

# PEP-1-t

## A USER GUIDE

**April 2012**

Véronique Robichaud <sup>1</sup>

André Lemelin <sup>2</sup>

Hélène Maisonnave <sup>3</sup>

Bernard Decaluwé <sup>4</sup>



*This work is a joint effort between IFPRI and PEP in the context of the AGRODEP project, and the authors acknowledge the financial support of the AGRODEP project.*

---

<sup>1</sup> CIRPÉE et PEP, Université Laval, Québec

<sup>2</sup> Centre INRS-UCS, Université du Québec, Montréal

<sup>3</sup> CIRPÉE et PEP, Université Laval, Québec

<sup>4</sup> Département d'économie, Université Laval, Québec

# PEP-1-t: A User Guide

---

## 1 Introduction

The PEP-1-t model follows the PEP-1-1 single-country, single-period model. It is a recursive dynamic model, the second research tool of the *Standard PEP Model* project that emerged spontaneously from the long-standing association between the co-authors. We propose a dynamic computable general equilibrium (CGE) model designed for the study of a national economy and intended to be an operational tool for PEP Network researchers and other users. With it, researchers will be able to develop a relatively standard model, and easily apply it to their country, whatever the particular structure of their social accounting matrix (SAM).

The objective of this paper is to present the different steps a user should follow to apply this particular model to a given country. It is intended for researchers who already have some background in general equilibrium and GAMS and who have in hand a balanced SAM.

The reader interested in the complete documentation of PEP-1-t, the underlying economic assumptions, functional forms, calibration procedures, mathematical structure and derivation should refer to *The PEP Standard Computable General Equilibrium Model Single-Country, Recursive Dynamic Version PEP-1-t* (Decaluwé, Lemelin, Robichaud and Maisonnave – hereafter DLRM, 2012). The present document does not replace DLRM, which, in our view, should be read by the user before this one.

## 2 General overview

Like the PEP 1-1 standard model, the PEP-1-t model distinguishes several categories of workers and capital. Also, PEP-1-t is capable of taking into account a broad set of tax instruments, and it models all possible transfers between institutions (agents). Moreover, the aggregate output of each industry consists of several products, consistent with rectangular input-output tables. Finally, the GAMS code has been written in a general form, thanks to the use of sets. This will facilitate the application of PEP-1-t to variously aggregated SAMs, by means of a few simple steps to make the SAM directly readable into the GAMS program.

The user should know beforehand that the PEP-1-t standard model is not designed to take the following situations into account.

- Industries that solely rely on value added (no intermediate consumption) or on intermediate consumption (no value added).
- Multiple trading partners.
- Re-exports.

The model is nonetheless readily modifiable to adapt it to any of the situations mentioned above and many others. But as a first step, we strongly suggest that the modeler not depart from the standard structure proposed in this document. Only when the standard version runs properly should the user add additional features.

Before starting to read this document, the user should have in hands the PEP-1-t GAMS code (PEP\_1-t\_V2\_0.gms), the model SAM it uses (SAM-V2\_0.xls), and the SAM the user wishes to build her/his model on. The model SAM was elaborated for illustrative purposes and is entirely fictitious. In fact, the bulk of the work to be done in order to incorporate a SAM into PEP-1-t, is reformatting it using the same structure as the fictitious one. Once this is done correctly, only minor changes will need to be made to the GAMS code itself.

In a dynamic framework, additional data is required, and if it is not available, the modeler needs to make assumptions in order to calibrate the missing information. In the present document, as in DLRM, we assume that the only data available is the SAM data. It might be the case that the reader has more information in hand and may relax some of the assumptions suggested in the documentation. That being said, we strongly suggest that the user first follow the guidelines in PEP-1-t, verify that the model runs properly and then add any other information, making sure that the calibration is consistent with the additional data available. We thus believe it is important to first review the accounts and flows in the fictitious SAM in order to put the real SAM the user wishes to use in the same format. As the present guide is intended for practitioners, we then suggest following the same sections as in the GAMS code. Section 5 then discusses what type of information is relevant for recursive dynamic modeling and how to adjust the calibration accordingly.

### 3 SAM<sup>5</sup>

A SAM is a snapshot of an economy, for a given year. It must be square because each account appears both as a row and as a column of the table: the account's income is registered in the corresponding row, and its expenditures in the corresponding column. The value in every cell of the matrix, therefore, is an expenditure for the corresponding column-account, and an income for the corresponding row-account. The SAM to be used as the basis of a CGE model must be balanced, meaning that, for each account, the sum of income from all sources must be exactly equal to the sum of expenditures.

The model SAM is presented in figure 1. The order in which the accounts are listed does not have to be followed; any order will do. In adapting his/her SAM to the format used in PEP-1-t, the user must however stick to the following guidelines:

- The user must supply his/her SAM as an Excel file.
- Accounts in the SAM should be grouped into five main categories: factors of production, institutions (or agents), commodities, industries and accumulation.
- Each account must have two titles (both in rows and in columns): one representing the set of accounts to which it belongs, the other indicating which element it is in that set. In order to facilitate the integration of the SAM in the GAMS code, the user must use the exact same titles as the ones in the shaded areas of the first two rows and columns of figure 1. A particular account may be referred to using both the first- and second-level titles, with a period mark in between (ex.: L.USK or AG.HUP).

---

<sup>5</sup> Figure F1 in DLRM presents the linkages between each cell of the fictitious SAM and the variables in the model.

- Only flows appearing in the shaded areas of the fictitious SAM are taken into account in the PEP-1-t model. Consequently, the matrix must be balanced when all non-shaded areas are blank. Conversely, the real SAM to be used does not have to show values for every flow present in the fictitious one.

The remainder of this section presents for each group of accounts the titles to be used and the flows the model can accommodate. For each subsection, income and expenditures appearing in the SAM are explained through two summary tables.

**Figure 1: A fictitious SAM for PEP-1-t v.2.0 (left panel)**

		L	L	K	K	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG
		USK	SK	CAP	LAND	HRP	HUP	HRR	HUR	FIRM	GVT	TD	TM	TI	USK	SK	CAP	LAND	ROW
L	USK																		
L	SK																		
K	CAP																		
K	LAND																		
AG	HRP	5915	5078	526	1132														
AG	HUP	7300	4697	1054							139								
AG	HRR	872	1242	526	3602														
AG	HUR	1210	1010	6221	100					1900									
AG	FIRM			4741	488			10			10								
AG	GVT					40	100		122	81		2308	2500	4375	1843	137	46		
AG	TD					44	428	146	390	1300									
AG	TM																		
AG	TI																		
AG	USK																		
AG	SK																		
AG	CAP																		
AG	LAND																		
AG	ROW			42	911				10	370	30								
J	AGR																		
J	IND																		
J	SER																		
J	ADM																		
I	AGR					6338	4496	2441	1755										
I	FOOD					2504	3050	950	2400										
I	OTHIND					1090	1601	1003	3043										
I	SER					2635	3515	1620	2426										
I	ADM										8255								
X	AGR																		7576
X	FOOD																		241
X	OTHIND																		1200
X	SER																		2653
OTH	INV						72	295	1598	1231									5425
OTH	VSTK																		
OTH	TOT	15297	12027	13110	6233	12651	13190	6242	10441	5249	9665	2308	2500	4375	1843	137	46		17095

**Figure 1: A fictitious SAM for PEP-1-t v.2.0 (right panel)**

		J	J	J	J	I	I	I	I	I	X	X	X	X	OTH	OTH	OTH
		AGR	IND	SER	ADM	AGR	FOOD	OTHIND	SER	ADM	AGR	FOOD	OTHIND	SER	INV	VSTK	TOT
L	USK	10002	2289		3006												15297
L	SK	910		10147	970												12027
K	CAP	2086	7015	4009													13110
K	LAND	6133	100														6233
AG	HRP																12651
AG	HUP																13190
AG	HRR																6242
AG	HUR																10441
AG	FIRM																5249
AG	GVT	-1693	-293								99						9665
AG	TD																2308
AG	TM					500	200	1800									2500
AG	TI					684	1400	1554	737								4375
AG	USK	1500	343														1843
AG	SK	137															137
AG	CAP	46															46
AG	LAND																
AG	ROW					2613	1458	9970	1691								17095
J	AGR					17834	250		210		7417						25711
J	IND					400	15698		20			241	1180				17539
J	SER						150		18532					2653			21335
J	ADM									8255							8255
I	AGR	2715	1402	1327	93										2164	-600	22131
I	FOOD	825	770	1600											6857	200	19156
I	OTHIND	1102	2930	455	2100												13324
I	SER	1948	2983	3797	2086	100					60		20				21190
I	ADM																8255
X	AGR																7576
X	FOOD																241
X	OTHIND																1200
X	SER																2653
OTH	INV																8621
OTH	VSTK														-400		-400
OTH	TOT	25711	17539	21335	8255	22131	19156	13324	21190	8255	7576	241	1200	2653	8621	-400	

### 3.1 Factors of production

Sets: *L – Labor; K – Capital*

As mentioned previously, the PEP model can manage multiple types of workers and capital, although the model will run perfectly if there is only one element in each set. Labor is referred to in the model by set L. In the SAM, the first title that should appear on top of the columns and on the left of the rows for the labor accounts should therefore be L and any other title will imply changes in the GAMS code.

There must be at least one labor category. In our example, there are two types of labor which are called USK (unskilled) and SK (skilled). In adapting her/his own SAM, the user is free to use the element labels that are relevant given her/his data. The names of the elements must respect GAMS syntax and must be the same in rows and in columns.

The same rules apply to capital. The title for the set of accounts must be K, but the number of capital categories and their names are up to the modeler. In our example, there are two types of capital, CAP and LAND.

Both labor and capital receive income solely from the industries. The model will not accommodate any other income. It is not necessary that each industry use every type of labor and capital; it can manage sectors that do not use capital, like the industry ADM in our example, as well as industries that do not use labor (but not industries that use neither).

Labor income is distributed amongst the different types of households; no other agent can receive revenue from labor. On the contrary, all agents can receive income from capital.

**Figure 2: Factors of production – Income**

		J	J	J	J
		AGR	IND	SER	ADM
L	USK	<i>Wages paid by industry J to type L labor</i>			
L	SK				
K	CAP	<i>Remuneration of type K capital paid by industry J</i>			
K	LAND				

**Figure 3: Factors of production – Expenditures**

		L	L	K	K
		USK	SK	CAP	LAND
AG	HRP	Type L labor income received by type H household		Type K capital income received by agent AG	
AG	HUP				
AG	HRR				
AG	HUR				
AG	FIRM				
AG	GVT				
AG	ROW				

## 3.2 Institutions<sup>6</sup>

*Set: AG – Agents*

Institutions, or agents, are represented in the model by set AG. The first title for these accounts, both in rows and in columns, should therefore be AG. There must be four types of agents: households, firms, government and the rest of the world.

### 3.2.1 Households

*Set: H – Households (subset of set AG)*

There can be several types of households but there needs to be at least one. In our fictitious SAM, there are four types of households named HRP, HUP, HRR and HUR. These acronyms are evocative of the following categories: rural poor, urban poor, rural rich and urban rich. These are the acronyms that are used in the standard PEP-1-t GAMS code. If the acronyms in the user's SAM are different, then it is important that the same acronyms be used in the GAMS code as in the SAM. Of course, the names should be the same in corresponding rows and columns.

Households may receive income from labor and capital, and in the form of transfers from other agents. They use their income to pay income taxes, make transfers to other agents, purchase commodities, and save.

**Figure 4: Households – Income**

		L	L	K	K	AG	AG	AG	AG	AG	AG	AG
		USK	SK	CAP	LAND	HRP	HUP	HRR	HUR	FIRM	GVT	ROW
AG	HRP	<i>Type L labor income received by type H household</i>		<i>Type K capital income received by type H household</i>		<i>Transfers from agent AG to household H</i>						
AG	HUP											
AG	HRR											
AG	HUR											

**Figure 5: Households – Expenditures**

		AG	AG	AG	AG
		HRP	HUP	HRR	HUR
AG	HRP	<i>Transfers from household H to agent AG</i>			
AG	HUP				
AG	HRR				
AG	HUR				
AG	FIRM				
AG	GVT	<i>Income taxes paid by household H</i>			
AG	TD				
AG	ROW	<i>Transfers from household H to agent ROW</i>			
I	AGR	<i>Consumption of commodity by household HI</i>			
I	FOOD				
I	OTHIND				
I	SER				
I	ADM				
OTH	INV	<i>Household H savings</i>			

<sup>6</sup> See Section 2 in DLRM for a complete description of the income and savings of each agent.



### 3.2.2 Firms

Set:  $F$  – Firms (subset of set AG)

As for the households, there needs to be at least one firm, and the model can manage multiple types of businesses. Here again, the number of firm categories will depend on data availability and on the structure of the user's SAM. In our example, there is only one firm, which is called FIRM.

Firms' income consists of capital income and transfers; they use it to pay direct taxes, make transfers to other agents, and save.

**Figure 6: Firms – Income**

		K	K	AG	AG	AG	AG	AG	AG	AG
		CAP	LAND	HRP	HUP	HRR	HUR	FIRM	GVT	ROW
AG	FIRM	Type K capital income received by type F firms		Transfers from agent AG to firm F						

**Figure 7: Firms – Expenditures**

		AG
		FIRM
AG	HRP	Transfers from firm F to agent AG
AG	HUP	
AG	HRR	
AG	HUR	
AG	FIRM	
AG	GVT	Direct taxes paid by firm F
AG	TD	Transfers from firm F to agent ROW
AG	ROW	
OTH	INV	Savings

### 3.2.3 Government

Element GVT in set AG

The model can only manage with a single government. In other words, it cannot deal with SAMs that show multiple government levels. To avoid having to modify the GAMS code, government should be called GVT.

Furthermore, as the model allows for multiple tax instruments, additional accounts should be created for the government. Here again, the titles for these accounts should stick to the ones used in the fictitious SAM.

- The government may receive money from households and firms in two different ways: through direct taxation and through transfers. The account AG.TD collects direct taxes, while transfers should appear directly at the intersection of row AG.GVT and columns representing households (AG.H) and firms (AG.F). The government will then receive the

- total amount of direct taxes collected from households and firms (total of row AG.TD), as shown at the intersection of row AG.GVT and column AG.TD.
- The government also collects taxes on imports. They are registered at the intersection of row AG.TM and commodity-account columns, labeled I. As for direct taxes, the sum of import duties collected on the different commodities (i.e. total of row AG.TM) is then paid to the government (intersection of row AG.GVT and column AG.TM).
  - Account AG.TI represents the other indirect taxes collected on commodities sold locally. Similarly to AG.TM, account AG.TI receives its income from the different commodity accounts (columns labeled I), and the sum of indirect taxes (i.e. total of row AG.TI) is a source of income for the government (intersection of row AG.GVT and column AG.TI).
  - Indirect taxes can also be applied on exports. Contrary to the three preceding types of taxes, they are not collected in a separate account, and they should appear at the intersection of row AG.GVT and of the exported commodity columns labeled X.
  - Three types of taxes can be paid by productive activities (industries). Taxes on production should appear directly at the intersection of row AG.GVT and of columns representing the industries, labeled J.
  - There can also be taxes paid by industries on their wage bills. These are collected from each industry by wage-bill tax accounts. These tax-collecting accounts belong to the agent category of accounts, labeled AG. To allow for the tax rates to vary according to the labor categories, there are as many wage-bill tax accounts as there are elements in set L. Hence, the intersection of row AG.USK and of columns J, will represent the taxes paid by industries on salaries paid to workers of type USK. The sum of each wage-bill tax row will then appear as an income for the government (intersection of row AG.GVT and columns AG.USK and AG.SK<sup>7</sup>).
  - The same mechanism applies to capital. The model considers taxes paid on the remuneration of capital by the productive activities. The user should add as many capital-tax accounts as there are types of capital, under the main first-level label AG. The taxes on the remuneration of capital appear under the industry columns, and the sum of taxes for each type of capital is then paid to the government.

**Figure 8: Government – Income**

		K	K	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	AG	
		CAP	LAND	HRP	HUP	HRR	HUR	FIRM	GVT	TD	TM	TI	USK	SK	CAP	LAND	ROW	
AG	GVT	Type K capital income received by agent GVT		Transfers from H and F to GVT						Total direct taxes	Total import duties	Total indirect taxes	Total taxes on wage bill L	Total taxes on capital K		Transfers from ROW to GVT		
AG	TD			Direct taxes paid by H and F														
AG	TM																	
AG	TI																	
AG	USK																	
AG	SK																	
AG	CAP																	
AG	LAND																	

<sup>7</sup> This illustrates the use of two-level labels: wages and salaries are paid to accounts L.USK and L.SK.

**Figure 8: Government – Income (continued)**

		J	J	J	J	I	I	I	I	I	X	X	X	X
		AGR	IND	SER	ADM	AGR	FOOD	OTHIND	SER	ADM	AGR	FOOD	OTHIND	SER
AG	GVT	Net taxes on production paid by industry J									Taxes on exports X			
AG	TD													
AG	TM													
AG	TI													
AG	USK	Taxes on wage bill L paid by industry J												
AG	SK	Taxes on the remuneration of type K capital paid by industry J												
AG	CAP													
AG	LAND													

**Figure 9: Government – Expenditures**

		AG GVT	AG TD	AG TM	AG TI	AG USK	AG SK	AG CAP	AG LAND
AG	HRP	<i>Transfers from GVT to H and F</i>							
AG	HUP								
AG	HRR								
AG	HUR								
AG	FIRM								
AG	GVT		<i>Total direct taxes</i>	<i>Total import duties</i>	<i>Total indirect taxes</i>	<i>Total taxes on wage bill L</i>	<i>Total taxes on capital K</i>		
AG	ROW	<i>Transfers from GVT to ROW</i>							
I	AGR	<i>Current public consumption of commodity I</i>							
I	FOOD								
I	OTHIND								
I	SER								
I	ADM								
OTH	INV	<i>Savings</i>							

To summarize, government receives income from direct taxation (account AG.TD), import duties (account AG.TM), indirect taxes on locally consumed commodities (account AG.TI), taxes on exports (account AG.GVT), wage-bill taxes (accounts AG.USK and AG.SK), taxes on the remuneration of capital (accounts AG.CAP and AG.LAND) and taxes on production (account AG.GVT). It can also receive transfers from other agents. It uses its income to purchase commodities, make transfers and save.

### 3.2.4 Rest of the world

*Element ROW in set AG*

As for government, there can only be one rest of the world, i.e. the model cannot manage multiple trading partners. In order to facilitate inclusion of the SAM into the GAMS code, this agent must be called ROW.

The rest of the world receives its income as capital income, as transfers from domestic agents, denoted AGD in the GAMS code, and from imports. It spends in the local economy through the purchase of export commodities and makes transfers to domestic agents. The surplus of rest-of-the-world income over its expenditures, i.e. its savings (equal to minus the current account balance) appear at the intersection of row OTH.INV and of column AG.ROW

**Figure 10: Rest of the world – Income**

		K	K	AG	AG	AG	AG	AG	AG	I	I	I	I	I
		CAP	LAND	HRP	HUP	HRR	HUR	FIRM	GVT	AGR	FOOD	OTHIND	SER	ADM
AG	ROW	<i>Type K capital income received by agent ROW</i>		<i>Transfers from AGD to ROW</i>						<i>Imports of commodity I (excluding taxes and margins)</i>				

**Figure 11: Rest of the world – Expenditures**

		AG ROW
AG	HRP	<i>Transfers from ROW to AGD</i>
AG	HUP	
AG	HRR	
AG	HUR	
AG	FIRM	
AG	GVT	
X	AGR	<i>Exports of commodity X (including taxes and margins)</i>
X	FOOD	
X	OTHIND	
X	SER	
OTH	INV	<i>Savings</i>

### 3.3 Commodities

*Set: I – Commodities*

Commodities consist of the different goods and services produced and/or consumed in the economy. They are referred to, in the GAMS code, as elements of set I. To accommodate for rectangular input-output tables, commodities and industries are represented through distinct sets, respectively I and J. If a user's SAM has exactly the same elements for industries and commodities, she/he should nevertheless create two separate sets, even if their elements are identical<sup>8</sup>.

It is also important to note that the PEP-1-t model does not manage re-exports (to do so, it would have to be modified accordingly). Only commodities that are produced locally can be exported, and imports can solely be purchased by local agents and industries.

In the SAM, commodities appear twice: once in account category I and a second time in account category X. Although this might seem repetitive, it is very convenient to represent exports at both producer and purchaser prices. Besides, it facilitates taking into account cases where, for a given

<sup>8</sup> It might be worth noting that GAMS can handle identical names for elements in different sets.

commodity, industries sell different proportions of their production on the local and international markets.

The list of commodities obviously depends on the user's SAM. In our example, there are five commodities named AGR, FOOD, OTHIND, SER and ADM. The user's commodity acronyms must be the same as will be used later in the GAMS code as elements of set I. Once again, these acronyms should be identical in corresponding rows and columns, as well as under account categories I and X. If some commodities are not exported, the user can simply omit them under category X, as in our example for commodity ADM.

Demand for commodities I consists of final private demand (purchases from households), current public consumption (purchases from government), intermediate demand (purchases from industries), demand for investment purposes and inventory changes<sup>9</sup>. Each purchaser buys a composite commodity which is composed of local production and/or imports.

The PEP-1-t model also takes into account trade and transport margins. In our example, only the commodity SER is used as a margin, but in a detailed SAM, there can be more than one margin commodity. Margins are applied to local purchases (thus under the I-columns) and to exports (under X-columns) and constitute an additional source of demand for the relevant commodities.

**Figure 12: Commodities – Income**

		AG	AG	AG	AG			J	J	J	J
		HRP	HUP	HRR	HUR	AG	AG	AGR	IND	SER	ADM
I	AGR	<i>Consumption of commodity I by household H</i>				<i>Public consumption of commodity I by GVT</i>		<i>Intermediate demand of commodity I by industry J</i>			
	FOOD										
	OTHIND										
	SER										
	ADM										
X	AGR					<i>Exports of commodity X (including taxes and margins)</i>					
	FOOD										
	OTHIND										
	SER										

		I	I	I	I	I	I	X	X	X	X	OTH	OTH
		AGR	FOOD	OTHIND	SER	ADM	AGR	FOOD	OTHIND	SER		INV	VSTK
I	AGR	Purchases of commodity I as margins on local purchases of composite commodity I					Purchases of commodity I as margins on exports of exports X					Demand of composite commodity I for investment purposes (GFCF)	Inventory changes of composite commodity I
I	FOOD												
I	OTHIND												
I	SER												
I	ADM												
X	AGR												
X	FOOD												
X	OTHIND												
X	SER												

<sup>9</sup> See section 3 in DLRM for equations related to demand.

**Figure 13: Commodities – Expenditures**

		I	I	I	I	I	X	X	X	X
		AGR	FOOD	OTHIND	SER	ADM	AGR	FOOD	OTHIND	SER
AG	GVT						Taxes on exports of commodity X			
AG	TM	Import duties on imports I								
AG	TI	Indirect taxes on commodity I								
AG	ROW	Imports of commodity I								
J	AGR	Supply of commodity I on the local market by sector J (excluding taxes and margins)					Supply of commodity X on the export market by sector J (excluding taxes and margins)			
J	IND									
J	SER									
J	ADM									
I	AGR	Purchases of commodity I as margins on local purchases of composite commodity I					Purchases of commodity I as margins on exports of exports X			
I	FOOD									
I	OTHIND									
I	SER									
I	ADM									

### 3.4 Industries

*Set: J – Industries*

Industries are represented in the GAMS code as elements of set J. As mentioned previously, the nomenclature of productive activities can be different from that of commodities. In our fictitious SAM, there are four industries: AGR, IND, SER and ADM. Once again, the user must use the same acronym for each industry as he will use in the GAMS code, and these acronyms must be identical in corresponding rows and columns.

Each industry can produce any commodity and sell it on the local market and/or on the export market. Local sales will appear at the intersection of J-rows and I-columns, while exports will appear under X-columns. All sales should be expressed at producer prices<sup>10</sup>.

To produce the different commodities, industries use production factors (i.e. labor and capital) as well as intermediate consumption<sup>11</sup>.

As mentioned in section 3.2.3, industries may pay taxes on their wage bill or on the remuneration of capital, or other taxes on production. Intermediate consumption appears at purchaser prices and thus includes indirect taxes, duties and margins.

<sup>10</sup> See section 4 in DLRM for equations related to supply.

<sup>11</sup> See section 1 in DLRM for equations related to production.

**Figure 14: Industries – Income**

		I	I	I	I	I	X	X	X	X
		AGR	FOOD	OTHIND	SER	ADM	AGR	FOOD	OTHIND	SER
J	AGR	<i>Supply of commodity I on the local market by industry J (excluding taxes and margins)</i>					<i>Supply of commodity X on the export market by industry J (excluding taxes and margins)</i>			
J	IND									
J	SER									
J	ADM									

**Figure 15: Industries – Expenditures**

		J	J	J	J
		AGR	IND	SER	ADM
L	USK	<i>Wages paid by industry J to type L labor</i>			
L	SK				
K	CAP	<i>Remuneration of type K capital paid by industry J</i>			
K	LAND				
AG	GVT	<i>Net taxes on production paid by industry J</i>			
AG	USK	<i>Taxes on wage bill L paid by industry J</i>			
AG	SK				
AG	CAP	<i>Taxes on the remuneration of capital K paid by industry J</i>			
AG	LAND				
I	AGR	<i>Intermediate demand of commodity I by industry J</i>			
I	FOOD				
I	OTHIND				
I	SER				
I	ADM				

### 3.5 Accumulation

The last account category in PEP-1-t consists in the savings and investment accounts. No set in the GAMS code is used to represent this category, so we chose the title OTH, which is also used as a header for the total (TOT) account. Accumulation is split into two accounts: savings/investment (INV) and inventory changes (VSTK). As part of the GAMS code directly refers to these accounts, the user must keep these exact same titles in her/his SAM (OTH, TOT, INV and VSTK); otherwise, the user must modify the GAMS code accordingly.

Savings from agents should appear at the intersection of row OTH.INV and of the corresponding AG-columns. Total savings, or the sum of row OTH.INV, will finance both gross fixed capital formation and inventory changes, which appear at the intersections of columns OTH.INV and OTH.VSTK respectively, and of the commodity rows (I).

**Figure 16: Accumulation – Income**

		AG	AG	AG	AG	AG	AG	AG	OTH
		HRP	HUP	HRR	HUR	FIRM	GVT	ROW	INV
OTH	INV	<i>Savings of agent AG</i>							
OTH	VSTK								<i>Total inventory changes</i>

**Figure 17: Accumulation – Expenditures**

		OTH INV	OTH VSTK
I	AGR	<i>Demand of composite commodity I for investment purposes (GFCF)</i>	<i>Inventory changes of composite commodity I</i>
I	FOOD		
I	OTHIND		
I	SER		
I	ADM		
OTH VSTK		<i>Total inventory changes</i>	

## 4 GAMS code

It is important to repeat that this document is intended to users who have some experience with CGE modeling and GAMS. This guide is by no means a tutorial for the software and we assume that the reader is already knowledgeable of the syntax. GAMS offers a user guide to which the reader can refer if required.

Also, in order to minimize the changes to be made in the code, all files should be saved in a single directory. The user will then create a project in that same directory and from then on, will not have to specify the complete path to access the different files. The directory should include the GAMS files (in our example PEP\_1-t\_v2\_0.gms, RESULTS PEP 1-t BAU.GMS, and RESULTS PEP 1-t SIM1.gms) and the Excel file of the SAM (SAM-V2\_0.xls). Note that the name of the Excel file in which the SAM is located must not include any space. Also, recent versions of GAMS can handle more recent versions of Excel, so the extension can be either XLS or XLSX.

Once the SAM is correctly formatted, changes to be made in the GAMS code should be straightforward. The remainder of this section follows the order in which the code is written.

### 4.1 Definition of sets

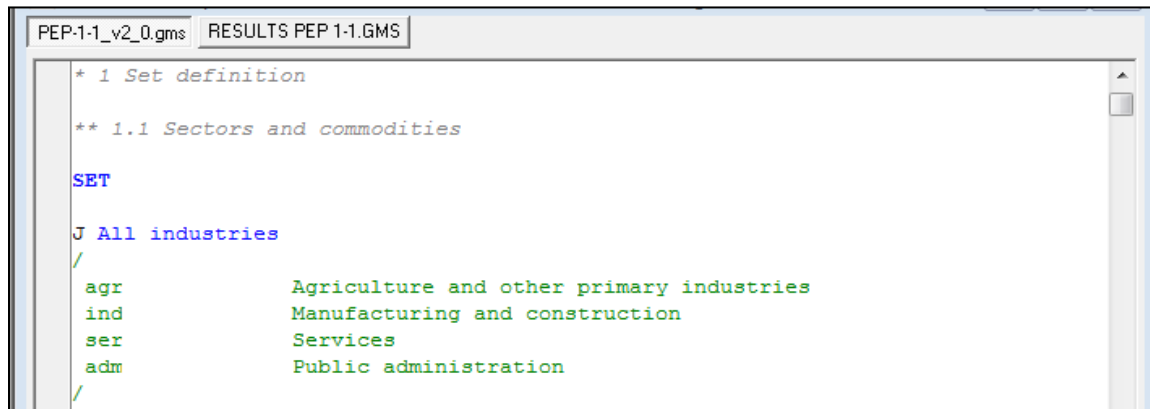
Following the GAMS code writing, the first step is to define the sets and their elements. As the rest of the coding uses these sets, it is not advisable to change their names.

#### 4.1.1 Sectors and commodities

Set J refers to the different industries in the model. The user can simply delete the list of elements in the PEP-1-t model (namely AGR, IND, SER and ADM) and replace it with the list of industries in her/his own SAM. The acronyms should be the same as under title J in the SAM.



**Figure 18: Definition of set J**

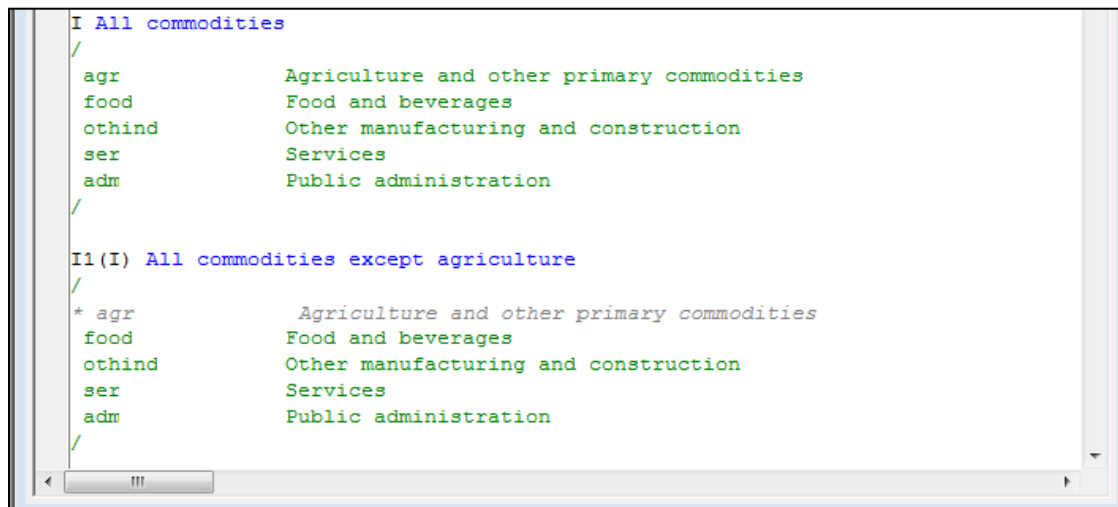


```
* 1 Set definition
** 1.1 Sectors and commodities
SET
J All industries
/
agr      Agriculture and other primary industries
ind      Manufacturing and construction
ser      Services
adm      Public administration
/
```

A similar procedure should be followed for set I, which refers to commodities in the model. In our example, there are five commodities, AGR, FOOD, OTHIND, SER and ADM. The user should replace these five elements with the acronyms used in her/his SAM under the title I.

Set I1 refers to all the elements in set I, but one. Indeed, by Walras' Law, we know that when all markets but one are in equilibrium, then the last one is automatically in equilibrium as well<sup>12</sup>. The equilibrium equation that states that supply is equal to demand is therefore defined over set I1. In order to verify that the model is correctly specified, an extra variable, called LEON, is added, which is equal to the difference between supply and demand for this last commodity. In our example, we removed element AGR, and the variable LEON is the difference between supply and demand for that specific commodity. For set I1, the user should copy and paste the elements of set I and take one element out.

**Figure 19: Definition of set I and subset I1**



```
I All commodities
/
agr      Agriculture and other primary commodities
food     Food and beverages
othind   Other manufacturing and construction
ser      Services
adm      Public administration
/

I1(I) All commodities except agriculture
/
* agr      Agriculture and other primary commodities
food     Food and beverages
othind   Other manufacturing and construction
ser      Services
adm      Public administration
/
```

<sup>12</sup> GAMS cannot handle systems with one or more redundant equations. See Appendix E in DLRM for a discussion of the implications of Walras' Law.

#### 4.1.2 Production factors

The PEP-1-t model allows for multiple labor and capital categories. Set L refers to the labor categories, set K to the ones related to capital. If only one type of labor (or capital) is present in the SAM, then the set should consist in a single element. The model will automatically accommodate to the size of these two sets. The acronyms used in set L must be the same as in the SAM under title L. Similarly, the names of the elements in set K must be identical to the ones under title K in the SAM.

**Figure 20: Definition of sets L and K**

```
** 1.2 Production factors

L Labor categories
/
  usk          Unskilled workers
  sk           Skilled workers
/

K Capital categories
/
  cap          Capital
  land         Land
/
```

#### 4.1.3 Agents

Set AG refers to the different institutions used in the model<sup>13</sup>. As mentioned previously, in our example, there are multiple households, one firm, the government and the rest of the world. The model will run if different assumptions are made (one single household and/or multiple firms), but it is important that the government element be called GVT and that the rest of the world be represented by element ROW. In fact, both these elements are specifically referred to in the GAMS code and any change in their names will imply further changes in the model. We therefore strongly suggest that the user use these names for those two elements of set AG.

**Figure 21: Definition of set AG**

```
AG All agents
/
  hrp          Poor rural households
  hup          Poor urban households
  hrr          Rich rural households
  hur          Rich urban households
  firm         Firms
  gvt          Government
  row          Rest of the world
/
```

<sup>13</sup> Note that the GAMS set AG is not the same as the AG category of SAM accounts. The latter includes, for instance, AG.CAP and AG.LAND.

From set AG, four subsets are created. The first one, AGNG refers to non-government agents and includes all elements of set AG except GVT. Similarly, subset AGD consists of domestic agents and includes all elements of set AG except ROW.

**Figure 22: Definition of subsets AGNG and AGD**

```

AGNG (AG) Non government agents
/
hrp      Poor rural households
hup      Poor urban households
hrr      Rich rural households
hur      Rich urban households
firm     Firms
* gvt    Government
row      Rest of the world
/

AGD (AG) Domestic agents
/
hrp      Poor rural households
hup      Poor urban households
hrr      Rich rural households
hur      Rich urban households
firm     Firms
gvt      Government
* row    Rest of the world
/

```

The last two subsets, H and F, include households and firms respectively. Once again, the user should not change these sets' names, because they are referred to later in the GAMS code. The list of elements in both sets must be the same as in the user's SAM.

**Figure 23: Definition of subsets H and F**

```

H (AG) Households
/
hrp      Poor rural households
hup      Poor urban households
hrr      Rich rural households
hur      Rich urban households
/

F (AG) Firms
/
firm     Firms
/

```

#### 4.1.4 Time periods

In a dynamic framework, variables are indexed with a time dimension, which is referred by using index T. In PEP-1-t GAMS code, there are 10 periods by default, but the user is free to choose any number **xxx** of periods by replacing 1\*10 with 1\***xxx** in Figure 24. Regardless of the number of periods included in set T, a subset T1 is created to refer to the first period only. This subset is used in the GAMS code and should therefore be defined as shown in the next figure.

**Figure 24: Definition of set T and subset T1**

```
** 1.4 Periods

T Time periods
/
1*10
/

T1(t) First period;
T1(t) = yes$(ord(t) eq 1);
```

#### 4.1.5 Aliases

The last part of the SET section defines a second name that could be used to refer to the sets just defined, through the ALIAS statement. For simplicity, we just added the letter “j” to the name of each set. As for the names for the different sets, and for the same reasons, it is not advisable to change the names used in the ALIAS statements.

**Figure 25: The alias statements**

```
ALIAS (j,jj)
ALIAS (i,ij)
ALIAS (ag,agj)
ALIAS (h,hj)
ALIAS (l,lj)
ALIAS (k,kj)
;
```

## 4.2 Parameters and benchmark variables definition

This part of the GAMS code defines the different parameters used in the model. It also creates one parameter for each variable; the name of the parameter is created by adding the letter “O” to the variable name. This facilitates the calibration process as well as the assignment of starting values for model resolution. In this part of the GAMS code, nothing should be changed if the user wants to utilize the PEP model exactly as it is.<sup>14</sup>

## 4.3 Data

### 4.3.1 Data from the SAM

If the user correctly followed the previous steps in formatting her/his SAM, this part will be straightforward, although a bit technical. There is a command in GAMS that allows converting Excel spreadsheets into GDX files, which can then be read in a GAMS code. That command is included in the recent versions of the GAMS software. The syntax for introducing the SAM into the code is the following.

```
$CALL GDXXRW.EXE file_name.xls par=SAM rng=sheet!range Rdim=2 Cdim=2
$GDXIN file_name.gdx
```

---

<sup>14</sup> The complete list of parameters, variables and equations can be found in Appendix A of DLRM.

It is important that only the highlighted parts be modified. Note that GAMS can read more recent versions of Excel files; make sure to use the correct extension (XLS or XLSX). In our example, as the fictitious SAM is located in the Excel file named SAM-V2\_0.xls, in the spreadsheet SAM, in cells A4 to AJ39, figure 26 shows the part of the GAMS code which introduces the SAM into the GAMS code.

### Figure 26: Introduction of the SAM

```
$CALL GDXXRW.EXE SAM-V2_0.xls par=SAM rng=SAM!A4:AJ39 Rdim=2 Cdim=2
$GDXXIN SAM-V2_0.gdx
$LOAD SAM
$GDXXIN
```

The assignment of values in the SAM to the different variables and parameters in the model is then straightforward, given that the titles are exactly the same as the ones discussed earlier. Figure 27 presents the part of the GAMS code in which this is done.

### Figure 27: Introduction of the SAM

```

CO(i,h)           = SAM('I',i,'AG',h);
CGO(i)            = SAM('I',i,'AG','gvt');
DSO(j,i)          = SAM('J',j,'I',i);
DDO(i)            = SUM[j,DSO(j,i)];
DIO(i,j)          = SAM('I',i,'J',j);
EXO(j,i)          = SAM('J',j,'X',i);
EXDO(i)           = SAM('X',i,'AG','ROW');
INVO(i)           = SAM('I',i,'OTH','INV');
VSTKO(i)          = SAM('I',i,'OTH','VSTK');
IMO(i)            = SAM('AG','ROW','I',i);
RKDO(k,j)         = SAM('K',k,'J',j);
LDO(l,j)          = SAM('L',l,'J',j);
SFO(f)            = SAM('OTH','INV','AG',f);
SGO               = SAM('OTH','INV','AG','GVT');
SHO(h)            = SAM('OTH','INV','AG',h);
SROWO             = SAM('OTH','INV','AG','ROW');
TDFO(f)           = SAM('AG','TD','AG',f);
TDHO(h)           = SAM('AG','TD','AG',h);
TICO(i)           = SAM('AG','TI','I',i);
TIKO(k,j)         = SAM('AG',k,'J',j);
TIMO(i)           = SAM('AG','TM','I',i);
TIPO(j)           = SAM('AG','GVT','J',j);
TIXO(i)           = SAM('AG','GVT','X',i);
TIWO(l,j)         = SAM('AG',l,'J',j);
TRO(ag,agj)       = SAM('AG',ag,'AG',agj);
lambda_RK(ag,k)   = SAM('AG',ag,'K',k);
lambda_WL(h,l)    = SAM('AG',h,'L',l);
tmrg(i,ij)        = SAM('I',i,'I',ij);
tmrg X(i,ij)      = SAM('I',i,'X',ij);

```

### 4.3.2 Other data

Besides the data taken from the SAM, some values must be assigned to the parameters in the model that cannot be calibrated (so-called “free” parameters).

**Figure 28: Exogenous parameters**

```
* Population growth
n(t)          = 0.03;
n1            = SUM[t1,n(t1)];
pop(t1)       = 1;
loop{t$[ORD(t) gt 1],
pop(t)        = pop(t-1)*[1+n(t-1)];
};

* Interest rate
IRO           = 0.04;

* Elasticity - Investment demand function
sigma_INV(k,j) = 2;

* Price elasticity (should be set equal to one when verifying model homogeneity)
eta           = 1;

* CES and CET elasticities
sigma_KD(j)   = 0.8;
sigma_LD(j)   = 0.8;
sigma_M(i)    = 2;
sigma_VA(j)   = 1.5;
sigma_X(j,i)  = 2;
sigma_XT(j)   = 2;

* Elasticity of international demand for exported commodity x
sigma_XD(i)   = 2;
```

- Population growth (parameter n) is set to 3% by default, for all periods. Of course, the user should put the value of population growth that is reflective of the country to be modeled. The population index (parameter pop) is then calibrated accordingly.
- The same is true for the interest rate; a value of 4% is assigned to the parameter IRO by default, but the user is free to change this value.
- The price elasticity (parameter eta) should be set equal to one when verifying model homogeneity. This parameter is used to adjust nominal values according to the price index. The user can assign a value different from one to this parameter, depending on her/his knowledge of the economy to be modeled.
- Values have been assigned to the elasticities used in CES and CET functions, as shown in figure 28. In our example, the same value has been assigned to all industries and commodities. Although these values will allow the model to run, the user should assign values that are specific to the economy being modeled, if possible.
- The elasticity of international demand for exported commodities (sigma\_XD) was assigned a value of two for all commodities. Once again, the user will want to eventually modify this value.

#### 4.3.2.1 LES parameters

It is assumed that households have Stone-Geary utility functions (from which derives the Linear Expenditure System, or LES)<sup>15</sup>. This specific functional form implies several parameters, and not all of them can be calibrated<sup>16</sup>. In the PEP-1-t code, we assigned a different income elasticity to each commodity, so the user will have to modify this part of the GAMS code to ensure that the names of the commodity elements correspond to the ones in the set definition (see 4.1.1 above). Hence, for every element of set I, there should be a value for the income elasticity of consumption. Finally, the same value has been assigned to the Frisch parameter for every household; the user might once again assign a value that better represents the economy she/he wishes to model.

**Figure 29: LES parameters**

```
** LES parameters
frisch(h)      = -1.5;
sigma_y('agr',h)= 0.7;
sigma_y('food',h)
               = 1.1;
sigma_y('othind',h)
               = 1.1;
sigma_y('ser',h)= 1.05;
sigma_y('adm',h)= 1.05;
```

#### 4.3.2.2 Intercepts of transfers, direct taxes and savings

In household savings and transfers-to-government functions, and in income-tax functions, one can choose to assign a value to the intercept and calibrate the slope accordingly, or the other way around. This type of modeling can be useful to take into account known marginal savings or taxation rates or to deal with negative average saving rates in cases where savings are negative for some household groups. When no specific information is available, one can simply set the intercepts to zero and calibrate an average rate: this is what we have done in our example. Once again, the user is free to do otherwise<sup>17</sup>.

**Figure 30: Intercepts**

```
sh0(h)         = 0;
tr0(h)         = 0;
ttdf0(f)       = 0;
ttdh0(h)       = 0;
```

---

<sup>15</sup> See section 3 of DLRM.

<sup>16</sup> In other words, the data does not contain enough information to calibrate all parameters, so some of them are “free” and must be assigned a specific value.

<sup>17</sup> See subsections 2.1, 2.3 and 2.5 in DLRM.

### 4.3.2.3 Exogenous prices

The base value of some prices is arbitrary, insofar as it is constrained only by the price  $\times$  quantity product. In such cases, the arbitrary value assigned to the price implicitly determines the measurement unit of the quantity. The most convenient arbitrary price is obviously 1, although the calibration procedure can automatically manage prices with different values.

**Figure 31: Prices**

eO	= 1;
PEO (i)	= 1;
PLO (i)	= 1;
PWMO (i)	= 1;
WO (1)	= 1;

## 4.4 Calibration

The rest of the calibration does not need to be adapted, if the SAM structure is the same as in our example, and if no other information is available. If so, it should not be modified. The user interested in further information on the calibration process will find a thorough explanation of every step in Appendix F of DLRM.

That being said, it could be the case that the user has in hand additional information on variables that are specific to dynamic modeling. For example, if the user has some information on public investment, the suggested calibration of investment should be modified. The calibration procedures would also be different if the user has information on sectoral capital stock, rate of return, and/or investment. In any case, we strongly suggest that the user, in a first step, do not depart from the calibration hypothesis in PEP 1-t. In fact, by doing so, the model will replicate a regular growth path in which all prices remain the same and all volumes and values grow at the same rate as the population. This step might be considered as the equivalent of the homogeneity test in a static framework, simply to verify that indeed the model is correctly specified. When this first step is done, then the user can depart from these hypotheses and use the data he/she has in hand.

## 4.5 Model

The part of the GAMS code which is commonly referred to as the model itself, includes six main sections. The first section defines the variables (both endogenous and exogenous) that are used in the model. The user does not need to modify anything in this section.

The second part consists of the definition of equations where, once again, the user does not need to change anything. In the third part, which consists of the equations *per se*, only the last equation should be modified. Indeed, and as mentioned earlier, the WALRAS equation evaluates the difference between supply and demand for the commodity that was removed from set I to create subset I1. Figure 32 shows that the commodity AGR is the commodity that has been set aside in our example. The modeler should replace AGR with the relevant acronym.



**Figure 32: Walras equation**

```
WALRAS(t) ..      LEON(t) =e= Q('agr',t)-SUM[h,C('agr',h,t)]-CG('agr',t)
                  -INV('agr',t)-VSTK('agr',t)-DIT('agr',t)
                  -MRGN('agr',t);
```

The fourth part consists in the initialization of variables to their benchmark value, i.e. their value in the SAM. Assuming that the SAM has been correctly formatted, nothing should be changed in that section.

Finally, the exogenous variables are defined. In our example, the nominal exchange rate is used as the *numeraire* of the model, the current account is fixed and so are the current government expenditures on goods and services. Labor supply, world prices and inventory changes are also exogenous.

**Figure 33: Model closure**

```
e.fx(t)           = 1;
CAB.fx(t)          = CABO*pop(t);
CMIN.fx(i,h,t)     = CMINO(i,h)*pop(t);
G.fx(t)            = GO*pop(t);
IND.fx(k, pub, t) $KDO(k, pub)
                  = INDO(k, pub)*pop(t);
KD.fx(k, j, t1) $KDO(k, j)
                  = KDO(k, j);
LS.fx(l, t)        = LSO(l)*pop(t);
PWM.fx(i, t)       = PWMO(i);
PWX.fx(i, t)       = PWXO(i);
VSTK.fx(i, t)      = VSTKO(i)*pop(t);

** 6.1.3 Rates and intercepts
sh0.fx(h, t)       = sh0O(h)*pop(t);
sh1.fx(h, t)       = sh1O(h);
tr0.fx(h, t)       = tr0O(h)*pop(t);
tr1.fx(h, t)       = tr1O(h);
ttdf0.fx(f, t)     = ttdf0O(f)*pop(t);
ttdf1.fx(f, t)     = ttdf1O(f);
ttdh0.fx(h, t)     = ttdh0O(h)*pop(t);
ttdh1.fx(h, t)     = ttdh1O(h);
ttic.fx(i, t)      = tticO(i);
ttik.fx(k, j, t)   = ttikO(k, j);
ttim.fx(i, t)      = ttimO(i);
ttip.fx(j, t)      = ttipO(j);
ttiw.fx(l, j, t)   = ttiwO(l, j);
ttix.fx(i, t)      = ttixO(i);
```

The user might want to make other choices regarding model closure. We suggest, however, that she/he first run the model using these rules to verify that there are no errors in the code. Once this verification is done, then the modeler can choose different hypotheses. That being said, the user should keep in mind that the model must be square, meaning that there should be as many endogenous variables as there are equations. Hence, the user cannot fix an additional variable unless she/he releases one constraint.

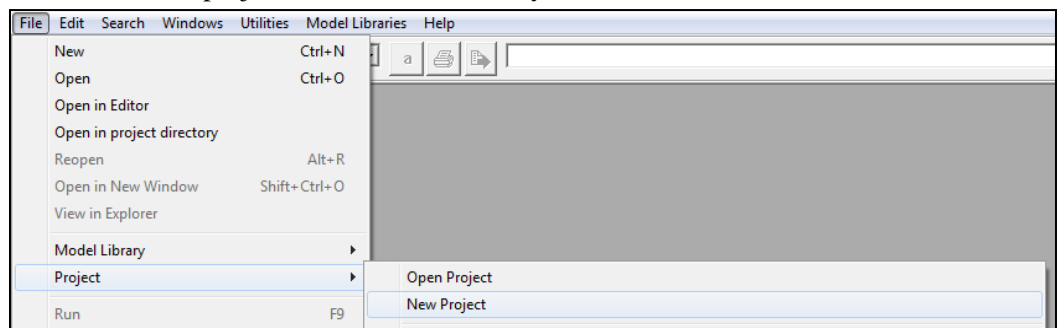
Regardless of the closure choices, this section is where the user will introduce simulations. Typically, a modeler will want to shock exogenous variables (i.e. the ones that appear with the suffix .FX), or tax rates. In all cases, the files RESULTS PEP 1-t BAU.GMS and RESULTS PEP 1-t SIM1.GMS will automatically produce two files containing both the reference values (usually referred to as BAU, “business as usual”) and the solution values of all variables in the model. These files are called RESULTS. Their contents are identical, but their formats are different. One of them has a GDX extension, and it can be opened directly in GAMS; the other has an XLS extension, and it can be opened in Excel. The user will find both files in the same directory where all other files are located and where she/he defined his project.

## 5 Troubleshooting

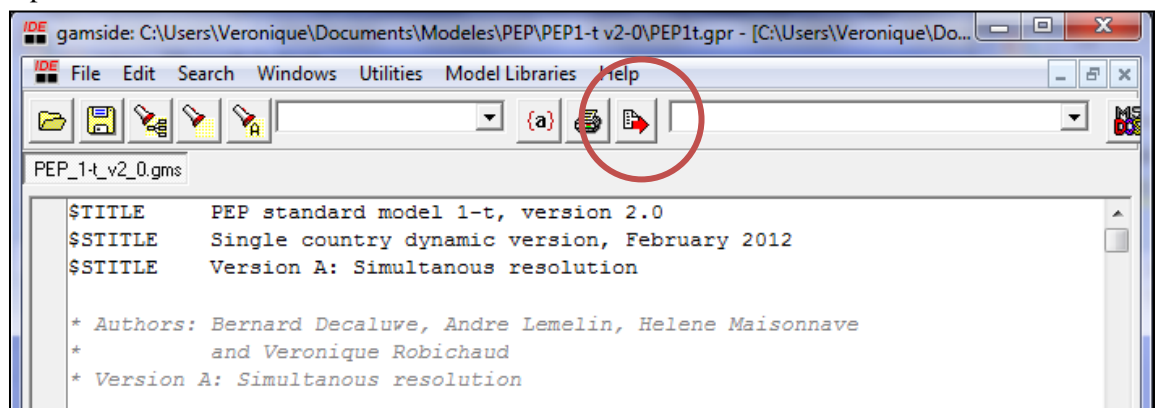
If the model does not run properly, the user may follow the following steps in order to identify the problem.

### 1. Run the standard PEP-1-t model

- a. Copy the all files of the PEP-1-t model in the same directory.
- b. Create a GAMS project in this same directory



- c. Open the file PEP-1-t\_v2\_0.gms.
- d. Run the model by pressing F9 or by clicking on the red arrow that appears on the top of the GAMS window.

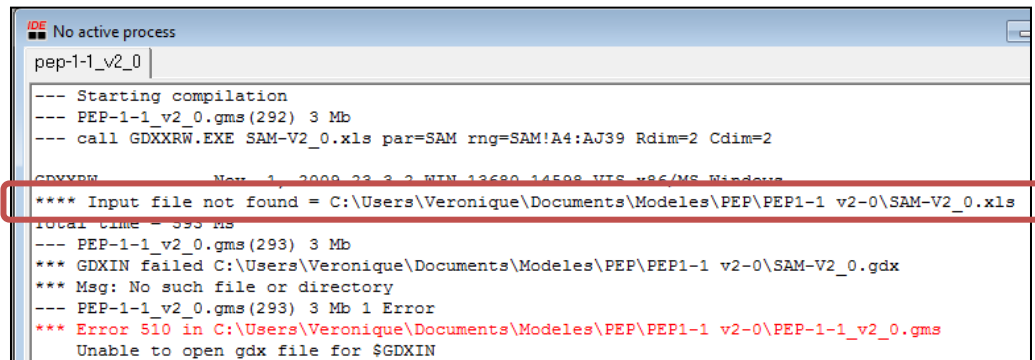


- e. The model should run properly.

If it does not, it must be for one of the three following reasons:

**i. All files are not in the same directory.**

If the SAM cannot be found, the process window in GAMS will show:

A screenshot of the GAMS process window titled 'No active process'. The window shows the command line 'pep-1-1\_v2\_0' and the output of the compilation process. The output includes the command 'call GDXXRW.EXE SAM-V2\_0.xls par=SAM rng=SAM!A4:AJ39 Rdim=2 Cdim=2'. A red rectangle highlights the error message: '\*\*\*\* Input file not found = C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\SAM-V2\_0.xls'. Below this, it says 'Total time = 395 MS', '\*\*\* GDGIN failed C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\SAM-V2\_0.gdx', '\*\*\* Msg: No such file or directory', '--- PEP-1-1 v2\_0.gms(293) 3 Mb 1 Error', and '\*\*\* Error S10 in C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\PEP-1-1\_v2\_0.gms Unable to open.gdx file for \$GDGIN'.

Make sure that all files (PEP-1-t\_v2\_0.gms, SAM-V2\_0.xls, RESULTS PEP 1-t BAU.gms and RESULTS PEP 1-t SIM1.gms) appear in the same directory as the one where the project was created.

**ii. The user does not own a GAMS license.**

The number of equations in PEP-1-t exceeds 300, which is the maximum size the free version can handle. The user should therefore own a GAMS license.

**iii. The user has an old version of GAMS.**

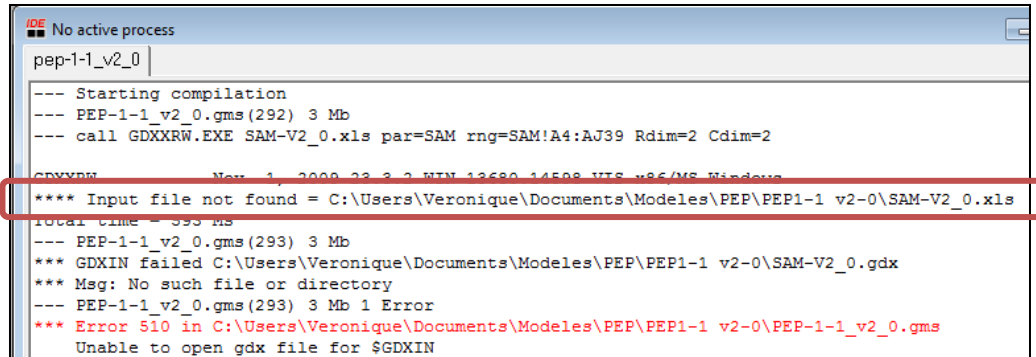
The user has an older version of GAMS that cannot handle the GDXXRW.EXE command. The user should download the most recent version of GAMS at [www.gams.com](http://www.gams.com).

**2. Run your version of PEP-1-t without any shock**

If the model does not run, the user might want to check the following elements:

**i. All files are not in the same directory.**

If the SAM cannot be found, the process window in GAMS will show:



```
pep-1-1_v2_0
--- Starting compilation
--- PEP-1-1_v2_0.gms(292) 3 Mb
--- call GDXXRW.EXE SAM-V2_0.xls par=SAM rng=SAM!A4:AJ39 Rdim=2 Cdim=2
GDXXRW Nov 1, 2009 22:32 WIN 12690 14508 VLS x86/MS Windows
**** Input file not found = C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\SAM-V2_0.xls
Total time = 333 MS
--- PEP-1-1_v2_0.gms(293) 3 Mb
*** GDXXIN failed C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\SAM-V2_0.gdx
*** Msg: No such file or directory
--- PEP-1-1_v2_0.gms(293) 3 Mb 1 Error
*** Error 510 in C:\Users\Veronique\Documents\Modeles\PEP\PEP1-1 v2-0\PEP-1-1_v2_0.gms
Unable to open.gdx file for $GDXXIN
```

Make sure that all files (PEP-1-t\_v2\_0.gms, SAM-V2\_0.xls, RESULTS PEP 1-t BAU.gms and RESULTS PEP 1-t SIM1.gms) appear in the same directory as the one where the project was created.

**ii. The user does not own a GAMS license.**

The number of equations in PEP-1-t exceeds 300, which is the maximum size the free version will handle. The user should therefore own a GAMS license.

**iii. The user has an old version of GAMS.**

The user has an older version of GAMS that cannot handle the GDXXRW.EXE command. The user should download the most recent version of GAMS at [www.gams.com](http://www.gams.com).

**iv. Division by zero.**

In this document, we have reviewed the specificities that cannot be handled by PEP-1-t. The user should make sure that her/his SAM respects those constraints. Division by zero can also happen if no values were assigned to parameters or variables. The user should check that she/he has assigned a value to all parameters (see section 4.3.2). Finally, no comma should appear in the description of set elements.

**v. Other syntax error.**

Correct the errors by comparing the program file to the original one (PEP\_1-t\_v2\_0.gms). For further information on how to debug a GAMS code, the user should refer to Maisonnave et al. (2012).

When the model runs without error, and if the SAM has been correctly formatted, then the model should reproduce the benchmark data. This is verified in the process window, where the first input point must be very close to zero.

Iter	Phase	Ninf	Infeasibility	R/Gmax	NSR	Step	InTr	MX	OK
0	0		4.6383541452E-10	(Input point)					
				Pre-triangular equations:				69	
				Post-triangular equations:				11	
1	0		4.6383541452E-10	(After pre-processing)					
2	0		5.3956838997E-14	(After scaling)					
2			5.3956838997E-14						

If the first input point is not close to zero, it might be for one of the following reasons:

**i. The SAM is not balanced.**

The model should be built using a balanced SAM. The user should verify in the Excel file that the row- and column-sums are exactly the same for every account.

**ii. Some cells in the SAM are not modeled.**

Only the cells in the gray areas of Figure 1 are modeled in PEP-1-t. The user should make sure that there are no flows outside these areas. Section 3 of this document explains how the SAM should be formatted.

**iii. Inconsistency between elements in the SAM and in the GAMS code.**

The elements of the different sets must be identical in the definition of the sets at the beginning of the GAMS code as the ones that figure in the SAM.

**iv. Incorrect initialization.**

If the SAM is balanced and shows flows that are considered in PEP-1-t, the model will run properly. If not, it is probably because the user changed some assignment, either in the calibration, or in the last part of the GAMS code. We suggest that the user stick to the original code.

**v. Incorrect values for the exogenous variables.**

The exogenous variables should be set at their initial values. No shock should be introduced in order to verify that the model reproduces the benchmark data.

**3. Run your version of PEP-1-t with a shock**

If the preceding steps have been followed successfully, then the model should be able to run simulations correctly. To verify that it does, the user should run a shock and check that the process window shows “feasible solution to a square system” and “normal completion”.

```

Reading Data

  Iter Phase Ninf   Infeasibility   RGmax   NSB   Step InItr MX OK
    0    0         3.5813771058E+03 (Input point)
                                Pre-triangular equations:    69
                                Post-triangular equations:    11
    1    0         4.2408039244E+03 (After pre-processing)
    2    0         1.2955715160E+00 (After scaling)
   10    0    0   9.6241317693E-10                1.0E+00      F  T

** Feasible solution to a square system.

--- Restarting execution
--- PEP-1-1_v2_0.gms(1276) 2 Mb
--- Reading solution for model PEP11
--- PEP-1-1_v2_0.gms(1276) 2 Mb
--- Executing after solve: elapsed 0:00:01.013
--- PEP-1-1_v2_0.gms(1671) 3 Mb
--- PEP-1-1_v2_0.gms(1671) 3 Mb
*** Status: Normal completion

```

The user should also make sure that the variable LEON remains very close to zero after a shock. The value of this variable appears at the end of the listing file (.LST).

```

---- VAR LEON  Excess supply on the last market

      LOWER      LEVEL      UPPER
1      -INF      .      +INF
2      -INF      .      +INF
3      -INF  -3.64E-12  +INF
4      -INF      .      +INF
5      -INF      .      +INF
6      -INF  -4.55E-12  +INF
7      -INF      .      +INF
8      -INF  -9.09E-13  +INF
9      -INF  9.095E-13  +INF
10     -INF  2.728E-12  +INF

```

If it does not, it might be because the user changed some specifications in the model. We strongly suggest that the user first start with the standard PEP-1-t model before introducing any changes. The standard specification should run properly, given that the preceding steps have been followed carefully.

DECALUWÉ, Bernard, André LEMELIN, Véronique ROBICHAUD et Hélène MAISONNAVE (2012), *The PEP Standard Computable General Equilibrium Model: single-country, Recursive Dynamic Version* (Version 2.0), Politique économique et Pauvreté/Poverty and Economic Policy Network, Université Laval, Québec.

MAISONNAVE Hélène, Bernard DECALUWÉ, Véronique ROBICHAUD and André LEMELIN, et (2012), *DEBUGATOR How to debug a Computable General Equilibrium*

*Model using GAMS, AGRODEP and Politique économique et Pauvreté/Poverty and Economic Policy Network, Université Laval, Québec.*