



PAEG “Hands On”

Instalação e Execução do Programa PAEG

Ângelo Costa Gurgel

PAEG Technical Paper No. 5

Versão revisada em Dezembro 2013

PAEG “Hands On”**Instalação e Execução do Programa PAEG**

Ângelo Costa Gurgel

1. Introdução

O objetivo deste documento é auxiliar na instalação do programa PAEG 3.0 e introduzir o leitor na implementação do modelo através de exercícios e exemplos práticos, de forma a familiarizar o pesquisador com o ambiente de programação GAMS, no qual o PAEG foi construído, e com a própria implementação do modelo.

Este documento foi formulado para ser utilizado no quinto minicurso do PAEG, realizado em dezembro de 2013 em Viçosa-MG, e atualizado para atender aos propósitos das edições mais recentes do minicurso.

2. CD-ROM de distribuição do PAEG

Os arquivos necessários para instalação e implementação do modelo estão armazenados em um CD-ROM de distribuição. As pastas e arquivos no CD-ROM são:

a) Pasta *Instalação*, com os arquivos e programas:

- windows_x64_64.exe (instala o programa GAMS na sua versão 23.9.5)¹;
- gamslice.txt (arquivo com a licença do GAMS)²; e,
- inclib.pck (contém programas auxiliares do GAMS).

b) Pasta *Exercicios_MPSGE*, com os arquivos:

- m1_mcp.gms (arquivo GAMS para exercícios); e,
- m1_mps.gms (arquivo GAMS para exercícios).

c) Pasta *PAEG_2013_minicurso*, que contém os arquivos com os dados e o modelo PAEG no formato do GTAP8inGAMS com uma agregação menor de regiões e setores para implementação no minicurso.

d) Pasta *PAEG_2013*, que contém a distribuição do modelo PAEG (desagregado em maior número de regiões e setores); e,

¹ O software GAMS pode ser baixado também através do link: <<http://www.gams.com/download/>>. No caso de sistemas 32 bits, o arquivo será o windows_x86_32.exe.

² Neste caso, a licença do PAEG Research Group, para uso específico em trabalhos associados ao PAEG.

e) Pasta *Documentos*, com diversos documentos sobre o PAEG, o MPSGE, o GTAPinGAMS, bem como este *Hands On*.

3. Instalação

O modelo PAEG foi construído no software *General Algebraic Modeling System – GAMS*, idealizado para o desenvolvimento e solução de modelos de otimização. Para que o software GAMS e o modelo PAEG possam ser utilizados, alguns passos devem ser seguidos para instalação dos programas:

- a) Primeiro, deve-se instalar o programa Gams. Copie a pasta de instalação para o seu computador e clique no programa `windows_x86_32.exe`, localizado dentro da pasta. O processo de instalação cria um diretório GAMS que contém, além do programa GAMS, arquivos de documentação e uma biblioteca de modelos já construídos. Ao final da instalação, uma mensagem aparece solicitando a cópia da licença do programa. Clicando em “Yes” o programa solicita a seleção da pasta que contém a licença. A pasta com a licença é a pasta “Instalação” do CD-ROM. Selecione esta pasta e clique “OK”. O programa efetuará a cópia do arquivo de licença (`gamslice.txt`) e terminará a instalação. Sem a licença o programa opera no modo *demo*, que limita o tamanho dos modelos que podem ser solucionados.
- b) Com o programa instalado, é necessário adicionar os programas auxiliares ao GAMS. Para tal, copie o arquivo `inclub.pck` localizado na pasta “Instalação” para a pasta do GAMS recém instalada no diretório `c:\` (pasta `c:\GAMS\win64\23.9`). Abra essa pasta e execute o arquivo `gamsinst.exe`, clicando duas vezes sobre esse arquivo e digitando “y”, seguido de enter, quando a primeira mensagem aparecer³. Durante esse processo, o programa deve perguntar quais *solvers* serão utilizados para resolver cada tipo de programa. Clique na tecla enter para que o solver padrão seja selecionado em cada caso.
- c) Para verificar se o processo de instalação foi realizado com sucesso, abra o programa GAMS-IDE (o editor de texto padrão do GAMS⁴) na pasta GAMS (o processo de instalação costuma criar uma tecla de atalho no desktop para esse programa). Abra o modelo *TRANSMCP* a partir da biblioteca de modelos do GAMS (File/Model

³ Para executar o `gamsinst.exe` no Windows 7, primeiro desative o controle da conta de usuário (UAC) por meio do <Painel de Controle\Contas de Usuário e Segurança Familiar\Contas de Usuário>.

⁴ Para visualizar um arquivo ou modelo escrito na syntax do software GAMS deve-se utilizar um editor de texto. O GAMS distribui o seu próprio editor de texto, o GAMS-IDE, com a instalação do programa. O GAMS-IDE é de fácil utilização e pode ser usado para implementar o PAEG.

Library/Open GAMS Model Library), e click no botão vermelho (Run GAMS – F9) para executar o modelo. Uma janela perguntando pela escolha dos solvers deve aparecer, clique em “yes” e em “OK”. Se o processo de instalação estiver correto, uma janela com os passos da execução irá aparecer (arquivo *transmcp.log*), e um arquivo com extensão *lst* (arquivo *transmcp.lst*) com os resultados será criada na mesma janela do arquivo de extensão *gms*. O arquivo *gms* contém o código do programa, o arquivo *log* apresenta a execução do programa e o arquivo *lst* mostra os resultados da execução do programa. Se a instalação estiver correta, deve aparecer uma mensagem ao final do arquivo de extensão *log* do tipo:

```
*** Status: Normal completion
```

4. Entendendo a Sequência de Execução do PAEG

A sequência de execução do PAEG será apresentada de maneira prática, estimulando o leitor a seguir essa sequência na medida em que trabalha com o modelo.

4.1 Criando um novo projeto

Copie a pasta *PAEG_2013_minicurso* do CD-ROM de distribuição para uma pasta qualquer do seu computador (atenção: não copie para a pasta GAMS recém-instalada no computador, mantenha os projetos em outra pasta/diretório). Dentro da pasta *PAEG_2013_minicurso* estão localizados os arquivos necessários para entender a sequência de execução e implementar o modelo.

Para trabalhar com o editor de texto GAMS-IDE, é necessário sempre iniciar um novo projeto, ou abrir um já criado. Então, abra o programa GAMS-IDE (se você ainda não o tiver aberto) e crie um novo projeto (File/Project/New Project), salvando-o com o nome de “minicurso” dentro da pasta *PAEG* (a mesma onde está localizado o arquivo *paeg_model.gms*) do seu computador. Sempre que criar um novo projeto, procure salvá-lo em uma pasta diferente da pasta do GAMS, como por exemplo, dentro da pasta *Meus Documentos*. Executar projetos dentro da pasta onde o GAMS está instalado pode gerar erros de execução difíceis de serem identificados.

4.2 Entendendo a sequência de operacionalização do modelo

O PAEG está organizado em diferentes pastas e arquivos, de forma a separar as etapas da criação da base de dados da implementação do modelo de equilíbrio geral. A pasta

GTAP8inGAMS contém o programa *GTAPinGAMS* na sua oitava versão⁵, bem como a base de dados das regiões internacionais que fazem parte da agregação em uso, que no caso, são dados do *GTAP* na sua oitava versão, referentes à economia mundial para o ano de 2007. A pasta *PAEG* contém o modelo de equilíbrio geral do PAEG (no arquivo *paeg_model.gms*). Ainda, alguns outros arquivos estão presentes na pasta *PAEG*. Esses arquivos são responsáveis por diferentes etapas da leitura das bases de dados das matrizes de insumo-produto regionais brasileiras e das demais regiões, compatibilização dessas bases de dados, e leitura da base de dados compatibilizada pelo modelo de equilíbrio geral.

A primeira implementação do modelo PAEG requer a execução de diversos arquivos diferentes para gerar os dados agregados e em equilíbrio. Após a primeira implementação, a base de dados já estará disponível no diretório, sendo necessário apenas executar o arquivo *paeg_model.gms* em futuras simulações. A Figura 1 apresenta o nome e a função dos arquivos necessários para a primeira implementação do PAEG. A sequência correta de execução do PAEG, quando da primeira vez de sua implementação, será descrita a seguir. Para melhor entendê-la, siga o texto e execute ao mesmo tempo os passos descritos.

Primeiro, abra o arquivo *pre_dados.gms*. Este arquivo lê os dados das matrizes brasileiras regionais e os armazena em um arquivo de extensão *gdx*, próprio para armazenamento de dados no GAMS a partir de arquivos do excel. Execute este arquivo, clicando no botão com uma seta vermelha (Run GAMS), ou clicando F9 (lembre-se que você deverá ter criado previamente um novo projeto dentro da pasta *PAEG*). Se a execução ocorrer normalmente, a seguinte mensagem deverá ter aparecido na janela de execução:

```
*** Status: Normal Completion
```

Segundo, abra o arquivo *dados.gms*. Este arquivo lê os dados das matrizes regionais brasileiras, lê os dados das demais regiões (internacionais), compatibiliza as duas fontes de dados, lê um arquivo de desagregação dos dados de impostos nas regiões brasileiras (*paeg_tax.dat*), e armazena o resultado dessas etapas em um arquivo de extensão *dat*, capaz de ser lido pelo GAMS. Execute o arquivo e verifique se tal execução foi correta (veja se a mensagem “*** Status: Normal Completion” aparece).

⁵ O modelo *GTAP8inGAMS* foi disponibilizado junto com o PAEG para que o pesquisador que tenha interesse em aprender mais sobre a modelagem de equilíbrio geral em GAMS possa estudar e utilizar esse programa na medida em que adquirir habilidades na modelagem com a implementação do PAEG.

Diretório GTAPinGAMS

Diretório PAEG

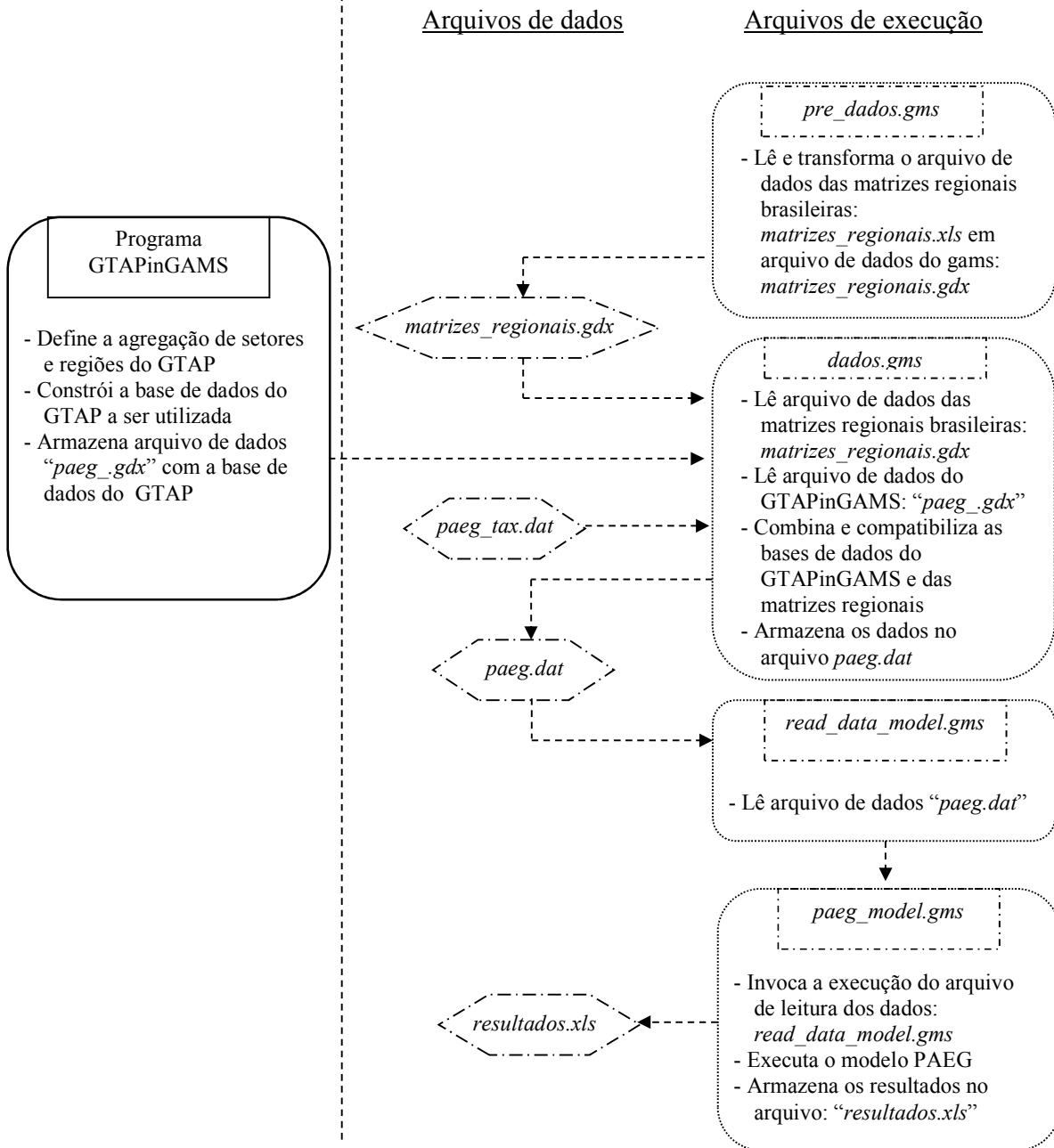


Figura 1 - Diagrama de Fluxo de Funcionamento do Modelo PAEG

Agora, as bases de dados estão prontas. Abra o arquivo *paeg_model.gms*. Esse arquivo contém o modelo de equilíbrio geral do PAEG no subsistema MPSGE. Esse arquivo invoca, durante a sua execução, o arquivo *read_data_model.gms*, responsável pela leitura dos dados. Execute o arquivo *paeg_model.gms* e verifique se sua execução foi correta.

Como dito anteriormente, após a primeira sequência completa de execução dos arquivos não se faz mais necessário a execução de todos os arquivos como descrito acima, mas apenas do arquivo *paeg_model.gms*. A seguir, detalha-se o código do modelo *paeg_model.gms* e como realizar simulações de políticas com o modelo.

5. Entendendo o código do Modelo PAEG

Para visualizar o modelo escrito na sintaxe do software GAMS deve-se abrir o arquivo *paeg_model.gms* no editor de texto GAMS-IDE (se você estiver seguindo o tópico anterior, o arquivo já estará aberto. Nesse caso, feche os demais arquivos abertos de extensão *gms* e *lst*, bem como a janela de *log*). Acompanhe no arquivo os vários trechos do programa descritos a seguir.

O arquivo *paeg_model.gms* começa com a declaração de um título (\$title). Em seguida, aparece a linha de comentário:

```
* Include sets parameters and data:
```

Sempre que aparecer um asterisco na primeira coluna de uma linha, o programa GAMS considerará o restante daquela linha como um comentário. Isto significa que aquela linha não será processada quando da execução do programa. A próxima declaração no arquivo é:

```
$include read_data_model.gms
```

Essa declaração informa ao programa GAMS que o arquivo *read_data_model.gms* deve ser incluído durante a execução do programa. O arquivo *read_data_model.gms* é responsável pela leitura dos dados do PAEG. Para visualizar este arquivo, abra-o no GAMS-IDE.

O arquivo *read_data_model.gms* define primeiro os conjuntos de setores, fatores primários e regiões do modelo, utilizando o termo “set”. Dessa forma, o conjunto de setores do modelo é representado pela letra *i*, o de fatores pela letra *f*, e o de regiões pela letra *r*. Alguns outros conjuntos e subconjuntos são definidos, e o termo “alias” define equivalência

de símbolos para representação de um mesmo conjunto. Por exemplo, o conjunto de regiões pode ser representado não apenas pela letra *r*, mas também pela letra *s*.

Após a declaração de conjuntos, o arquivo *read_data_model.gms* apresenta uma declaração para leitura dos dados:

```
$include paeg.dat
```

O comando `$include` orienta o programa GAMS a incluir algum outro arquivo durante a execução do programa. O arquivo *paeg.dat* contém a base de dados do PAEG, previamente construída pela compatibilização dos dados das matrizes regionais do Brasil com os dados do GTAP.

Após a inclusão da base de dados, vários parâmetros são declarados e calculados a partir dos dados originais do PAEG. O nome de cada parâmetro é determinado pela declaração *parameter*. Esses parâmetros são calculados com base nos dados do PAEG previamente construídos e incluídos no arquivo *paeg.dat*. Apresenta-se a seguir a simbologia dos parâmetros da base de dados com a sua descrição. Os valores desses parâmetros estão armazenados no arquivo Excel *paeg_3x8.xls*, criado quando da execução do arquivo *dados.gms*. Os dados de fluxos estão em bilhões de dólares, enquanto os impostos e subsídios são taxas *ad valorem* apresentadas como fração⁶.

Quando se executa o arquivo *paeg_model.gms* no GAMS, o arquivo *read_data_model.gms* será incluído, permitindo assim a leitura da base de dados e o cálculo dos diversos parâmetros declarados naquele arquivo. Findas as tarefas determinadas no arquivo *read_data_model.gms*, o GAMS continuará a executar o arquivo *paeg_model.gms*.

Retorne agora para o código do arquivo *paeg_model.gms*, para continuar observando sua sequência de execução. As próximas linhas do arquivo criam um parâmetro (*mobf*) e um conjunto (*mobfd*). Estes *mobf* e *mobfd* controlam a possibilidade de movimento de fatores (capital e trabalho) entre as regiões brasileiras. Quando *mobf* for igual a 1, capital e trabalho podem deslocar-se entre regiões dentro do país, de forma que, caso um choque qualquer tenha provocado mudanças nas remunerações dos fatores, as diferenças no retorno ao capital e no salário entre regiões sejam diminuídos. Quando *mobf* for igual a 0, cada região é considerada como um mercado segmentado de fatores, com possibilidades de grandes diferenças nas remunerações dos mesmos em relação às demais regiões dentro do país.

⁶ Por exemplo, se um bem qualquer em uma região possui um imposto de 10% ao ser consumido, o arquivo *paeg_gtap.xls* apresentará o valor 0,1 para este imposto.

Na sequência do arquivo *paeg_model.gms*, o código do PAEG em MPSGE é o próximo a ser lido. A Tabela 1 apresenta a simbologia dos diversos parâmetros que armazenam os dados do modelo PAEG.

Tabela 1 - Parâmetros representados explicitamente na base de dados do PAEG

Parâmetro	Descrição
<i>Fluxos</i>	
$vdgm_{ir}$	Demanda do governo (doméstica) pela commodity i na região r
$vigm_{ir}$	Demanda do governo (internacional)
$vdpm_{ir}$	Demanda dos agentes privados (doméstica)
$vipm_{ir}$	Demanda dos agentes privados (internacional)
$vdim_{ir}$	Demanda por investimentos
vfm_{fir}	Demanda das firmas por fatores primários
$vdfm_{ijr}$	Demanda das firmas por insumos intermediários domésticos
$vifm$	Demanda das firmas por insumos intermediários importados
vxd_{irs}	Fluxo comercial bilateral da região r para a região s
vst_{ir}	Exportações de serviços de transportes
$vtwr_{irs}$	Margens de transporte internacional
<i>Impostos e Subsídios</i>	
rto_{ir}	Subsídios domésticos à produção (%)
rtf_{fir}	Impostos ao uso de fatores primários na produção (%)
$rtpd_{ir}$	Imposto ao consumo doméstico dos agentes privados (%)
$rtpi_{ir}$	Imposto ao consumo importado dos agentes privados (%)
$rtgd_{ir}$	Imposto ao consumo doméstico do governo (%)
$rtgi_{ir}$	Imposto ao consumo importado do governo (%)
$rtfd_{ijr}$	Imposto ao consumo intermediário doméstico (%)
$rtfi_{ijr}$	Imposto ao consumo intermediário importado (%)
$rtxs_{irs}$	Subsídios às exportações (%)
$rtms_{irs}$	Tarifas de importação (%)
<i>Elasticidades</i>	
$esubd_i$	Elasticidade de substituição entre domésticos e importados
$esubva_i$	Elasticidade de substituição entre fatores primários
$esubm_i$	Elasticidade de substituição entre importações de diferentes origens
$etrae_f$	Elasticidade de transformação
eta_{ir}	Elasticidade renda da demanda
ϵ_{pi}_{ir}	Elasticidade preço da oferta

5.1 Código do PAEG em MPSGE

Para entender o código do PAEG contido no arquivo *paeg_model.gms*, é preciso entender os componentes e a lógica do modelo MPSGE. Os elementos de um programa MPSGE podem ser descritos da seguinte forma:

a) *Declaração de parâmetros e valores da base de dados*

No caso do PAEG, esses parâmetros foram incluídos ou criados através dos arquivos *read_data_model.gms* e *paeg.dat*, descritos anteriormente;

b) *Comando \$ontext seguido do comando \$model:paeg*

Informa o compilador do GAMS que um código em MPSGE será iniciado, sendo que o compilador do MPSGE irá reconhecer o nome do modelo como *paeg*. O nome do modelo deve ser escolhido pelo pesquisador e deve ser um nome válido para um arquivo, uma vez que um arquivo *.GEN*, no nosso caso, *PAEG.GEN*, será gerado pelo programa durante a sua execução;

c) *Declaração das variáveis do modelo.*

A declaração do modelo é feita através das variáveis que irão compor o próprio. Tais variáveis são apresentadas e classificadas pelas seguintes palavras-chaves:

\$SECTORS - Atividades de produção que convertem insumos em produtos. A variável associada a um setor é o seu nível de atividade.

\$COMMODITIES - Indica bens (intermediários ou finais) ou fatores. A variável associada a uma commodity é o seu preço.

\$CONSUMERS - Indivíduos que suprem os fatores recebem receitas dos impostos e pagam subsídios. A variável associada a um consumidor é a renda que este recebe de todas as fontes.

\$AUXILIARY - Variáveis adicionais, como fórmulas de markup ou de impostos com seus valores definidos endogenamente. Essas variáveis são funções de outras variáveis endógenas, como preços e quantidades.

Note que comentários, que são ignorados pelo MPSGE durante a execução do programa, podem ser adicionados na frente da declaração das variáveis após o sinal “!”.

No modelo PAEG, o código declarando os setores apresenta-se como:

```
$sectors:
    c(r)                ! Consumption
    g(r)                ! Government demand
    y(i,r)$vom(i,r)    ! Supply
```

$m(i,r)\$vim(i,r)$! Imports
$yt(j)\$vtw(j)$! Transportation services
$ft(f,r)\$(sf(f) \text{ and } evom(f,r))$! Specific factor transformation
$ftr(f,bra)\$mobf$! Factor transformation for Brazilian regions

Os conjuntos entre parênteses indicam que o setor existe para cada elemento deste conjunto. Por exemplo, o setor de consumo, $c(r)$, será criado para cada região r do modelo. A notação “\$” após o nome do setor indica uma condição para a existência daquele setor. Por exemplo, na declaração do setor de oferta, $y(i,r)$, o símbolo \$ seguido de $vom(i,r)$ indica que o setor y será criado para cada região r e *commodity* i se existir um parâmetro $vom(i,r)$ para tal região e setor na base de dados inicial. O parâmetro $vom(i,r)$ é criado a partir do arquivo *read_data_model.gms* que representa a oferta total de um bem em uma região na base de dados inicial. Dessa forma, se um país qualquer não produzir determinada *commodity* no equilíbrio inicial da base de dados, não existirá uma variável no modelo para a oferta daquele setor.

A declaração das *commodities* no modelo apresenta-se como:

\$commodities:	
$pc(r)$! Private consumption price index
$pg(r)$! Public consumption price index
$py(j,r)\$vom(j,r)$! Domestic output price
$pm(j,r)\$vim(j,r)$! Import price
$pt(j)\$vtw(j)$! Transportation services
$pf(f,r)\$vom(f,r)$! Primary factors rent
$ps(f,j,r)\$(sf(f) \text{ and } vfm(f,j,r))$! Sector-specific primary factors
$pfbra(f)\$mobf$! Primary factors price in Brazil

Novamente, a notação “\$” indica uma condição para a existência de determinada *commodity*.

O modelo PAEG possui dois agentes consumidores (households e government), declarados pelo código:

\$consumers:	
$hh(r)\$(not \text{ bra}(r))$! Representative household
$hhbr(bra)$! Representative household in Brazil
$govt(r)$! Representative government

Após a declaração dos consumidores é comum encontrar a declaração de variáveis auxiliares. O modelo PAEG não faz uso das mesmas, não sendo necessário utilizar tal declaração no modelo.

Declaradas as variáveis do modelo, o código passa para a descrição/construção dos blocos de produção e consumo.

d) *Construção dos blocos de produção*

Tais blocos definem as árvores tecnológicas que combinam os insumos, fatores intermediários, bens importados e bens de consumo. Ao ler tais blocos, o modelo MPSGE constrói as equações do modelo de equilíbrio geral durante a execução do mesmo no GAMS. A especificação de um bloco de produção em MPSGE se dá através da seguinte notação:

```
$PROD:X      s:(elasticidade de substituição)
O:PX         Q:(produção de X)   P:(preço de X)
I:PL        Q:(insumo L)       P:(preço de L)
I:PK        Q:(insumo K)       P:(preço de K)
```

Na primeira linha são especificados o nome da atividade (X), o valor das elasticidades de substituição (campo s:) e de transformação (campo t: , não mostrado no exemplo acima por se tratar de uma atividade que produz um único produto). A não especificação das elasticidades (campos s: e t:) faz com que o MPSGE atribua o valor padrão de zero a essas, ou seja, construa funções do tipo Leontief.

A partir da segunda linha, na primeira coluna são apresentados os nomes das commodities que representam os produtos (campo O:) e os insumos (campo I:). Na segunda coluna são apresentados os valores das commodities (campos Q:). Esses valores são as quantidades de referência dessas commodities no equilíbrio inicial, que são utilizados para fins de calibração do modelo. A não declaração desses campos é interpretada pelo MPSGE como o valor padrão de 1 para as quantidades. Na terceira coluna são declarados os preços de referência das commodities (campo P:), também usados para calibração do modelo. O valor padrão, na falta de declaração destes, também é considerado como igual a 1.

Um exemplo específico de um bloco de produção seria:

```
$PROD:X      s:1          (o nome da atividade é X, a função é Cobb-Douglas)
O:PX Q:100 P:1          (PX é a commodity produzida pelo setor X)
I:PL Q:40 P:1          (PL é a commodity trabalho)
I:PK Q:60 P:1          (PK é a commodity capital)
```

O MPSGE constrói a função de custo subjacente ao bloco de produção a partir das quantidades e preços de referência. No resultado do modelo, os valores associados às *commodities* são seus preços.

As quantidades e preços de referência (preenchidas nos campos Q: e P:) são usadas apenas na construção da função de custo correspondente. Os seus valores não são repassados ao *solver* como valores iniciais das variáveis. Todas as funções CES de retornos constantes são completamente especificadas a partir de um ponto de referência dado pelas quantidades e preços de insumos e produto e pela elasticidade de substituição (essas informações servem para determinar a inclinação da isoquanta). Dessa forma, as informações declaradas no bloco de produção especificam completamente a função de produção (ou de custo unitário).

No caso do bloco exemplificado acima, este descreve uma função de produção Cobb-Douglas. O campo *s:1* informa que a elasticidade de substituição entre insumos é igual à unidade. Os insumos são PL e PK (representados pelos campos I:PL e I:PK). O produto é PX (O:PX). Esse setor de produção converte 40 unidades de PL e 60 unidades de PK em 100 unidades de PX. Essa função de produção poderia ser matematicamente representada como $X = \phi L^{0,4} K^{0,6}$. A Figura 2 ilustra a calibração dessa função de produção para os preços e quantidades de *benchmark*.

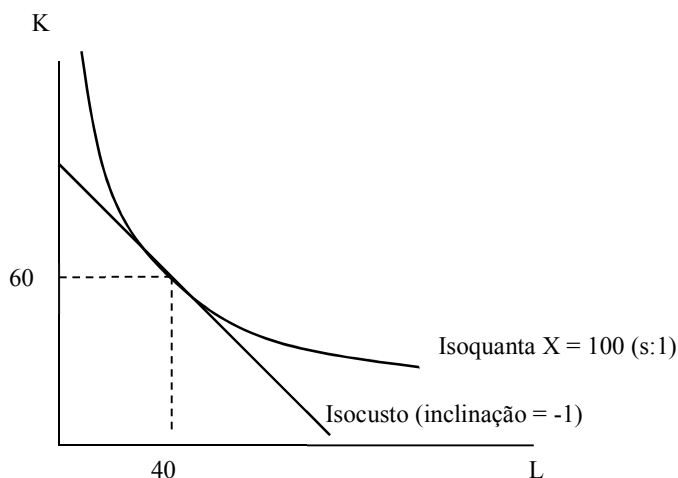


Figura 2 – Calibração da função de produção

O MPSGE especifica a função de produção através de um simples ponto de referência. Nesse exemplo especificam-se explicitamente apenas quantidades de referência, já que os preços de referência são considerados como unitários. Preços relativos dos insumos determinam a inclinação da função de isocusto, que por sua vez, se igualam à inclinação da isoquanta no ponto das quantidades de referência.

A curvatura da isoquanta é determinada pelo *s*, definido como a elasticidade de substituição entre os insumos. No exemplo, *s:1* significa que a função de produção é uma Cobb-Douglas. O valor padrão para a elasticidade *s*, na ausência da declaração deste

parâmetro, é considerado como zero. Quantidades e preços de referência no equilíbrio inicial são utilizados apenas para calibração. Eles não são utilizados pelo programa como valor inicial das variáveis (os níveis iniciais são considerados como unitários).

Agora que já se expôs como o modelo é declarado, serão observados os blocos de produção do PAEG.

O primeiro bloco de produção no modelo é o bloco de oferta:

```
$prod:y(j,r)$vom(j,r) s:0 i.tl:esubd(i) va:esubva(j)
o:py(j,r) q:vom(j,r) a:govt(r) t:rto(j,r)
i:py(i,r) q:vdfm(i,j,r) p:(1+rtfd0(i,j,r)) i.tl: a:govt(r) t:rtfd(i,j,r)
i:pm(i,r) q:vifm(i,j,r) p:(1+rtfi0(i,j,r)) i.tl: a:govt(r) t:rtfi(i,j,r)
i:ps(sf,j,r) q:vfm(sf,j,r) p:(1+rtf0(sf,j,r)) va: a:govt(r) t:rtf(sf,j,r)
i:pf(mf,r) q:vfm(mf,j,r) p:(1+rtf0(mf,j,r)) va: a:govt(r) t:rtf(mf,j,r)
```

Este bloco indica que o setor y produz a *commodity* $py(j,r)$ a partir da combinação de insumos intermediários, que são compostos de *commodities* produzidas domesticamente ($py(i,r)$) e *commodities* importadas ($pm(i,r)$), e dos fatores primários de produção ($ps(sf,j,r)$ e $pf(mf,r)$)⁷. Como existe a possibilidade de cobrança de impostos no consumo intermediário e no uso de fatores, os preços iniciais podem diferir da unidade para efeitos de calibração do modelo, o que explica a existência dos campos p : para cada insumo. Dessa forma, os campos p : indicam o nível do preço de referência no equilíbrio inicial para garantir a calibração do modelo, quando a presença do imposto torna o preço de referência diferente de um. Os insumos intermediários domésticos e importados são combinados a partir da elasticidade de substituição $esubd(i)$ para cada *commodity* i , o que é determinado pela declaração de $i.tl$ na primeira linha do bloco de produção e nas linhas dos insumos py e pm . Os fatores primários de produção são combinados a partir da elasticidade de substituição $esubdva(j)$, que é específica do setor j , e é representada pela declaração va :. No nível mais alto da árvore tecnológica, o conjunto agregado dos fatores de produção é combinado na forma de uma função Leontief (elasticidade de substituição igual a 0, definida pelo campo $S:0$) com os insumos intermediários. Os campos a : e t : especificam a cobrança de impostos sobre o uso dos fatores e insumos. O campo a : determina quem recebe a receita do imposto (nesse caso, o governo de cada região, ou agente $govt$), e o campo t : determina o nível do imposto.

A árvore tecnológica construída a partir do bloco de produção y pode ser representada como na Figura 3.

⁷ Neste caso, sf é o conjunto de fatores *sluggish* (sem mobilidade ou mobilidade restrita) e mf o conjunto de fatores com mobilidade.

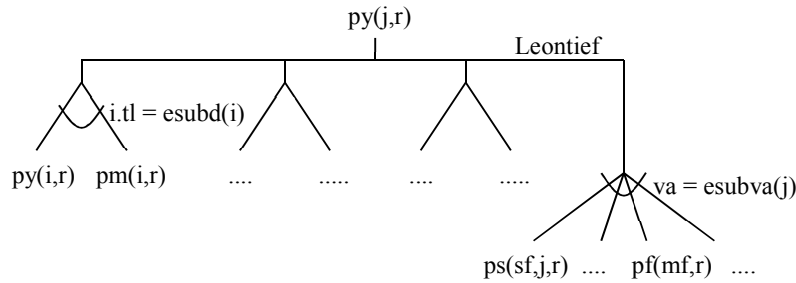


Figura 3 – Representação da árvore tecnológica do bloco de produção y

Os demais blocos de produção seguem a mesma lógica do bloco de produção y. Faz-se uma breve exposição desses blocos a partir daqui.

O bloco de produção yt determina a produção dos serviços de transportes pt, como abaixo.

$$\begin{array}{l}
 \$prod:yt(j)\$vtw(j) \quad s:1 \\
 o:pt(j) \quad q:vtw(j) \\
 i:py(j,r) \quad q:vst(j,r)
 \end{array}$$

Este bloco combina os insumos $py(j,r)$ para produzir $pt(j)$.

Exercício 1: Desenhe a árvore tecnológica que o MPSGE constrói a partir do bloco de produção de yt.

O próximo bloco de produção, bloco c, representa a combinação de bens e serviços para consumo. Este bloco combina *commodities* domésticas (py) e importadas (pm) para formar um índice agregado de consumo (pc). O bloco é declarado como:

$$\begin{array}{l}
 \$prod:c(r) \quad s:1 \quad i.tl:esubd(i) \\
 o:pc(r) \quad q:vpm(r) \\
 i:py(i,r) \quad q:vdpm(i,r) \quad i.tl: \quad p:(1+rtpd0(i,r)) \quad a:govt(r) \quad t:rtpd(i,r) \\
 i:pm(i,r) \quad q:vipm(i,r) \quad i.tl: \quad p:(1+rtpi0(i,r)) \quad a:govt(r) \quad t:rtpi(i,r)
 \end{array}$$

Exercício 2: Desenhe a árvore tecnológica que o MPSGE constrói a partir do bloco de produção acima.

A seguir, o código do PAEG representa o bloco de produção do consumo agregado do governo, bloco g, que como o bloco anterior, combina *commodities* domésticas (py) e importadas (pm) para formar um índice agregado de consumo do governo (pg):

```
$prod:g(r) s:0 i.tl:esubd(i)
o:pg(r) q:vgm(r)
i:py(i,r) q:vdgm(i,r) i.tl: p:(1+rtgd0(i,r)) a:govt(r) t:rtgd(i,r)
i:pm(i,r) q:vigm(i,r) i.tl: p:(1+rtgi0(i,r)) a:govt(r) t:rtgi(i,r)
```

Exercício 3: Desenhe a árvore tecnológica que o MPSGE constrói a partir do bloco de produção g(r) acima.

O próximo bloco de produção define o setor m, que agrega as importações bilaterais com origem em diferentes regiões e os gastos com transportes no comércio:

```
$prod:m(i,r)$vim(i,r) s:esubm(i) s.tl:0
o:pm(i,r) q:vim(i,r)
i:py(i,s) q:vxmd(i,s,r) p:pvxmd(i,s,r) s.tl:
+ a:govt(s) t:(-rtxs(i,s,r))
+ a:govt(r) t:(rtms(i,s,r)*(1-rtxs(i,s,r)))
i:pt(j)#(s) q:vtwr(j,i,s,r) p:pvtwr(i,s,r) s.tl:
+ a:govt(r) t:rtms(i,s,r)
```


O sinal “+” no meio do bloco apenas indica a continuidade da linha imediatamente acima, seja por falta de espaço para colocar tudo na mesma linha, seja pela facilidade de visualização de todas as informações do bloco de produção. Esse bloco mostra que o agregado de importações do bem i pela região r ($pm(i,r)$) é formado pelas *commodities* produzidas nas diversas regiões s ($py(i,s)$), considerando os subsídios e impostos às exportações ($rtxs(i,s,r)$) nas regiões produtoras s com destino à região importadora r e as tarifas às importações ($rtms(i,s,r)$) cobradas pela região importadora r dos bens com origem na região s .

Exercício 4: Desenhe a árvore tecnológica que o MPSGE constrói a partir do bloco de produção $m(i,r)$ acima.

O próximo bloco de produção é o bloco ft , responsável por determinar os fatores primários de produção com mobilidade restrita. Esse bloco determina que os fatores terra e recursos naturais possuam limitações na sua mobilidade. Como esses fatores não são representados de forma desagregada no modelo PAEG, uma vez que o PAEG apenas representa capital e trabalho, este bloco não será construído durante o processo de execução do modelo. Preservou-se o código deste bloco como no modelo GTAP uma vez que futuros avanços na construção da base de dados devem ser alcançados, permitindo separar os fatores terra e recursos naturais para as regiões brasileiras.

e) *Construção dos blocos de demanda*

Após os blocos de produção, o modelo PAEG declara os blocos de demanda, que definem as fontes de renda e os gastos dos agentes consumidores do modelo. Cada bloco de demanda especifica a *commodity* sendo consumida, a partir do campo d :, e as dotações de fatores do agente, a partir do campo e :. Os blocos de demanda são importantes para definir o fechamento macroeconômico do modelo. As variáveis colocadas nos campos e : serão consideradas exógenas, podendo sofrer choques nos exercícios de simulação.

O primeiro bloco de demanda é o do agente privado representativo, hh, como abaixo:

```
$demand:hh(r)$ (not bra(r))
  d:pc(r)          q:vpm(r)
  e:py(i,r)        q:(-vdim(i,r))
  e:pf(f,r)        q:evom(f,r)
  e:pc(r)          q:(-vtax(r))
```

Este bloco mostra que o agente privado representativo consome a *commodity* denotada pelo preço pc e recebe como renda a dotação de fatores (pf). Ainda, os agentes representativos são responsáveis pelos gastos com investimentos privados, determinados pelo campo e:py(i,r) seguido do campo q:(-vdim(i,r)). Os agentes representativos ainda recebem (ou pagam) uma transferência direta do governo, dado pelo campo e:pc(r) seguido de q:(-vtax(r)). Se o valor de vtax for negativo, os agentes privados recebem uma transferência, se for positivo, transferem para o governo. A soma dos campos e: deve ser igual ao campo d:, ou seja, a demanda privada agregada (vpm) deve igualar-se à renda dos fatores primários (evom) menos a demanda de investimentos (vdim) mais as transferências do governo (vtax). Reorganizando esses parâmetros, o fechamento determinado pelo bloco de demanda dos agentes privados simplesmente determina que toda a renda recebida pelos agentes privados, na forma de renda dos fatores produtivos e transferências do governo, deve ser completamente esgotada pelo consumo privado de bens e serviços e demanda de investimentos (ou poupança privada). Note que os campos “e:” determinam que a poupança privada, a dotação de fatores e as transferências são exógenas, ou seja, não reagem a mudanças em preços no modelo, uma vez que estão definidas por campos “e:”.

Vale notar que o bloco de demanda hh possui uma condição expressa pelo símbolo \$(not bra(r)) que determina que este bloco não seja construído para as regiões brasileiras, que compõe o subconjunto bra(r). O bloco de demanda para os agentes representativos de cada região brasileira é representado mais abaixo, como hhbr. Essa distinção entre hh e hhbr permite incluir o comportamento de movimento de fatores entre as regiões brasileiras. No bloco hhbr a dotação de fatores (evom) aparece associada a duas variáveis diferentes de preços, pf e pfbra. Quando a livre mobilidade de fatores entre regiões é permitida, pfbra será o preço determinado pelo modelo e pf será ignorado, pois não estará ativo no modelo. O preço pfbra é específico a cada região, sendo determinado pelo bloco de produção ftr. Esse bloco é responsável pela alocação de fatores entre regiões diferentes em resposta a mudanças na economia:

```
$prod: ftr(f)$mobf t:1 s:0  
o: pf(f, bra) q: evom(f, bra)  
i: pfbra(f, bra) q: evom(f, bra)
```

Nesse bloco, as dotações de um tipo de fator (f) advindas de todas as regiões estão disponíveis como insumos no campo i:pfbra(f) seguido de q:evom(f, bra), para serem transformadas em fatores regionais que serão utilizados especificamente em cada uma das regiões, no campo o:pf(f, bra) seguido de q:evom(f, bra). O símbolo t:1 na linha de definição do bloco (iniciada por \$prod:) determina a elasticidade de transformação entre os fatores das diferentes regiões, enquanto o símbolo s:0 determina a elasticidade de substituição igual a zero (Leontief) entre os diferentes insumos (fatores). Essa simbologia de elasticidades significa que os fatores das diferentes regiões são sempre combinados em proporções fixas, de acordo com a dotação regional inicial, e então distribuídos para as diversas regiões considerando uma função Cobb-Douglas de transformação entre as regiões. Isso significa que há mobilidade imperfeita de fatores entre as regiões, ou seja, não é possível o livre movimento de capital ou trabalho de uma região para outra diante de diferenças nas remunerações dos fatores, uma vez que as características e composição dos fatores de cada região não são exatamente iguais. Contudo, esse bloco permite representar algum grau de mobilidade de fatores entre regiões, na medida em que uma mudança na remuneração relativa de uma região em relação às demais tende a atrair trabalho e capital das demais regiões do país. Uma formulação alternativa seria considerar a livre mobilidade de fatores entre regiões, de forma que, após um choque, qualquer diferença na remuneração de um fator entre regiões fosse completamente eliminada pela migração de fatores, o que significaria que haveria apenas um único preço (salário ou retorno ao capital) em todas as regiões do país. Dadas as diferenças regionais em fatores bem como a persistência empírica de diferenças nas remunerações dos fatores entre regiões, mesmo no longo prazo, optou-se por considerar a hipótese de mobilidade imperfeita dos fatores.

Exercício 5: Desenhe a árvore tecnológica que o MPSGE constrói a partir do bloco de alocação de fatores $f(r)$.

O próximo bloco de demanda é o do agente governo, *govt*, como especificado abaixo:

```
$demand:govt(r)
  d:pg(r)
  e:pc(r)          q:vtax(r)
  e:pc(rnum)       q:vb(r)
```

Este bloco mostra que o governo recebe como renda as transferências internacionais calculadas na forma de saldo da conta corrente do balanço de pagamentos (parâmetro *vb*), determinada pelo campo *e:pc(rnum)* seguido de *q:vb(r)*, bem como recebe (paga) transferência do (para) agente privado representativo se *vtax(r)* for positivo (negativo). Isso é determinado pelo campo *e:pc(r)* seguido de *q:vtax(r)*. Ainda, não explícito no bloco de demanda, porém declarado ao longo dos blocos de produção, o agente *govt* recebe toda a renda dos impostos cobrados na economia, como explicado anteriormente a respeito dos campos *a:govt* e *t*: presentes nos blocos de produção. A renda do governo é utilizada para consumir a *commodity* *pg(r)*⁸. Os campos declarados no bloco do governo indicam que a renda total do governo, dada pelas transferências internacionais, pelas transferências das famílias e receita dos impostos, deve ser igual aos gastos do governo. Transferências internacionais e das famílias são consideradas exógenas, uma vez que foram declaradas a partir do campo *e*:

Note que o fechamento macroeconômico do modelo, a partir da combinação dos blocos de demanda privada e do governo, determina que: gastos dos agentes privados (*vpm*) + gastos do governo (*vgm*) + investimentos (*vdim*) = renda dos fatores (*evom*) + transferências internacionais (*vb*) + impostos. Como as transferências internacionais são dadas pelo negativo do saldo da conta corrente (importações – exportações), o fechamento macroeconômico

⁸ Note que não foi declarado o campo *q*: associado à *commodity* *pg*. Tal simplificação ocorre uma vez que só existe apenas mais um campo no modelo em que a *commodity* *pg* aparece, no bloco de produção *g*. Isso significa que poderíamos associar ao campo *d:pg(r)* o campo de quantidade *q:vgm(r)*.

apenas mostra a identidade de que a demanda agregada da economia (consumo+investimento+saldo líquido do governo+exportações-importações) deve igualar-se à renda (retorno dos fatores).

O fechamento macroeconômico é muito importante para explicar os efeitos de um choque qualquer nos modelos de equilíbrio geral. Alguns detalhes são importantes no fechamento do modelo PAEG. Na forma como os blocos de demanda estão escritos, o fechamento macroeconômico determina que: a dotação de fatores de produção é fixa, perfeitamente móvel entre setores e sem mobilidade entre regiões (a não ser dentro do Brasil quando se permite tal mobilidade); não há desemprego dos fatores (pressuposição de pleno emprego), o que significa que os preços dos fatores são flexíveis; a oferta de bens de capital é fixa, uma vez que investimentos não respondem a mudanças em taxas de retorno do capital e a dotação de capital é constante; transferências entre o governo e as famílias não respondem a mudanças em preços; os gastos do governo se ajustam a mudanças em preços dos bens e arrecadação de impostos, que por sua vez respondem a mudanças no nível de atividade e consumo; o saldo em conta corrente não é afetado por choques, ou seja, a taxa de câmbio real entre os países se ajusta para acomodar mudanças em exportações e importações.

f) *Construção de variáveis para reportar valores.*

Após os blocos de demanda, o modelo PAEG apresenta um bloco de definição de variáveis criadas com fins de armazenar valores, o bloco report:

\$report:

v:vxmd_(i,s,r)\$vxmd(i,s,r)	i:py(i,s)	prod:m(i,r)
v:vpm_(r)	o:pc(r)	prod:c(r)
v:vgm_(r)	o:pg(r)	prod:g(r)
v:vom_(i,r)	o:py(i,r)	prod:y(i,r)

Esse bloco cria variáveis cujos nomes são determinados no campo v:. Essas variáveis armazenam o valor da variável no bloco de produção determinado pelo campo prod: e pelo campo i: ou o: do bloco de produção. Por exemplo, a variável vxmd_(i,s,r) armazena os valores dos fluxos comerciais bilaterais do bloco de produção m(i,r), encontradas no campo da *commodity* py(i,s). Essas definições de variáveis são importantes para calcular mudanças nas mesmas após a implementação de choques no modelo.

g) *Construção das equações de restrições com as variáveis auxiliares*

Se fossem declaradas variáveis auxiliares (através da declaração: \$AUXILIARY), o código do modelo deveria apresentar as equações de restrições (*\$constraint*) que especificariam equações matemáticas para incorporação dessas restrições, como por exemplo, desemprego no mercado de trabalho.

No presente modelo não existe nenhuma equação de restrição especificada. Vale notar que o modelo é considerado, do ponto de vista da teoria macroeconômica, como um modelo de médio prazo, no conceito especificado por Blanchard (2006). Isso significa dizer que, diante de choques ou alterações na economia, esta se ajusta pela completa mobilidade dos fatores de produção até atingir um novo equilíbrio sem, contudo, alterar a dotação desses fatores, ou seja, sem acumulação de capital e/ou trabalho. Mudanças nas dotações de capital e/ou trabalho tornariam o modelo, na visão daquele autor, como sendo de longo prazo. Ainda, no médio prazo a economia retornaria para o seu nível de produto e emprego natural, o que significa dizer que a economia voltaria a produzir e empregar os fatores no limite máximo permitido pela sua estrutura e instituições. Nesse caso, todo desemprego e capacidade ociosa existentes seriam decorrentes de problemas na economia não relacionados diretamente à atividade econômica (desemprego natural).

Ainda, existe grande controvérsia e dificuldade na modelagem do desemprego em modelos de equilíbrio geral, uma vez que as teorias e entendimento do mercado de trabalho são diversas e geralmente não unânimes na explicação desse fenômeno econômico. Geralmente, as formas de modelagem matemática do desemprego são extremamente simples e sujeitas a muitas dúvidas na interpretação dos resultados do modelo.

Diante do conceito do modelo como de médio prazo e das dificuldades na especificação matemática do desemprego no modelo de equilíbrio geral, opta-se por não modelar o desemprego explicitamente, considerando que a taxa de desemprego observada na matriz de contabilidade social no equilíbrio inicial pode ser considerada como taxa de desemprego natural, que será a mesma no médio prazo, independente do choque de política implementado. Mais adiante neste documento faremos um exercício introduzindo o desemprego no modelo.

Mesmo com essa visão de pleno emprego, o modelo permite alguma inferência sobre possíveis mudanças na taxa de desemprego, uma vez que as remunerações dos fatores de produção são passíveis de mudanças. Se, por exemplo, após um choque o modelo indica um aumento na remuneração dos fatores mão de obra e capital isso significa uma pressão para a redução do desemprego e da capacidade ociosa na economia.

h) Comandos *\$offtext* e *\$sysinclude*

O comando *\$offtext* determina o final do código do modelo em MPSGE, pois retorna o controle do programa do MPSGE para o GAMS. Após o *\$offtext*, deve-se declarar o comando *\$sysinclude mpsgeset* seguido do nome do modelo, no nosso caso, PAEG. Note que o código do modelo em MPSGE fica contido entre os comandos *\$ontext* e *\$offtext*. O comando *\$offtext*, junto com o comando *\$sysinclude*, finalmente informam ao compilador do MPSGE que a declaração do programa terminou, e o compilador do GAMS volta a ler o programa a partir daí. No modelo PAEG, essas são as linhas que terminam o modelo em MPSGE:

```
$offtext  
$sysinclude mpsgeset paeg
```

i) Definição do numerário

Antes da declaração dos comandos que determinam a execução do modelo e logo após o final do código em MPSGE, podem ser escolhidos valores iniciais ou fixos para algumas variáveis, bem como a variável que servirá como numerário no modelo. Na ausência de uma definição explícita do numerário, como é o caso do modelo PAEG, o MPSGE define a renda da região mais rica como numerário. Isso significa que todas as mudanças em preços que forem observadas a partir dos choques são mudanças em relação à renda dessa região, que é mantida fixa. Esse numerário se torna desejável em um modelo multirregional do ponto de vista computacional e de interpretação dos resultados, ao invés das variáveis de preços comumente escolhidas como numerários nos modelos de única região. Isso por que, ao se fixar o preço de um bem ou fator específico de uma região qualquer, a interpretação das mudanças nas variáveis de outras regiões, principalmente preços, fica bem mais complexa e sujeita a erros, uma vez que não se sabe previamente qual poderia ser a mudança no preço escolhido como numerário caso este não o fosse e, portanto, estivesse passível de mudança. Assim, os resultados poderiam ser completamente diferentes se o preço escolhido como numerário pudesse variar. Já a renda da região mais rica é a variável do modelo que tende a sofrer as menores alterações, já que os efeitos de políticas comerciais globais são menores quanto maior for a economia sujeita a essas políticas. Portanto, tem-se como numerário uma variável pouco sujeita à mudanças. Com isso, as variações em preços podem ser interpretadas mais facilmente e com menor chance de erro.

j) Checagem do benchmark

A checagem do benchmark é realizada através de uma pré-simulação para verificar se a base de dados inicial do modelo está em equilíbrio. Após o comando \$offtext o programa segue com declarações de comandos que determinam a quantidade de memória disponível (em MB) para ser usada pelo GAMS para resolução do modelo (comando *.workspace*), o número máximo de iterações que serão realizadas pelo *solver* (comando *.iterlim*, precedido do nome do modelo) e a execução do GAMS e do MPSGE naquele ponto do programa (comando *solve <nome do modelo> using mcp*). Quando o número após o comando *iterlim* é definido como zero, o *solver* não irá resolver o modelo, mas sim, retornar os valores constantes na base de dados do modelo. Dessa forma, pode-se verificar se os dados da matriz de contabilidade social estão equilibrados. Definir o número de iterações como zero é importante para assegurar que os dados de *benchmark* contidos nas matrizes representam a solução inicial de equilíbrio. No PAEG esses comandos estão representados como a seguir:

```
paeg.workspace = 128;  
paeg.iterlim = 0;  
$include paeg.gen  
solve paeg using mcp;
```

Exercício 6: Checagem do equilíbrio inicial da base de dados – execute o modelo PAEG, clicando no botão com uma seta vermelha (Run GAMS ou F9). Assegure-se de que você está trabalhando em um projeto previamente aberto dentro da pasta PAEG. Após o término da execução, verifique no arquivo *paeg_model.lst* se a base de dados inicial está em equilíbrio. Para tal, digite ctrl+F para ativar a ferramenta de procura do GAMS, digite na mesma: “var c” e clique no botão *search*. A variável *c* é a primeira que aparece na listagem de resultados. Para que o modelo esteja em equilíbrio, o nível das variáveis deve ser igual a 1 e o valor marginal deve ser nulo ou próximo de zero (valores descritos por E-5 indicam cinco casas decimais à esquerda do número, o que pode ser considerado suficientemente próximo de zero). Corra a barra de rolagem para baixo e você verá as estatísticas descritivas para todas as variáveis do modelo. Note que apenas as variáveis HH e GOVT, que se referem à renda dos agentes representativos, e aquelas definidas no bloco \$REPORT, possuem níveis diferentes de 1. Essas variáveis são representadas pelos seus valores absolutos. Para que estejam em equilíbrio, o valor marginal das mesmas deve ser nulo ou próximo de zero.

k) Simulação de limpeza

A simulação de limpeza é uma pré-simulação para "limpar" o modelo de aproximações numéricas. Aqui, como ponto de partida, determina-se que o *solver* execute o modelo para um número elevado de iterações, simplesmente para maior clareza numérica e arredondamento das aproximações de casas decimais. Essa parte do modelo não precisa estar presente para que ele funcione normalmente. O código para tal simulação apresenta-se como:

```
paeg.iterlim = 8000;
#include paeg.gen
solve paeg using mcp;
```

l) *Armazenamento de valores iniciais e criação de parâmetros para reportar resultados*

Nessa parte são definidos diversos parâmetros que auxiliarão no cálculo dos resultados após a aplicação de choques ao modelo. Esses parâmetros são definidos com base nos valores iniciais dos dados das matrizes de insumo-produto e das variáveis definidas no modelo em MPSGE. Nessa parte, o código do programa é bastante autoexplicativo. Apenas como detalhe, quando se deseja armazenar o valor inicial de uma variável endógena na forma de um parâmetro, utiliza-se a sintaxe: <parâmetro> = <variável>.l, como no exemplo:

```
m0(i,r) = m.l(i,r);
```

Isso significa que o parâmetro $m0$, definido para cada região r e *commodity* i , armazenará o valor absoluto inicial (representado pela letra l , de *level*, após o ponto) da variável m .

m) *Implementação de cenários*

Essa parte é a mais importante do ponto de vista da utilização do modelo PAEG pelos usuários em geral. É aqui que serão definidos os choques de política a serem aplicados no modelo.

Os cenários são definidos a partir de alterações nas variáveis de política, como as tarifas às importações (tm), impostos às exportações (tx), impostos domésticos (rto , rtf , $rtfd$, $rtpd$, $rtgd$), ou ainda de variáveis exógenas de fechamento do modelo, como a dotação de fatores. O código do modelo já contém alguns cenários como sugestões para adoção de choques. O primeiro cenário colocado como exemplo simula a formação da ALCA, ou seja, a eliminação das tarifas de importação entre as regiões das Américas explicitamente representadas no modelo. O código de implantação desse cenário é apresentado como:

```

set      sc      Scenarios /alca/;

set      fta(sc,r)  Regions to form a free trade area
                    /alca.(nor,nde,coe,sde,sul,naf)/;

loop(sc,

* Apply the policies:

* Remove tariffs in some block or agreement:
rtms(i,s,r)$(fta(sc,r) and fta(sc,s)) = 0;
rtms(i,bra,bbra) = rtms0(i,bra,bbra);

```

O código mostra que, para implementar o cenário, primeiro foi criado um subconjunto $fta(sc,r)$ do conjunto de regiões r e de cenários sc , que define todas as regiões participantes desse acordo. Segundo, os elementos do conjunto r que farão parte da Alca foram definidos dentro do conjunto fta . Finalmente, determinou-se a eliminação das tarifas às importações em todos os países e regiões da Alca, através da linha:

```
rtms(i,s,r)$(fta(sc,r) and fta(sc,s)) = 0
```

Essa linha mostra que as tarifas às importações, dadas por $rtms$, serão eliminadas para todas as regiões r e s que estiverem contidas no subconjunto fta . Deve-se notar a linha logo abaixo desta, com a seguinte informação:

```
rtms(i,bra,bbra) = rtms0(i,bra,bbra);
```

Essa linha simplesmente mantém os impostos domésticos entre as regiões brasileiras (representadas pelos conjuntos bra e $bbra$), como o ICMS por exemplo, que não devem ser removidos diante de uma abertura comercial, mas que também são representados no PAEG pelo parâmetro $rtms$, já que tratam-se de impostos cobrados nas transações entre diferentes regiões.

No código do PAEG também se inclui a linha:

```
* rtxs(i,s,r)$(fta(sc,r) and fta(sc,s)) = 0;
```

Esta linha indica a remoção dos subsídios e impostos às exportações. Contudo, como esta linha possui um asterisco no seu início, ela não está ativa, o que significa que o GAMS irá ignorá-la durante a execução do modelo. Esta linha foi colocada para demonstrar como a remoção de subsídios e impostos às exportações poderia ser implementada no modelo.

Alguns outros possíveis choques, exemplificados no código do PAEG, consideram a redução de impostos domésticos, e estão descritos no código do modelo pelas seguintes linhas (mais adiante faremos alguns exercícios com esses choques):

```
*Remove ICMS inside the SDE region on inputs and goods from man sector
*rtfd("man",i,"sde") = rtfd0("man",i,"sde") - icms("man",i,"sde");
*rtpd("man","sde") = rtpd0("man","sde") - icms("man","cf","sde");

*      Remove ICMS in some transaction from SUL to SDE region:
*rtms("man","sul","sde") = rtms0("man","sul","sde") -
*                          icms_("sul","sde","man","man");
```

Após a descrição do cenário, o código instrui o GAMS a resolver o modelo mais uma vez, através do comando Solve:

```
$include paeg.gen
solve paeg using mcp;
```

n) Mensuração dos impactos do cenário implementado

A próxima parte do programa define diversos parâmetros e cálculos dos impactos do cenário em implementação. As fórmulas que definem os parâmetros são autoexplicativas. Como exemplo, a mudança percentual no nível de bem-estar de uma dada região é calculada pela fórmula:

```
ev(r,"ch_w_%") = round(100 * (C.L(r)-1),3);
```

Ev é o nome do parâmetro que armazena o resultado da mudança percentual no bem-estar.⁹ A variável c, que dá o nível de atividade do bloco de produção do consumo privado, é o índice de bem-estar do modelo. Seu valor após a implementação do choque é dado pela variável C.L (L significa *level*), enquanto seu nível antes do choque é considerado como

⁹ O parâmetro $ev(r, "ch_w_%")$ calcula a mudança em bem-estar para cada país na forma de variação equivalente em termos percentuais. De acordo com VARIAN (1992), tal medida procura indicar o aumento na utilidade dos consumidores domésticos, em termos de aumento da renda. A equação a seguir representa a fórmula da medida de variação equivalente:

$$VE = \frac{(U^f - U^0)}{U^0} RA^0$$

em que: VE representa a variação equivalente, U^f representa o nível de utilidade final, U^0 representa o nível de utilidade inicial e RA^0 representa a renda do agente privado no equilíbrio inicial. De acordo com VARIAN (1992), a medida de variação equivalente expressa a mudança na renda do consumidor necessária para que se mantenha o mesmo nível de utilidade, aos preços do equilíbrio inicial, quando o consumidor enfrenta um novo conjunto de preços. Tal medida indica aumentos de bem-estar para valores positivos, e redução de bem estar para valores negativos.

sendo igual a 1 por definição do MPSGE. A equação calcula a variação percentual, dada pela diferença entre o nível final e o nível inicial da variável.

Após a determinação dos diversos parâmetros de resultados, o restante do código do programa define quantas casas decimais se deseja apresentar para algumas variáveis (definido pelo comando *option*), determina que alguns resultados sejam apresentados no arquivo de extensão *lst* (comando *display*), e finalmente, exporta os resultados para um arquivo Excel tipo *xls*. Tal exportação é feita através da interface do GAMS com os arquivos *gdx*, que podem facilmente ser convertidos para planilhas do Excel (formato 97-2003). O comando *execute_unload* cria um arquivo *gdx* com o nome de “resultados” que armazena o parâmetro calculado, enquanto o comando *execute* transforma o arquivo *resultados.gdx* em *resultados.xls*, armazenando em cada planilha do arquivo um dos parâmetros de resultados calculados anteriormente. O arquivo *resultados.xls* será atualizada cada vez que o programa PAEG for executado, e deverá estar fechado para que o GAMS consiga atualizá-lo a cada execução do modelo. Caso queira preservar algum resultado, impedindo que este seja atualizado por novas simulações do modelo, deve-se alterar o nome do arquivo excel com esses resultados, diretamente no excel através do comando “salvar como”, ou através do windows explorer.

A seguir, serão implementados os choques da ALCA bem como outros choques ilustrativos.

6. Trabalhando com o Modelo PAEG

Para implementar o modelo PAEG pela primeira vez, é necessário seguir os passos descritos no item 4 deste documento – *Entendendo a Sequência de Execução do PAEG*. A partir da segunda simulação, basta executar o arquivo “*paeg_model.gms*” através do GAMS-IDE. Agora se demonstra como usar e modificar o código do modelo PAEG através de diversos exercícios.

Exercício 7a: Implementar o cenário ALCA no modelo PAEG e verificar os resultados.

O código do modelo PAEG já vem com um cenário pronto para adoção, o cenário ALCA. Para implementar o modelo basta abrir o arquivo *paeg_model.gms* no editor de texto GAMS-IDE e clicar no botão com uma seta vermelha (Run GAMS ou F9). Lembre-se de verificar se você está trabalhando em um projeto do Gams dentro da pasta PAEG, onde se encontra o arquivo *paeg_model.gms*.

Para verificar se o modelo foi executado corretamente, é necessário olhar as descrições sobre a solução do modelo no arquivo *paeg_model.lst*. A forma mais simples de fazer tal verificação é procurar, desde o início do arquivo *paeg_model.lst*, por 4 asteriscos seguidos. Para tal, pressione ctrl+F, digite “****” e click no botão *search*. Você deve encontrar o seguinte trecho do arquivo *paeg_model.lst*:

```

                S O L V E          S U M M A R Y

MODEL    PAEG
TYPE     MCP
SOLVER   PATH                      FROM LINE  3359

**** SOLVER STATUS      2 ITERATION INTERRUPT
**** MODEL STATUS      6 INTERMEDIATE INFEASIBLE

RESOURCE USAGE, LIMIT      0.078      1000.000
ITERATION COUNT, LIMIT     0          0
EVALUATION ERRORS          0          0

```

O programa GAMS cria um *Solve Summary* para cada comando *solve* presente no código do modelo. Se você retornar ao arquivo *paeg_model.gms* vai perceber que o comando *solve paeg using mcp* foi declarado três vezes. A primeira vez que esta declaração apareceu foi para checagem do *benchmark* (ver exercício 6). Para tal, determinamos que o número de iterações do solver seria zero, para reproduzir os dados do equilíbrio inicial. Isso explica a mensagem acima, em que o *solver status* indica 2 iterações interrompidas e o *model status* indica 6 soluções não factíveis ou inválidas.

Continue procurando pelos 4 asteriscos (para procurar a mesma expressão seguidas vezes basta pressionar F3 após realizar a primeira procura). O segundo *Solve Summary* que aparece diz respeito à simulação de limpeza para arredondar as aproximações numéricas. Na terceira vez que o *Solve Summary* aparecer, têm-se finalmente os resultados da simulação do cenário. Se o modelo tiver sido executado corretamente, o *solve summary* deve parecer como¹⁰:

```

                L O O P S                      sc alca

                S O L V E          S U M M A R Y

MODEL    PAEG
TYPE     MCP
SOLVER   PATH                      FROM LINE  4198

**** SOLVER STATUS      1 Normal Completion
**** MODEL STATUS      1 Optimal

RESOURCE USAGE, LIMIT      0.046      1000.000

```

¹⁰ Os valores do campo Resource Usage variarão com a capacidade do computador em uso.

ITERATION COUNT, LIMIT	3	8000
EVALUATION ERRORS	0	0

Este resultado mostra que o modelo encontrou um resultado ótimo, como esperado, e teve uma solução normal. Se você correr a barra de rolagem mais um pouco para baixo, visualizará os resultados da simulação do cenário para o nível das diversas variáveis. Veja, por exemplo, que a Var C para a região Norte (NOR) apresenta o nível de 1.001. Esses resultados no arquivo *lst* são geralmente de pouca praticidade para leitura, razão pela qual foram criados diversos parâmetros de resultados no código do modelo e instruído ao modelo para transferir tais parâmetros para o Excel.

Para visualizar os resultados no arquivo Excel, abra o arquivo *resultados.xls* a partir do Windows Explorer ou do próprio Excel. O arquivo *resultados.xls* deve ter sido criado dentro da pasta PAEG, a mesma onde está localizado o arquivo *paeg_model.gms*. O arquivo *resultados.xls* possui diversas planilhas com resultados da simulação. Essas planilhas são identificadas por símbolos relacionados aos seus conteúdos. A planilha *lwelfare*, por exemplo, apresenta os resultados de mudanças em bem-estar, PIB e gastos do governo, como na Tabela 2 a seguir.

Tabela 2 - Mudanças em bem-estar e PIB no cenário ALCA

	ch_w_%	ch_w_bi\$	ch_pib%	ch_Gov%
NOR	0.121	0.039	-0.004	-1.33
NDE	0.203	0.183	0.064	-0.711
COE	0.219	0.117	0.056	-0.539
SDE	0.256	1.045	0.067	-1.231
SUL	0.222	0.312	0.098	-2.322
NAF	0.061	6.971	0.008	-0.127
EUR	-0.002	-0.217	-0.006	0.001
ROW	-0.009	-0.977	0.004	-0.043

Para se ter certeza do que significa cada coluna, basta olhar no código do modelo (arquivo *paeg_model.gms*) como cada um dos parâmetros foi gerado e calculado. No caso dos resultados da planilha *lwelfare*, procure no arquivo *paeg_model.gms* o código:

```
execute_unload "resultados.gdx" ev
execute 'gdxrw.exe resultados.gdx o=resultados.xls par=ev rng=lwelfare!a1'
```

O código acima cria a planilha *Iwelfare* no arquivo de resultados. Ele ainda indica que o parâmetro armazenado nesta planilha é o parâmetro *ev*. Se você procurar pelo parâmetro *ev* no arquivo *paeg_model.gms*, você vai encontrar as seguintes linhas:

```
parameter          ev          Equivalent variation

ev(r,"ch_w_%") = round(100 * (C.L(r)-1),2);
ev(r,"ch_w_bi$") = vpm(r) * (C.L(r)-1);

ev(r,"ch_pib%") = chpib(r);
ev(r,"ch_Gov%") = gch(r);
```

Apesar de “rastrear” cada parâmetro de resultado gerado parecer complicado de início, a vantagem é que todos estão documentados nas linhas do modelo. Com o uso e prática do modelo os parâmetros vão se tornando usuais e o rastreamento deixa de ser necessário. Outra vantagem é que, em pouco tempo, o usuário poderá criar outros parâmetros de resultados que desejar visualizar, tomando como exemplo os resultados existentes. A Tabela 3 apresenta a lista dos parâmetros de resultados por hora apresentados no arquivo resultados.xls.

Tabela 3 – Parâmetros de resultados

Parâmetros	Descrição
1welfare	mudanças em bem-estar (variação equivalente % e em bi\$) e PIB (%)
2output	mudança no valor da produção setorial (%)
3br_exp	mudança no valor das exportações bilaterais com origem nas regiões brasileiras (%)
4br_imp	mudança no valor das importações bilaterais para as regiões brasileiras (%)
5tot_exp	mudança no valor das exportações totais por região (%)
6tot_imp	mudança no valor das importações totais por região (%)
7pc_ch	mudança no índice de preço do consumidor (%)
8py_ch	mudança nos preços das <i>commodities</i> domésticas (%)
9pf_ch	mudança na remuneração dos fatores primários (%)
10pm_ch	mudança nos preços das <i>commodities</i> importadas (%)
11chpib	decomposição das mudanças percentuais no PIB, ótica da demanda (%)
12pib	componentes do PIB antes e depois da simulação do cenário de política (US\$ bi)

Para finalizar este exercício, tente explicar os resultados de bem-estar e PIB encontrados na simulação da ALCA, e salve o arquivo Excel de resultados com um nome diferente para posterior comparação (por exemplo: resultados_alca.xls).

Exercício 7b: Implementar o modelo PAEG com o cenário ALCA, considerando a mobilidade de fatores entre regiões brasileiras, e verificar os resultados.

Procure logo no início do arquivo *paeg_model.gms* pelo parâmetro “mobf”. Esse parâmetro controla a hipótese de mobilidade parcial de fatores primários entre as regiões brasileiras. Mude o valor de mobf de zero para um para ativar a mobilidade de fatores entre regiões:

```
mobf = 1;
```

Agora execute novamente o modelo (clique no botão vermelho Run Gams ou F9) e tente entender os novos resultados gerados¹¹.

Exercício 8: Criando novos parâmetros de resultados

Os resultados da simulação foram apresentados em termos de variações percentuais. O usuário interessado em visualizar os resultados em valores, ou mesmo os dados iniciais, pode fazê-lo facilmente criando novas linhas de código no arquivo do modelo¹². Como exemplo, será criado um parâmetro para visualizar o valor da produção setorial antes e após a resolução do modelo. Para tal, procure o trecho do programa no arquivo *paeg_model.gms* que declara os parâmetros de resultados. Tal trecho inicia-se com o comentário:

```
* Define parameters to report:
```

Ao final da lista de parâmetros e antes do ponto e vírgula, acrescente a palavra “vom_rep”. Este será o parâmetro onde será armazenado o valor de produção. Inclua um comentário à frente da palavra vom_rep, como: “*value of output report*”.

Agora, acrescente as seguintes linhas após a fórmula que define o cálculo do PIB (parâmetro *chpib(r)*):

```
vom_rep(r,i,"Antes") = vom(i,r);
vom_rep(r,i,"Depois") = vom_l(i,r);
vom_rep(r,i,"%ch") = (vom_l(i,r)/vom(i,r) - 1)*100;
```

Para terminar, acrescente ao final do programa as linhas abaixo para que o GAMS transfira os resultados do parâmetro vom_rep para o excel:

```
execute_unload "resultados.gdx" vom_rep
execute 'gdxrw.exe resultados.gdx o=resultados.xls par=vom_rep rng=13vom_rep!a1'
```

¹¹ Fica a sugestão para sempre salvar o arquivo “resultados.xls” com nome diferente a cada exercício.

¹² Os dados iniciais dos parâmetros contidos na base de dados (fluxos, impostos, subsídios e elasticidades), como descritos na Tabela 1, estão armazenados no arquivo “paeg_3x8.xls”.

Antes de executar o programa novamente, mude o valor de *mobf* de um para zero ($mobf = 0$) e lembre-se de fechar o arquivo excel de resultados, para que este possa ser atualizado na próxima execução. Finalmente, execute novamente o programa clicando no botão vermelho (Run Gams ou F9). Agora, abra o arquivo excel de resultados e confira as variações em produção. Verifique se as variações percentuais contidas na planilha *13vom_rep* são iguais às armazenadas anteriormente na planilha *output* do arquivo de resultados.

Exercício 9: Alterando elasticidades no modelo

Este exercício introduz a ideia de análise de sensibilidade dos parâmetros importantes do modelo. Uma análise de sensibilidade envolve a simulação do mesmo cenário diversas vezes, alterando parâmetros e pressuposições que podem ser determinantes para os resultados daquele cenário. Vamos escolher alguns parâmetros quaisquer e simular o cenário Alca alterando o valor desses parâmetros.

Dois dos parâmetros mais importantes para determinar os resultados de um cenário de acordo comercial são as elasticidades de substituição entre bens importados e domésticos e elasticidades de substituição entre as importações bilaterais. No PAEG esses dois parâmetros são representados por *esubd(i)* e *esubm(i)*, respectivamente. Localize em que parte do modelo em MPSGE esses parâmetros aparecem. Agora, abra o arquivo *paeg_3x8.xls*, localizado na pasta PAEG, para visualizar os valores iniciais desses parâmetros no modelo. Esses valores são baseados no modelo GTAP. É possível alterar esses valores, caso o pesquisador encontre evidências de valores diferentes na literatura, ou mesmo caso queira realizar alguma análise de sensibilidade, como estamos demonstrando aqui.

Será realizado um experimento alterando os valores dessas elasticidades. Mas antes, para preservar os resultados do cenário ALCA simulado, o ideal é você alterar o nome do arquivo resultados a partir do programa windows explorer ou do próprio excel. Sugiro alterar o arquivo para *resultados_alca.xls*. Feche o arquivo para permitir futuras atualizações quando da simulação de novos cenários. Agora, acrescente as seguintes linhas ao modelo, imediatamente antes da declaração do modelo em MPSGE (ou seja, antes das linhas *\$ontext* e *\$model:paeg*):

```
esubd(i) = 2*esubd(i);  
esubm(i) = 2*esubm(i);
```

Essas linhas simplesmente determinam que *esubd* e *esubm* devem ser o dobro do valor inicialmente observado no arquivo *paeg_3x8.xls*. Agora, salve o modelo e execute-o. Olhe os

resultados de mudanças em bem-estar e compare-os aos obtidos anteriormente no cenário Alca. Você consegue explicar o que houve?

Exercício 10: Implementando cenários alternativos: ALCA com restrições

Agora será implementado um cenário ALCA com exceções na liberalização de produtos do agronegócio. Apague ou comente as linhas introduzidas no último exercício de alteração das elasticidades (para comentar basta adicionar asteriscos na frente das linhas). Feche o arquivo de resultados. Agora, localize no modelo o código que determina o cenário sendo simulado. Em particular procure pelo código:

```
rtms(i,s,r)$(fta(sc,r) and fta(sc,s)) = 0;
rtms(i,bra,bbra) = rtms0(i,bra,bbra);
```

Acrescente logo abaixo dessas linhas o seguinte código:

```
rtms("agr",s,r) = rtms0("agr",s,r);
```

O código acima determina que as tarifas às importações sobre as *commodities* do setor agricultura continuem nos seus níveis iniciais, *rtms0*, ou seja, não sejam reduzidas com a Alca. Salve o modelo, execute-o e confira os resultados. Compare os resultados encontrados com aqueles observados no cenário anterior, em que os produtos agropecuários não eram excluídos do acordo. Os resultados fazem sentido? Explique.

Exercício 11: Implementando cenários alternativos: outra formulação para a ALCA com restrições

Há várias formas de acrescentar linhas ao código para gerar um mesmo cenário. Vamos explorar essa flexibilidade agora. Salve o arquivo de resultados da Alca sem produtos agrícolas com algum nome diferente, por exemplo, *resultados_alca_noag.xls* e feche-o. Agora, coloque um asterisco na frente da linha criada anteriormente para excluir os produtos agrícolas do cenário. Isso fará com que o GAMS ignore aquela linha. Agora, acrescente a seguinte linha logo abaixo da definição do set *fta* e logo acima do código *loop(sc, :*

```
set      ftai(i) /man, ser/;
```

Essa linha cria um subconjunto de bens e serviços, que serão afetados pela formação da ALCA. Agora, substitua o conjunto *i* pelo subconjunto *ftai* na definição da eliminação da tarifa, como abaixo:

```
rtms(ftai,s,r)$(fta(sc,r) and fta(sc,s)) = 0;
```

Salve o arquivo e execute-o. Abra o arquivo de resultados e compare-o com os do arquivo *resultados_alca_noag.xls*.

Exercício 12a: Implementando cenários alternativos: Liberalização multilateral

É possível em um mesmo cenário alterar vários instrumentos de políticas ao mesmo tempo. Neste exemplo, vamos reduzir as distorções comerciais em tarifas, subsídios às exportações e subsídios à produção doméstica. Vamos definir, por exemplo, uma liberalização comercial mundial hipotética com os cortes em barreiras protecionistas definidos como na Tabela 4.

Tabela 4 - Reduções hipotéticas nas distorções comerciais e domésticas

Distorções	Países desenvolvidos	Países em desenvolvimento
Tarifas às importações na agricultura	30%	15%
Tarifas às importações outros setores	50%	30%
Subsídios às exportações agrícolas	100%	50%
Subsídios domésticos na agricultura	50%	30%

Para implementar esses cortes no modelo, feche o arquivo de resultados (salve com outro nome se quiser preservá-lo). Mude o nome do cenário de *alca* para *wto*:

```
set sc Scenarios /wto/;
```

Coloque asteriscos na frente das linhas definindo as regiões da ALCA (FTA). Agora, acrescente as seguintes definições de conjuntos para separar regiões e produtos e facilitar a implementação dos cortes acima:

```
set wto(r) All regions /nor,nde,coe,sde,sul,naf,eur,row/;
set rich(r) Developed Regions /naf,eur/;
set poor(r) Developing Regions /nor,nde,coe,sde,sul,row/;
set ag(i) /agr/;
set oth(i) /man, ser/;
```

Finalmente, coloque asteriscos na frente da remoção de tarifas da ALCA e acrescente logo em seguida o código:

```
* Reduce tariffs:
rtms(ag,s,r)$(rich(r) and wto(s)) = 0.7*rtms0(ag,s,r);
rtms(ag,s,r)$(poor(r) and wto(s)) = 0.85*rtms0(ag,s,r);
rtms(oth,s,r)$(rich(r) and wto(s)) = 0.5*rtms0(oth,s,r);
```

```

rtms(oth,s,r)$ (poor(r) and wto(s)) = 0.7*rtms0(oth,s,r);

*
Reduce export subsidies:
rtxs(ag,s,r)$ (wto(r) and rich(s) and rtxs0(ag,s,r) gt 0) = 0;
rtxs(ag,s,r)$ (wto(r) and poor(s) and rtxs0(ag,s,r) gt 0) =
0.5*rtxs0(ag,s,r);

*
Reduce domestic subsidies:
rto(ag,r)$ (rich(r) and rto(ag,r) gt 0) = 0.5*rto0(ag,r);
rto(ag,r)$ (poor(r) and rto(ag,r) gt 0) = 0.7*rto0(ag,r);

```

Note que, pelo código acima, subsídios às exportações e à produção doméstica só são reduzidos se estes são realmente subsídios ao invés de impostos (rtxs e rto na base de dados do GTAP, se positivos, são considerados subsídios, se negativos, são impostos), indicado pela condição entre parêntesis de que rtxs0 e rto0 sejam maiores (gt) que zero. Ainda, seria possível definir os mesmos cortes para cada país e cada setor individualmente, como por exemplo:

```
rtms("agr","sde","naf") = 0.7*rtms0("agr","sde","naf");
```

Esta instrução determina que as tarifas às importações cobradas pelo NAFTA sobre as exportações de bens agropecuários com origem na Região Sudeste sejam reduzidas em 30%. Pode-se imaginar que seria muito mais trabalhoso listar cada corte individualmente para atingir os objetivos da Tabela 4.

Não se esqueça de preservar a linha:

```
rtms(i,bra,bbra) = rtms0(i,bra,bbra);
```

para evitar que impostos nas transações entre regiões brasileiras sejam eliminados.

Salve o modelo e execute-o. Abra o arquivo de resultados no excel e verifique os impactos do cenário simulado. Este cenário é melhor ou pior do que a ALCA?

Exercício 12b: Liberalização multilateral com livre mobilidade dos fatores entre as regiões

Procure pelo parâmetro mobf no início do código do arquivo paeg_model.gms. Mude este parâmetro de zero para um:

```
mobf = 1;
```

Agora execute novamente o modelo e tente entender os novos resultados gerados. Compare-os com os resultados obtidos sem a mobilidade dos fatores.

Antes de prosseguir, mude o parâmetro mobf de volta para zero:

```
mobf = 0;
```

Exercício 13: Implementando cenários de redução de impostos domésticos

No trabalho de preparação da base de dados do PAEG foi feito um esforço de desagregação e representação de diferentes alíquotas de impostos incidentes em transações dentro do território nacional, como ICMS, IPI e outros. Dessa forma, é possível simular políticas fiscais e de mudança na estrutura tributária através do modelo PAEG. No presente exercício ilustra-se a remoção de impostos domésticos, como o ICMS.

Para tal, procure no modelo PAEG pelas linhas:

```
*          Remove ICMS inside the SDE region on inputs and goods from man
sector
* rtf("man",i,"sde") = rtf0("man",i,"sde") - icms("man",i,"sde");
* rtpd("man","sde") = rtpd0("man","sde") - icms("man","cf","sde");

*          Remove ICMS in some transaction from SUL to SDE region:
* rtms("man","sul","sde") = rtms0("man","sul","sde") -
*                               icms_ ("sul","sde","man","man");
```

Essas linhas simulam a retirada do ICMS que incide sobre as transações de consumo intermediário e consumo final de bens manufaturados na região Sudeste, bem como no ICMS cobrado de bens manufaturados com origem na região Sul e destinados à região Sudeste. Vale notar que a alíquota do ICMS na base de dados do PAEG é parte da alíquota total de impostos incidentes sobre o consumo intermediário (rtfd) e o consumo final dentro de uma região (rtpd), bem como parte da alíquota de impostos sobre as transações entre regiões diferentes (rtms).

Para implementar esses choques, remova os asteriscos na frente das linhas que se iniciam por rtf, rtpd e rtms. Não se esqueça de acrescentar asteriscos na frente das linhas que simulavam choques anteriores (WTO e ALCA), inclusive a linha do “rtms(i,bra,bbra) = rtms0(i,bra,bbra);”. Embora se recomende não alterar os comandos do loop, observar que ao comentar as linhas de wto e alca, deixa de ser necessária a linha do loop, podendo ser incluído asterisco, mas com cuidado para também colocar asterisco no fechamento do loop “);” pouco antes dos comandos de display.

Execute o modelo e analise os resultados. Como simulação alternativa, pode-se testar cada um dos choques separadamente, removendo cada asterisco de uma vez. Isso permitiria

verificar qual das várias reduções em impostos contribui mais para o resultado dos choques simulados todos ao mesmo tempo.

Exercício 14: Alterando o fechamento do modelo

O fechamento do modelo pode ser alterado de várias formas. A escolha do fechamento é importante na interpretação dos resultados e deve ser definida com base no problema de pesquisa. Será feito um teste simples de alteração do fechamento do modelo. Ao invés de considerar que os investimentos são determinados pela oferta de bens de capital, que é fixa, adota-se a hipótese de que os investimentos são responsivos a preços e realizados de acordo com uma propensão marginal a poupar constante. Para tal, basta alterar, no bloco de demanda do agente privado (blocos hh e hhbr), o código de:

```
e:py(i,r)      q:(-vdim(i,r))
```

para:

```
d:py(i,r)      q:(vdim(i,r))
```

e no bloco hhbr:

```
e:py(i,bra)      q:(-vdim(i,bra))
```

para

```
d:py(i,bra)      q:(vdim(i,bra))
```

Os elementos na frente do campo “e:” (de *endowment*) são exógenos, ou seja, os valores atribuídos aos mesmos nos campos “q:” não variam (a menos que sejam usados para simular algum choque), enquanto os elementos na frente dos campos “d:” (de *demand*) são endógenos, e portanto, reagem a quaisquer choques e mudanças em preços no modelo. Os blocos de demandas determinam a renda e o gasto agregado dos agentes. Dessa forma, valores positivos na frente dos campos “e:” indicam dotações de fatores e outros elementos que contribuem com a renda do agente, enquanto valores negativos nesses campos representam elementos exógenos que reduzem a renda dos agentes disponível para consumo. A alteração acima no bloco de demanda significa que, agora, a poupança na economia irá variar com a renda dos agentes representativos, e como poupança deve se igualar ao investimento, a oferta de bens de capital na economia muda com a renda das famílias. Em termos práticos, qualquer aumento de renda dos agentes privados proveniente de algum choque deve ser distribuído entre consumo e investimento, o que significa que o consumo deve variar menos do que sob o fechamento anterior, em que os investimentos eram fixos. Verifique se essa ideia está correta,

executando o modelo com a alteração no fechamento. Mas antes, feche o arquivo de resultados obtido no último exercício. O que aconteceu com os resultados de bem-estar com o novo fechamento?

Exercício 15: Incluindo variáveis auxiliares e restrições: acordo comercial com efeito neutro sobre a arrecadação do governo.

Antes de iniciar esse exercício, volte o fechamento do modelo para a forma inicialmente representada, ou seja, altere os blocos hh e hhbr de forma a exibirem:

```
e:py(i,r)          q:(-vdim(i,r))
```

Ainda, remova ou comente o código de simulação do cenário wto do exercício 11 e da redução de impostos do exercício 13, e ative novamente o cenário Alca.

Foi explicado anteriormente que é comum encontrar no modelo MPSGE variáveis auxiliares e equações de restrições. Essas variáveis e equações permitem acrescentar ao modelo aspectos e fenômenos econômicos adicionais ou alternativos, como fórmulas de markup ou de impostos com valores endógenos que são funções de outras variáveis, como preços e quantidades. Para exemplificar o uso da variável auxiliar e das equações de restrições, modifica-se o modelo para representar a possibilidade de implementação de redução de impostos sem alteração da arrecadação do governo. Para tal, serão permitidas transferências diretas do agente privado para o governo quando a política provocar menor arrecadação de impostos, até o ponto em que tais transferências reequilibrem os gastos do governo. Considerando o mesmo cenário implementado até agora, a Alca, é preciso acrescentar as seguintes linhas ao código do modelo:

a) após a declaração dos consumidores (\$consumers: hh(r) e govt(r)):

```
$auxiliary:
    tau(r)      ! Lumpsum replacement tax rate
```

Essa linha cria a variável auxiliar tau.

b) ao final do bloco de demanda hh(r):

```
e:pc(r)          q:(-vpm(r))          r:tau(r)
```

e ao final do bloco de demanda hhbr(r):

```
e:pc(bra)        q:(-vpm(bra))          r:tau(bra)
```

Essas linhas adicionam uma transferência negativa (com origem no agente representativo $hh(r)$) determinada pela variável auxiliar τ .

c) ao final do bloco de demanda $govt(r)$:

```
e:pc(r)           q:(vpm(r))           r:tau(r)
```

Essa linha adiciona uma transferência positiva (com destino ao agente representativo $g(r)$) determinada pela variável auxiliar τ .

d) após o bloco de demanda $g(r)$ e antes do bloco `report`:

```
$constraint:tau(r)
g(r) =e= 1;
```

Essas linhas definem a regra que ajustará endogenamente a variável τ .

Algumas observações sobre as linhas inseridas: a variável auxiliar τ multiplica $vmp(r)$ e $pc(r)$ até que a condição determinada pela equação de restrição seja atingida. Essa condição assegura que o nível de atividade do bloco de produção do consumo agregado do governo (g) não seja alterado em relação ao equilíbrio inicial. Isso significa que, se um choque qualquer diminuir a arrecadação do governo, o agente representativo transferirá renda ao agente governo de forma a garantir que o nível inicial de atividade deste não seja afetado.

e) falta acrescentar uma última linha antes de implementar o modelo, necessária para introduzir um valor ou limite inicial à variável τ . Acrescente:

```
tau.lo(r) = -inf;
```

imediatamente após as linhas:

```
$offtext
$sysinclude mpsgeset paeg
```

Nesse caso, estamos introduzindo um limite inferior (dado pelo sufixo `.lo`) para a variável. Agora, execute o modelo e verifique os resultados. Você conseguiria explicá-los?

Exercício 16: Incluindo variáveis auxiliares e restrições: desemprego

Antes de continuar, apague as linhas acrescentadas no último exercício, ou transforme-as em comentários e feche o arquivo de resultados. Agora será incluído desemprego nos mercados de trabalho no modelo. Isso é feito através de uma variável auxiliar para representar uma rigidez no nível do salário na economia. Devem-se seguir os seguintes passos:

a) primeiro, tem que definir o nível de desemprego em cada região. Considera-se hipoteticamente que as regiões brasileiras possuam 10% de desemprego e as demais regiões do mundo possuam 6%¹³. Cria-se um parâmetro que reflita esses números. Para tal, insira as seguintes linhas antes do início do código do modelo em MPSGE:

```
parameter u0;
u0(r) = 0.06;
u0(bra) = 0.1;
```

b) Agora, cria-se a variável auxiliar que ajusta o nível de emprego. Acrescente as linhas seguintes após a declaração dos consumidores (\$consumers hh(r) e govt(r)):

```
$auxiliary:
    unemp(r) $u0(r)           ! Unemployment
```

c) como se quer caracterizar o desemprego no mercado de trabalho, sem alterar o mercado de capital, é preciso desmembrar a dotação destes dois fatores. Substitua a linha da dotação de fatores no bloco de demanda do agente privado hh(r):

```
e:pf(f,r)           q:(evom(f,r))
```

por:

```
e:pf("cap",r)       q:(evom("cap",r))
e:pf("lab",r)       q:(evom("lab",r)/(1-u0(r)))
```

Dica: coloque um asterisco à frente da linha a ser substituída, ao invés de removê-la do programa. A fórmula na última linha acima amplia a oferta total de trabalhadores na economia de forma a incluir os desempregados.

Para o bloco de demanda do agente privado nas regiões brasileiras hhbr(bra), coloque um asterisco diante da linha:

```
e:pf(f,bra) $ (mobfb(bra))   q:evom(f,bra)
```

e acrescente as linhas:

¹³ A escolha aqui é aleatória. O correto seria procurar estatísticas de desemprego nas regiões no ano base dos dados e utilizar tais estatísticas para definir o nível de desemprego.

```
e:pf("cap",bra)$mobfb(bra)      q:evom("cap",bra)
e:pf("lab",bra)                  q:(evom("lab",bra)/(1-u0(bra)))
```

d) agora cria-se mais uma linha com a dotação de trabalho que representa o total de desempregados, a serem reduzidos da oferta total de trabalhadores da economia. Esse total de desempregados será ajustado quando choques ocorrerem no modelo, de acordo com os ajustamentos na variável auxiliar *unemp*. Acrescente a seguinte linha logo abaixo da linha da dotação de trabalho acrescentada por último no bloco *hh(r)*:

```
e:pf("lab",r)$u0(r)  q:(-evom("lab",r)/(1-u0(r)))  r:unemp(r)$u0(r)
```

E as seguintes linhas no bloco *hhbr(bra)*:

```
e:pf("lab",bra)$mobfb(bra) and u0(bra)  q:(-evom("lab",bra)/(1-u0(bra)))
+                                         r:unemp(bra)$u0(bra)
```

Perceba o uso do condicional \$, que determina que esta linha só seja considerada no código do modelo se existir o parâmetro *u0(r)*. Isso quer dizer que, para que o modelo deixe de considerar a formulação de desemprego e volte à formulação original de pleno emprego, basta definir *u0(r)* como igual a zero, sem a necessidade de apagar as linhas acrescentadas no código para representar o desemprego.

e) Define-se agora a equação de restrição que determina a rigidez salarial. Inclua as linhas seguintes logo abaixo do bloco *\$demand:govt(r)* e acima do bloco *\$report*:

```
$constraint:unemp(r)$u0(r)
             pf("lab",r) =g= pc(r);
```

Essa equação determina que a remuneração paga ao fator trabalho tenha que ser maior ou igual (=g=) ao índice de preços do consumidor. Lembrando que os preços no equilíbrio inicial são todos unitários, essa restrição apenas determina a rigidez salarial de que os trabalhadores não admitem quedas na sua renda real do equilíbrio inicial, mesmo que haja desemprego na economia. A variável *unemp* irá ajustar o nível de desemprego de forma a garantir que a condição de rigidez salarial seja respeitada. Por exemplo, se um choque qualquer aumenta o nível de atividade da economia, pressionando os salários em direção a um aumento, mais trabalhadores serão contratados na economia enquanto o nível do salário real for igual ou superior ao seu nível original. Isso levaria a uma diminuição do desemprego se o índice de preços ao consumidor variar menos ou na mesma proporção que o nível do salário nominal.

f) falta apenas acrescentar uma última linha antes de implementar o modelo, necessária para determinar um valor inicial à variável unemp. Acrescente:

```
unemp.l(r) = u0(r);
```

imediatamente após as linhas:

```
$offtext  
$sysinclude mpsgeset paeg
```

Agora, execute o modelo, observe os resultados e interprete-os.

7. Conclusões e sugestões de leitura

Neste documento procurou-se introduzir os princípios básicos do modelo PAEG através de exemplos que demonstram a sua mecânica e as ideias gerais da modelagem de equilíbrio geral no GAMS e no MPSGE. Os exercícios aqui explorados permitem ao usuário do PAEG começar a implementar seus cenários de interesse. Contudo, estão longe de abranger todas as possibilidades de uso do modelo e dos softwares GAMS e MPSGE. Portanto, nada mais natural nesta conclusão do que sugerir alguns dos inúmeros recursos disponíveis para avanços no conhecimento das técnicas de modelagem em equilíbrio geral.

O site: <http://www.mpsge.org/mainpage/mpsge.htm> apresenta a mais completa coleção de modelos, textos e apresentações sobre o MPSGE, desde a introdução básica sobre a técnica até exemplos avançados de modelagem de dinâmica intertemporal. Um dos links recomendado nesta página é: <http://www.gams.com/solvers/mpsge/index.htm>, que é a página oficial do GAMS sobre o MPSGE. Entre os links presentes na página do MPSGE, há um específico para o GTAP6inGAMS, que é a base do modelo PAEG: <http://www.mpsge.org/gtap6/>. Por fim, recomenda-se o contato com a página oficial do GTAP (<http://www.agecon.purdue.edu/gtap>), que, apesar de não ser escrito em GAMS, é a principal contribuição no desenvolvimento de bases de dados e modelos de equilíbrio geral multirregionais em escala mundial.

Referências Bibliográficas e Textos Sugeridos

BLANCHARD, O. **Macroeconomia**. 4ª Ed., Prentice Hall, 2006.

HERTEL, T. W. **Global trade analysis: modeling and applications**. New York: Cambridge University Press, 1997.

RUTHERFORD, T. F., **GTAP6inGAMS: The Dataset and Static Model**. Ann Arbor, MI, 2005. 42 p. (mimeo). (<http://www.mpsge.org/gtap6/gtap6gams.pdf>)

VARIAN, H.R. **Microeconomic analysis**. 3.ed. New York: Norton, 1992. 506 p.