic1]];
  saco=Join[saco,Flatten[$ativ[sTx,cl[r],sqx,sqx,pr,qr,0.,ir]]];
];
saco
];

ClearAll[$PrivateDemand];
$PrivateDemand[r_]:=Module[{ir,px,qx,pr,qr,sX,arv,saco},
  ir=indice[sR,r];

  (* Private Imports and Domestic Armington Aggregation *)
  saco={};
  For[ic1=1,ic1<=nT,ic1++,
    arv={sT[[ic1]][r],sT[[ic1]][im,r]};
    qx={vdpm[[ir,ic1]],vipm[[ir,ic1]]};
    px={vdpa[[ir,ic1]],vipa[[ir,ic1]]};
    qr=Plus@@px; pr=Plus@@px;
    saco=Join[saco,
      Flatten[$ativ[arv,sT[[ic1]][pd,r],
        px,qx,pr,qr,esubd[[ic1]],ir]]];
  ];

  (* Private Demand Aggregation *)
  arv=(#[pd,r]&)/@sT;
  px=vdpa[[ir]]+vipa[[ir]];
  qx=px;
  qr=Plus@@qx; pr=qr;
  saco=Join[saco,
    Flatten[$ativ[arv,pd[r],px,qx,pr,qr,esubp[[ir]],ir]];
    (*pd[r]->$CDE[pd[r],arv,Thread[{1,px}],[1,Plus@@qx],
      subpar[[ir]],incpar[[ir]]];*)
  ];
saco

```

```

];

ClearAll[$PublicDemand];
$PublicDemand[r_]:=Module[{i,ic1,ir,qx,px,qr,pr,arv,saco},
  ir=indice[sR,r];
  sacco={};

  (* Public Imports and Domestic Armington Aggregation *)
  For[ic1=1,ic1<=nT,ic1++,
    arv={sT[[ic1]][r],sT[[ic1]][im,r]};
    qx={vdgm[[ir,ic1]],vigm[[ir,ic1]]};
    px={vdga[[ir,ic1]],viga[[ir,ic1]]};
    pr=Plus@@px; qr=pr;
    sacco=Join[saco,
      Flatten[$ativ[arv,sT[[ic1]][gd,r],
        px,qx,pr,qr,esubd[[ic1],ir]]];
  ];

  (* Public Demand Aggregation *)
  arv=(#[gd,r]&)/@sT;
  px=vdga[[ir]]+viga[[ir]]; qx=px;
  pr=Plus@@px;qr=pr;
  sacco=Join[saco,Flatten[$ativ[arv,gd[r],px,qx,pr,qr,cobbk,ir]]]
];

ClearAll[$RegionConsumption];
$RegionConsumption[r_]:=Module[{ir,px,arv,alfas,saco,dotacoes},
  ir=indice[sR,r];
  arv={pd[r],gd[r],globalcgds};
  px={Plus@@vipa[[ir]]+Plus@@vdpa[[ir]],
    Plus@@viga[[ir]]+Plus@@vdga[[ir]],save[[ir]]};
  alfas=Thread[{1,px}];
  dotacoes=Thread[# [r] &/@sE->Apply[Plus,evfa,{1}][[ir]]];
  sacco={r->{dotacoes,$CES[arv,alfas,{cobbk},$OutputLevel->Plus@@px]}}
];

ClearAll[$Depreciation];
$Depreciation[r_]:=Module[{ir,saco},
  ir=indice[sR,r];
  sacco={depr[r]->{{cgds[r]->-vdep[[ir]]->r}};
  sacco
];

ClearAll[$Imports];
$Imports[r_]:=Module[{ic,ir,is,saco,sX,nX,qx,px,qr,pr},
  ir=indice[sR,r]; sacco={};
  For[ic=1,ic<=nT,ic++,
    sX={};nX=0;qx={};px={};
    For[is=1,is<=nR,is++,
      nX++;
      AppendTo[sX,sT[[ic]][cif,sR[[is]],r]];
      AppendTo[qx,vivs[[ir,is,ic]]];
      AppendTo[px,vims[[ir,is,ic]]];
    ];
  (* pr = qr = Plus@@px *)
  pr=Plus@@px;qr=pr;
  sacco=Join[saco,

```

```

                Flatten[ $\$ativ[sX, sT[[ic]][im, r],$ 
                        px, qx, pr, qr, esubm[[ic], ir]]];
];
saco
];

ClearAll[ $\$Exports$ ];
 $\$Exports[r_] := Module[{ic, is, ir, qr, sacco, coefs},
  ir = indice[sR, r];
  qr = 0;
  sacco = {};
  For[ic = 1, ic <= nT, ic++,
    For[is = 1, is <= nR, is++,
      If[vxmd[[is, ir, ic]] > corte,
        coefs = {};
        AppendTo[coefs, sT[[ic]][r] -> vxmd[[is, ir, ic]]];
        If[Abs[viws[[is, ir, ic]] - vxwd[[is, ir, ic]]] > corte,
          AppendTo[coefs, qt -> -(viws[[is, ir, ic]] - vxwd[[is, ir, ic]])];
        AppendTo[coefs, sT[[ic]][cif, r, sR[[is]]] -> viws[[is, ir, ic]]];
        If[Abs[vxwd[[is, ir, ic]]/vxmd[[is, ir, ic]] - 1] > corte,
          AppendTo[sacco, sT[[ic]][cif, r, sR[[is]]] ->
            {coefs, {p[sT[[ic]][r]] -> vxwd[[is, ir, ic]]/
              vxmd[[is, ir, ic]] p[sT[[ic]][r]] -> r}}];
          AppendTo[sacco, sT[[ic]][cif, r, sR[[is]]] -> coefs];
        ];
      ];
    ];
  sacco
];

ClearAll[ $\$GlobalTransport$ ];
 $\$GlobalTransport[] := Module[{ic, ir, qr, arv, sacco},
  arv = {}; qr = {};
  For[ir = 1, ir <= nR, ir++,
    For[ic = 1, ic <= nT, ic++,
      If[vst[[ir, ic]] > corte,
        AppendTo[arv, sT[[ic]][sR[[ir]]]];
        AppendTo[qr, vst[[ir, ic]]];
      ];
    ];
  ];
  sacco = {qt ->  $\$CES$ [{arv -> qt}, Thread[{1, qr}], {1, Plus@@qr}, {cobbk},
    $OutputLevel -> Plus@@qr]
];

ClearAll[ $\$GlobalCGDS$ ];
 $\$GlobalCGDS[] := Module[{ic, ir, qr, arv, sacco},
  arv = {}; qr = {};
  For[ir = 1, ir <= nR, ir++,
    AppendTo[arv, cgds[sR[[ir]]]];
    AppendTo[qr, voa[[ir, nP]] - vdep[[ir]]];
  ];
  sacco = {globalcgds ->
     $\$CES$ [{arv -> globalcgds}, Thread[{1, qr}], {1, Plus@@qr},
    {cobbk}, $OutputLevel -> Plus@@qr]
];
];$$$ 
```

```

ClearAll[{$Shock};
$Shock[obj_,bem_,ativ_,cons_,taxa_] :=
Module[{pos,ka,posb},
ka=0;
pos=Position[obj,ativ->{(__),_->}];
If[pos!={},
ka=1;
',
pos=Position[obj,ativ->{(__),{_->}}];
If[pos!={},
ka=1;
',
pos=Position[obj,ativ->{(__),{__}}];
If[pos!={},
ka=3;
',
pos=Position[obj,ativ->{__}];
If[pos!={},
ka=4;
',
Print["Erro: atividade ",ativ, " inexistente"];
Return[obj];
];
];
];
];
If[ka==1,
posb=Position[obj][[Sequence@@pos[[1]]],p[bem]->_ p[bem]];
If[posb!={},
pos={Join[pos[[1]],posb[[1]]];
ka=2;
];
];
If[ka==1,
pos={Join[pos[[1]],{2,2,1,1}];
Return[MapAt[{(Join[#, {p[bem]->taxa p[bem]})&},obj,pos]];
];
If[ka==2,
pos={Append[pos[[1]],2};
Return[MapAt[{(#/.p[bem]->taxa p[bem] &),obj,pos]];
];
If[ka==3,
pos={Join[pos[[1]],{2,2}];
Return[MapAt[{(Join[#, {(p[bem]->taxa p[bem])->cons})&},obj,pos]];
];
If[ka==4,
pos={Append[pos[[1]],2};
Return[MapAt[{(Join[#, {(p[bem]->taxa p[bem])->cons})&},obj,pos]];
];
Print["Erro: atividade ",ativ, " ka=",ka];
];
Null;
(*End[*];*)
(*EndPackage[*];*)

```

A8 – Tabelas do modelo GTAP8X9

A8.1 – Modelo de agregação usado com programa DATA-AGG

```
!
!   AGGREGATION TEMPLATE FOR GTAP DATABASE VERSION 4
!                               (18-Jun-98 BD)
!
! NOTES:
!
!   a. All lines starting with ! are ignored.
!
!   b. Empty lines are also ignored.
!
!   c. Ampersands "&" are used to separate "fields."
!
!   d. The lines starting "= = =" denote start and end of
!       aggregation sections.
!
!   e. In sections 1. and 3., a number of lines to be
!       filled in are provided. Additional lines can be
!       added as needed. Unused lines should be removed.
!
!   f. In the sample lines included below, the columns
!       to be filled in are indicated with a number of
!       carats "^" below the relevant part of the line.
!
!   g. Names for aggregated commodities/regions/factors can
!       include 12 characters. This can include letters, digits
!       and the characters @ and _ . The first character in the
!       name must be a letter.
!
! Send comments on the use of this form to dimaranan@agecon.purdue.edu
!
!
! 1. AGGREGATED COMMODITIES
!
!   In this section, the user fills in:
!
!       a) short names
!       b) (optional) explanatory text
!
!   Please note that in this section additional lines
!   can be added as needed and unused lines should be
!   removed.
!
!
!   Sample line follows:
!
!   AGR      & Food and Ag Includes onion pizzas
!   ^^^      ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
!
!   NAME      | Explanatory Text
!             | (Rest of line)
! (<=12 char) | ( Optional )
!
! = = = = =
! AGR      & Agricultura
! PEC      & Pecuaria
! OPRIM     & Pesca e atividade florestal
! MIN       & Mineracao (inclui petroleo e gas)
! ALIM      & Industria de alimentacao e couro.
! TRANSP    & Fabricacao de meios de transporte
! MAQ       & Maquinas e equipamentos
! MNFC      & Resto da ind. manufaturados
! SERV      & Resto da economia
!
! = = = = =
!
```

! 2. COMMODITY AGGREGATION MAPPING

!
! In this section the user fills in the short name of
! the appropriate aggregated commodity (from the previous
! list) for each GTAP commodity.
!

! Sample line follows:

! pdr, paddy rice & AGR
! ^^^

!
! GTAP COMMODITY : Aggregated
! : Commodity
! * * * * * DO NOT CHANGE THIS COLUMN * * * * * : (NAME)

```

=====
pdr, Paddy rice & AGR
wht, Wheat & AGR
gro, Cereal grains nec & AGR
v_f, Vegetables, fruit, nuts & AGR
osd, Oil seeds & AGR
c_b, Sugar cane, sugar beet & AGR
pfb, Plant-based fibers & AGR
ocr, Crops nec & AGR
ctl, Bovine cattle, sheep and goats, horses & PEC
oap, Animal products nec & ALIM
rmk, Raw milk & PEC
wol, Wool silk-worm cocoons & MNFC
for, Forestry & OPRIM
fsh, Fishing & OPRIM
col, Coal & MIN
oil, Oil & MIN
gas, Gas & MIN
omn, Minerals nec & MIN
cmt, Bovine cattle, sheep and goat, horse meat prods & ALIM
cmt, Meat products nec & ALIM
vol, Vegetable oils and fats & ALIM
mil, Dairy products & ALIM
pcr, Processed rice & ALIM
sgr, Sugar & ALIM
ofd, Food products nec & ALIM
b_t, Beverages and tobacco products & ALIM
tex, Textiles & MNFC
wap, Wearing apparel & MNFC
lea, Leather products & ALIM
lum, Wood products & MNFC
ppp, Paper products, publishing & MNFC
p_c, Petroleum, coal products & MIN
crp, Chemical, rubber, plastic products & MNFC
nmm, Mineral products nec & MNFC
i_s, Ferrous metals & MNFC
nfm, Metals nec & MNFC
fmp, Metal products & MNFC
mvh, Motor vehicles and parts & TRANSP
otn, Transport equipment nec & TRANSP
ele, Electronic equipment & MNFC
ome, Machinery and equipment nec & MAQ
omf, Manufactures nec & MNFC
ely, Electricity & SERV
gdt, Gas manufacture, distribution & SERV
wtr, Water & SERV
cns, Construction & SERV
t_t, Trade, transport & SERV
osp, Financial, business, recreational services & SERV
osg, Public admin and defence, education, health & SERV
dwe, Dwellings & SERV
=====

```

! 3. AGGREGATED REGIONS

!
! In this section, the user fills in:
!

```

!           a) NAMES
!           b) (optional) explanatory text
!
!   Please note that in this section additional lines
!   can be added as needed and unused lines should be
!   removed.
!
!
! Sample line follows:
!
!   LOW   & Low Income (GNP per capita under 2,000)
!   ^^^   ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
!
!           | Explanatory Text
!   NAME   | (Rest of line)
! (<=12 char)| ( Optional )
=====
ARG       & Argentina.
BRA       & Brasil.
CHI       & Chile.
NAFTA    & USA, Mexico e Canada.
RPM       & Complemento de NAFTA, ARG e BRA.
UE        & Uniao Europeia de 15.
ASIA     & Leste de Asia, China e India.
ROW      & Resto do mundo.
=====
!
! 4. REGIONAL AGGREGATION MAPPING
!
! In this section, the user fills in the short
! name for the aggregated region to which each
! of the GTAP regions belong.
!
! Sample line follows:
!
! IDN   Indonesia           & LOW
!           ^^^
!           GTAP REGION     | Aggregated
!           |               | Region
! * * DO NOT CHANGE THESE COLUMNS * * | (NAME)
=====
AUS   Australia           & ROW
NZL   New Zealand         & ROW
JPN   Japan               & ASIA
KOR   Republic of Korea   & ASIA
IDN   Indonesia           & ASIA
MYS   Malaysia            & ASIA
PHL   Philippines         & ASIA
SGP   Singapore           & ASIA
THA   Thailand            & ASIA
VNM   Viet Nam            & ASIA
CHN   China               & ASIA
HKG   Hong Kong           & ASIA
TWN   Taiwan              & ASIA
IND   India               & ASIA
LKA   Sri Lanka           & ASIA
RAS   Rest of South Asia   & ASIA
CAN   Canada              & NAFTA
USA   United States of America & NAFTA
MEX   Mexico              & NAFTA
CAM   Central America and Caribbean & RAM
VEN   Venezuela           & RAM
COL   Colombia            & RAM
RAP   Rest of Andean Pact  & RAM
ARG   Argentina           & ARG
BRA   Brazil              & BRA
CHL   Chile               & CHI
URY   Uruguay             & RAM
RSM   Rest of South America & RAM
GBR   United Kingdom       & UE
DEU   Germany             & UE

```



```

DNK   Denmark           &   UE
SWE   Sweden            &   UE
FIN   Finland           &   UE
REU   Rest of European Union &   UE
EFT   European Free Trade Area &   ROW
CEA   Central European Associates &   ROW
FSU   Former Soviet Union &   ROW
TUR   Turkey            &   ROW
RME   Rest of Middle East &   ROW
MAR   Morocco           &   ROW
RNF   Rest of North Africa &   ROW
SAF   South African Customs Union &   ROW
RSA   Rest of Southern Africa &   ROW
RSS   Rest of Sub Saharan Africa &   ROW
ROW   Rest of World     &   ROW

```

=====

! 5. AGGREGATED ENDOWMENTS

! In this section, the default endowment aggregation for GTAP
! Version 4 has been specified. Users may change the aggregation
! following these rules:

- ! a) One of the aggregated endowment names should be "capital"
! since this plays a key role in the GTAP model.
- ! b) Specify the classification of the aggregated factor,
! whether sluggish or mobile. There should be at least one
! sluggish and at least one mobile factor.
- ! c) Specify the value of the elasticity of transformation,
! etrae, for each aggregated sluggish endowment.

! Sample line follows:

```

! Capital      & sluggish      & -1.000      & Capital and NatRes
! ~~~~~~      ~~~~~~      ~~~~~~      ~~~~~~

```

! NAME	! Sluggish ! or Mobile	! etrae for ! sluggish	! Explanatory Text ! (Optional)
Land	& sluggish	& -1.000	& Land
Labor	& mobile	& ---	& unskilled labor
Capital	& mobile	& ---	& capital

! 6. ENDOWMENT AGGREGATION MAPPING

! In this section, the user fills in the short
! name for the aggregated endowment to which each
! of the GTAP endowments belong.

! Sample line follows:

```

! SIA   Skilled Labor           & LABOR
!                                     ~~~~~~
!           GTAP FACTOR           | Aggregated
!                                     | Factor
! * * DO NOT CHANGE THESE COLUMNS * * | (NAME)

```

LAN	Land	&	Land
ULA	UnSkLab	&	Labor
SLA	SkLab	&	Labor
CAP	Capital	&	Capital
NTR	NatRes	&	Capital

A8.2 – Listagem do experimento de simulação do GTAP8x9

```

In[1]:=
  SetOptions[$Output, PageWidth -> 90];
  SetDirectory["\\Pegasus"];
  <<Pegasus.s3
  Pegasus 1.2 (October 30, 1999)

In[4]:=
  arq="gtap8x94";

In[6]:=
  Get[arq<>".dat"];

In[7]:=
  Get["\\doc\\tese\\GTAPproc.m"];
  Cobb-Douglas elasticity used in CES: 0.999
                                     -6
  Truncation value limit: 1. 10
                                     -11
  Non-CRS activity relative weight: 1. 10

In[8]:=
  $DataPrep[]
  Endowment Commodities: 3
  Tradeable Commodities: 9
  Produced Commodities: 10
  Regions: 8
  Surplus: -1.81046

In[9]:=
  prod[0]={};
  For[i=1,i<=nR,i++, r=sR[[i]];
    prod[0]=
      Join[prod[0],
          $SectorOutput[r],
          $PrivateDemand[r],
          $PublicDemand[r],
          $Imports[r],
          $Exports[r],
          $Depreciation[r]
        ];
  ];
  prod[0]=
    Join[prod[0],
        $GlobalTransport[],
        $GlobalCGDS[]
    ];
  Warning: Coefficient pec[cif, asia, arg] in activity pec[im, arg] is zero
  Warning: Coefficient pec[cif, asia, bra] in activity pec[im, bra] is zero
  Warning: Coefficient pec[cif, asia, chi] in activity pec[im, chi] is zero
  Warning: Coefficient pec[cif, bra, asia] in activity pec[im, asia] is zero
  Warning: Coefficient pec[cif, chi, asia] in activity pec[im, asia] is zero
  Warning: Coefficient pec[cif, ram, asia] in activity pec[im, asia] is zero

In[12]:=
  cons[0]={};
  For[i=1,i<=nR,i++, r=sR[[i]];
    cons[0]=
      Join[cons[0],
          $RegionConsumption[r]
        ];
  ];
  ];

```

```

In[14]:=
modelo[0]=FindEquilibrium[
    $Production->prod[0],
    $Consumption->cons[0],
    $Numeraire->globalcgds,$Report->None,$Solver->ACI
];
$Time{modelo[0]}
Out[13]=
{2637.02 Second, 43.9503 Minute}

In[20]:=
Put[{$Solution[modelo[0]],"8x9bra0.sol"}]

In[21]:=
$Time{modelo[0]}
Out[21]=
{2637.02 Second, 43.9503 Minute}

In[22]:=
(* choque BAL/imports modelo 8x9 -> expr=bra *)
expr=bra;
iexpr=Flatten[Position[sR,expr]][[1]];
prod[1]=prod[0];
For[r=1,r<=nR,r++, impR=sR[[r]];
  For[i=1,i<=nT,i++, tradcom=sT[[i]];
    If[Chop[viws[[r,iexpr,i]],corte]>0.,
      prod[1]=
        $Shock[prod[1],tradcom[cif,impR,expr],tradcom[im,expr],expr,n[restricao]]; ,
      Print["Warning: Coefficient ",tradcom[cif,impR,expr],
        " in activity ",tradcom[im,expr]," is zero"];
    ];
  ];
];
cons[1]=cons[0];
prod[1]=Evaluate[prod[1]];
Warning: Coefficient pec[cif,asia,bra] in activity pec[im,bra] is zero

In[32]:=
(* restrição BAL/imports modelo 8x9 *)
qex={}; pex={}; qim={}; pim={};
For[r=1,r<=nR,r++, impR=sR[[r]];
  For[i=1,i<=nT,i++, tradcom=sT[[i]];
    qex=Join[qex,{-Quantity[tradcom[expr],tradcom[cif,expr,impR],$prod]}];
    pex=Join[pex,{p[tradcom[expr]]}];
    qim=Join[qim,{-Quantity[tradcom[cif,impR,expr],tradcom[im,expr],$prod]}];
    pim=Join[pim,{p[tradcom[cif,impR,expr]]}];
  ];
];
bal=Plus@@(pex qex)-Plus@@(pim qim);
bal0=Evaluate[(((bal/. $prod->$Production[prod[0]])
  (*/. $cons->$Consumption[cons[0]]*))
  /. $Solution[modelo[0]]]
ball=Evaluate[((bal/. $prod->$Production[prod[1]])
  (*/. $cons->$Consumption[cons[1]]*))];
(* ball <= 0.75 bal0 -> 0 <= 0.75 bal0 - ball -> 0.75 bal0 - ball >= 0 *)
rest[1]=restricao->0.75 bal0 - ball;
Out[38]=
-14703.6

```

```

In[42]:=
modelo[1]=FindEquilibrium[
    $Production->prod[1], $Consumption->cons[1],
    $Constraints->rest[1], $M0->{m[restricao]->.2},
    $Numeraire->globalcgds, $Report->None, $Trace->True, $Solver->ACI];

Constructing NLCP ...

OK (1195.89 Second)

Normalization: p[globalcgds] -> 1

Solving ACI ...
Compiling f ...
OK (1335.51 Second)
Solving problem ...
avaliacao f.....:      0.1100 segundos
construcao jacobiano:    7.3600 segundos
fatoracao matriz.....:  5.2800 segundos
Harwell MA28 statistics:
  n      = 3525
  nz     = 21525
  mtype  = 1
  tol    = 0.000000
  ndrcp  = 0
  u      = 0.100000
  lblock = 1
  grow   = 1
  w(1)   = 1829370580012593.200000
  lbig   = 0
  big    = 268504055
  abort1 = 1
  abort2 = 1
  abort3 = 0
  irank  = 3525
  themax = 10246217.304841
  num    = 1
  large  = 3525
  resid  = 0.000000
  iflag  = 0
0, jk= 2177, teta= 1.000000, fkmax= 3448.549685
1, jk= 2177, teta= 1.000000, fkmax= 751.141084
2, jk= 2177, teta= 1.000000, fkmax= 299.795459
3, jk= 198, teta= 1.000000, fkmax= 52.262166
4, jk= 198, teta= 1.000000, fkmax= -18.944148
5, jk= 198, teta= 1.000000, fkmax= -86.400048
6, jk= 198, teta= 1.000000, fkmax= -36.130543
7, jk= 198, teta= 1.000000, fkmax= 58.906246
8, jk= 198, teta= 1.000000, fkmax= 63.384979
9, jk= 198, teta= 1.000000, fkmax= -27.146532
10, jk= 198, teta= 1.000000, fkmax= 6.581251
11, jk= 198, teta= 1.000000, fkmax= 0.720818
12, jk= 198, teta= 1.000000, fkmax= -0.239357
13, jk= 198, teta= 1.000000, fkmax= -0.115630
14, jk= 198, teta= 1.000000, fkmax= -0.442895
15, jk= 198, teta= 1.000000, fkmax= -0.083468
16, jk= 198, teta= 1.000000, fkmax= -0.085033
17, jk= 198, teta= 1.000000, fkmax= -0.095319
18, jk= 198, teta= 1.000000, fkmax= -0.024600
19, jk= 198, teta= 1.000000, fkmax= 0.023734
20, jk= 198, teta= 1.000000, fkmax= 0.146872
21, jk= 198, teta= 1.000000, fkmax= -0.027284
22, jk= 198, teta= 1.000000, fkmax= 0.054664
23, jk= 198, teta= 1.000000, fkmax= 0.019876

```

```

24, jk= 198, teta= 1.000000, fmax= -0.000846
tempo das iteracoes.: 5.4300 segundos
tempo total ACI.....: 19.2200 segundos
OK (20.27 Second)
OK (1357.81 Second)

```

Equilibrium found by ACI

Total time elapsed: 2567.33 Second/42.7888 Minute

```

In[47]:=
Put[Solution[modelo[1]],"8x9bra1.sol"];
m[restricao]/.Solution[modelo[1]]
Out[47]=
m[restricao] -> 1.207452

In[48]:=
bal0
bal1/.Solution[modelo[1]]
prod0=$Production[prod[0]]/.Solution[modelo[0]];
cons0=$Consumption[cons[0]]/.Solution[modelo[0]];
prod1=$Production[prod[1]]/.Solution[modelo[1]];
cons1=$Consumption[cons[1]]/.Solution[modelo[1]];
s0=($Quantity[globalcgds,#,cons0] p[globalcgds]e/@sR)/.Solution[modelo[0]];
i0=($Quantity[cgds[#],globalcgds,prod0] p[cgds[#]]e/@sR)/.Solution[modelo[0]];
{s0,i0,s0-i0}
Plus@@(s0-i0)
s1=($Quantity[globalcgds,#,cons1] p[globalcgds]e/@sR)/.Solution[modelo[1]];
i1=($Quantity[cgds[#],globalcgds,prod1] p[cgds[#]]e/@sR)/.Solution[modelo[1]];
{s1,i1,s1-i1}
Plus@@(s1-i1)
Out[48]=
-14703.6
Out[49]=
-11027.7
Out[56]=
{{8435.36, 55282.2, 9987.57, 537569., 18305.1, 598232., 1.35623*10^6, 209994.},
 {8579.77, 65076.5, 9703.35, 627976., 30431.5, 559048., 1.28213*10^6, 211093.},
 {-144.414, -9794.25, 284.213, -90407.6, -12126.4, 39184.4, 74103., -1098.86}}
Out[57]=
1.31773*10^-6
Out[60]=
{{8333.88, 58360.5, 9929.07, 536505., 18237.9, 597114., 1.35406*10^6, 209607.},
 {8573.91, 65037.1, 9696.77, 627551., 30410.9, 558669., 1.28126*10^6, 210950.},
 {-240.029, -6676.64, 232.296, -91045.5, -12172.9, 38445.3, 72800.4, -1342.87}}
Out[61]=
0.000539735

In[83]:=
Plus@@{8435.36, 55282.2, 9987.57, 537569., 18305.1, 598232., 1.35623*10^6, 209994.}
Plus@@{8579.77, 65076.5, 9703.35, 627976., 30431.5, 559048., 1.28213*10^6, 211093.}
Plus@@{-144.414, -9794.25, 284.213, -90407.6, -12126.4, 39184.4, 74103., -1098.86}
Out[83]=
2.79404*10^6
Out[84]=
2.79404*10^6
Out[85]=
1.31773*10^-6

In[90]:=
Plus@@{8333.88, 58360.5, 9929.07, 536505., 18237.9, 597114., 1.35406*10^6, 209607.}
Plus@@{8573.91, 65037.1, 9696.77, 627551., 30410.9, 558669., 1.28126*10^6, 210950.}
Plus@@{-240.029, -6676.64, 232.296, -91045.5, -12172.9, 38445.3, 72800.4, -1342.87}
Out[80]=
2.79215*10^6
Out[81]=
2.79215*10^6
Out[82]=
0.000539735

```

```

In[63]:=
pfac={};
qfac={};
For[r=1,r<=nR,r++,
  AppendTo[pfac,(p[#{sR[[r]]}]&/@sE)/.$Solution[modelo[1]]];
  qland=($Quantity[land[sR[[r]]],#{va,sR[[r]],prod1}&/@sT)/.
    $Solution[modelo[1]];
  qlabor=($Quantity[labor[sR[[r]]],#{va,sR[[r]],prod1}&/@sT)/.
    $Solution[modelo[1]];
  qcapital=($Quantity[capital[sR[[r]]],#{va,sR[[r]],prod1}&/@sT)/.
    $Solution[modelo[1]];
  AppendTo[qfac,{-Plus@@qland,-Plus@@qlabor,-Plus@@qcapital}];
];

```

```

In[66]:=
pfac
Out[66]=
{{0.994279, 0.988662, 0.989036}, {1.03101, 1.04702, 1.04657},
 {0.994352, 0.994477, 0.994459}, {1.00052, 0.998035, 0.99804},
 {0.99587, 0.996383, 0.996458}, {1.00246, 0.998147, 0.998154},
 {0.999257, 0.998373, 0.998375}, {1.00042, 0.998144, 0.998098}}

```

```

In[67]:=
qfac
Out[67]=
{{8369.22, 127023., 119219.}, {9240.3, 289529., 333400.}, {1245.29, 20564.4,
 33737.6},
 {35758.4, 4.59133*10^6, 2.7201*10^6}, {10884., 120651., 178333.},
 {23200.5, 5.10616*10^6, 2.52361*10^6}, {153653., 3.7285*10^6, 3.00926*10^6},
 {49768.4, 1.49814*10^6, 1.21802*10^6}}

```

```

In[68]:=
pfacg=Flatten[pfac];
qfacg=Flatten[qfac];
pfacgm=Plus@@(qfacg pfacg)/Plus@@qfacg
Out[70]=
0.99925

```

```

In[71]:=
pfacr=(Plus@@#&/@{pfac qfac})/(Plus@@#&/@qfac)
pfacr/pfacgm
Out[71]=
{0.998693, 1.04655, 0.994463, 0.998049, 0.996408, 0.998162, 0.998394, 0.998164}
Out[72]=
{0.989435, 1.04733, 0.995209, 0.998797, 0.997155, 0.998911, 0.999143, 0.998913}

```

```

In[73]:=
$IncomeVariation[modelo[0]]/$Solution[modelo[1]]
Out[73]=
{arg -> -662.665, bra -> -634.989, chi -> -152.395, nafta -> -1910.68,
 ram -> -456.832, ue -> -2091.44, asia -> -2557.73, row -> -825.425}

```

A8.3 – E-mail aos participantes do GTAPLIST

>From owner-gtap-l@VM.CC.PURDUE.EDU Thu Mar 11 11:14 EST 1999
Date: Thu, 11 Mar 1999 09:10:41 EST
From: "Thomas W. Hertel" <Hertel@AGECON.PURDUE.EDU>
Subject: Re: Exchange rates
To: Multiple recipients of list GTAP-L <GTAP-L@VM.CC.PURDUE.EDU>
X-Status:
X-Keywords:
X-UID: 346

Dear James,

The question which you posed to GTAP-L definitely qualifies as a frequently-asked question. Indeed, it comes up at virtually every short course. Therefore, even though I know you have already gotten some direct responses from individuals, I will respond to the list as well.

The question follows:

> Can anyone provide me with advice on how to make GTAP mimic a change in exchange rates? For example how would you impose a 10% depreciation of the Canadian dollar against all other currencies?
>
> James Rude
> Agriculture and Agrifood-Canada

Here is my response.

1. GTAP, like most other CGE models, is a real model with no money. Therefore there is no nominal exchange rate which can be directly linked to those observed in the foreign exchange markets.
2. However, GTAP does have a mechanism for restoring external balance after a shock. In particular, the return to primary factors in each region will adjust, either: (a) downward to promote exports and choke off import competition, or (b) upward to restrict exports and encourage imports. Therefore, the relative price of primary factors across regions is a logical "real exchange rate" use.

By way of example, I use the version of RunGTAP from the web and choose the following:

- book3x3 version
- NEWMERGE closure so that the numeraire is the average return to primary factors, world-wide
- shock tms(food,usa,eu) = -10 so that I am reducing the bilateral tariff on food from USA to Eu
- solution method: Gragg 2-4-6

Then I get the following results for the variable pfactor(r) which measures the change in each region-specific primary factor price index, relative to the global average pfactwld (which is the numeraire).

pfactor	%chg	PreLev	PostLev	AbsChng
USA	0.12837	1.00000	1.00128	0.00128
EU	-0.066641	0.00000	0.99933	-0.00067
ROW	-0.028781	0.00000	0.99971	-0.00029

You can see that the relative price of primary factors in the USA rose to restore external balance. This is because, with savings and investment largely unaffected, the increased demand for USA exports creates a disequilibrium -- provided imports are also unchanged. To restore equilibrium we require a real appreciation of primary factor prices in USA. This is very effective in restoring external balance, since it works to reduce exports of other products (by raising their price) and it also works to increase imports of competing goods as domestic prices rise. Of course, the opposite must occur in the EU, where the increase in imports has created an external imbalance at initial levels of exports, savings and investment. Therefore, the US experiences a real appreciation of primary factor returns, relative to the EU (as well as relative to ROW which has its demand for exports curbed due to the diversion of food imports into the EU from ROW to USA).

3. Now to your question of mimicing observed changes in exchange rates -- Since GTAP does not have a nominal exchange rate -- and the real exchange rate as defined above is normally endogenous, there is no direct means to achieve your goal. However, this turns out to be a strength, not a limitation. The fact is, that the impact of a Canadian devaluation will depend very much on the root cause of this devaluation. Is it due to a decline in relative factor productivity? Or is it due to capital flight in the wake of increased risk premia on investment in Canada? You have to be more specific in the question which you seek to answer. By specifying the experiment in terms of the fundamentals you will deliver much better analysis.

Of course, there are times when you just want to "shock the model" -- or perhaps you can't model the underlying phenomena in GTAP. Then you are better off taking a dual approach to the exchange rate and shocking the trade balance. (Swap $cgdslack(r)$ with $DTBAL(r)$ and shock $DTBAL(r)$.) This will force the same kind of adjustment in the real exchange rate.

I hope that this is helpful.

Sincerely,

Tom Hertel