Aggregating Input-Output Table in GAMS

Erwin L. Corong[1]

April 2007

An Input-Output (IO) table shows the linkage among producers and consumers in an economy. Presented in a matrix format, it validates the consistency of national accounts within a simplified framework by providing information on intermediate inputs, gross value added, imports, exports, government expenditure, household consumption and investments. IO tables are not only used for input-output analyses but also serve as a normal starting point towards the construction of social accounting matrices (SAM).

IO tables vary in sizes, and often one needs to aggregate them. However, aggregating an IO table is a painstaking task since one has to reduce corresponding row and columns at a time—i.e., establish a one to one column and row reduction between the IO table's Standard Industrial classification (SIC) code to that of intermediate demand, intermediate input, value added and final demand.

This document demonstrates a way to mechanize the aggregation of IO tables in GAMS (General Algebraic Modeling System software). Although aggregation can be done in any spreadsheet software like Excel, using GAMS allow researchers to save time and minimize errors. The rest of the document is organized as follows. Section 1 lays out the GAMS data exchange (GDX) commands employed to read data from Excel into GAMS. Section 2 demonstrates the use of multi-dimensional sets and illustrates how IO sectoral aggregation can be undertaken in GAMS. Finally, section 3 illustrates data exporting from GAMS to Excel.

1. Importing Input-Output table Data from Excel

Importing IO data from Excel can be undertaken using GAMS data exchange (GDX) facility. When using GDX, it is important that all the relevant information be given to GAMS at the onset, i.e., the file name, the parameter and the Excel spreadsheet range. Moreover, it is essential to note that under the GAMS integrated development environment (IDE), the Excel file containing the data must be saved in the same project directory[2].

Box 1 shows the GAMS command directives employed to read the file IO_table.xls from Excel into GAMS. The command ***$call gdxxrw*** instructs GAMS to use the GDXXRW utility[3]. This is followed by the name of the file ***IO_table.xls*** which contains the data. Then,

[1] Please send comments to erwincorong@gmail.com This note greatly benefited from Jesper Jensen's lecture on aggregating input-output table using GAMS/GDX during his GAMS-MPS/GE training course. The codes used in this exercise is a modified version of the one presented during the training. The usual disclaimer applies.

[2] See the GAMS manual for details about project directory.

[3] See GAMS GDX facilities and tools (2006) for details

the command ***par=iodata*** is declared to inform GAMS that a symbol or parameter definition—named iodata—is being created.

Box 1: Importing the Data

$call gdxxrw IO_table.xls par=iodata rng=b6

Where:

$call gdxxrw	– GAMS command syntax
IO_table.xls	– Name of the Excel file (with .xls extension)
par=iodata	– Name of the declared parameter
rng=b6	– Specified spreadsheet range

Afterward, the Excel spreadsheet range using the command ***rng=b6*** is defined. This is crucial as it informs GAMS the specific Excel worksheet and cells that the user would like to access. In this case, a specific Excel worksheet is not defined and would result in GAMS accessing the first Excel worksheet. Similarly, writing **rng=b5** means that the user is instructing GAMS to read all cells starting from b5, while writing **rng=*b5:v24*** implies accessing cells beginning at b5 and ending at v24. In summary, importing data from Excel by invoking the ***$call gdxxrw.exe*** command means that the user is unloading or passing all the relevant information to a GDX file. Unloading occurs during the compilation stage and is discussed in the next section.

1.2 Compilation Phase

The compilation phase is the step where one specifies the GDX file (in .GDX extension) to be used, the symbols to be read, and the declaration of the elements of a set. This is the phase where the Excel data is unloaded or converted into a GDX file which GAMS uses. Box 2 shows the compilation directives used to unload data from a GDX file into GAMS during compilation[4]. It is important to note that during the compilation phase, only one GDX file can be opened and that any symbol to be used must be declared prior to usage.

Initially, the parameter ***iodata*** is declared. The directive ***$gdxin*** means that the current GDX input file is being closed, while ***IO_table.gdx*** specifies the GDX file that GAMS must read[5]. The ***$load*** command is the directive required to list all symbols in the GDX file, whereas ***iodata*** (found after the $load) is for GAMS to read all symbols belonging to this parameter. One note of caution deserves attention here as the *$load* directive ignores elements that are not in the domain.

[4] See GAMS GDX Facilities and tools document (GAMS 2006) for a thorough list of all directives available during the compilation stage.

[5] Note that GAMS transforms the *IO_table.xls* file into a GDX file *IO_table.gdx* which the user must call.

Box 2: Directives used during the compilation phase

```
Parameter
iodata(*,*)      io data

$gdxin IO_table.gdx
$load iodata
```

2. Set Declaration and Mapping

The user can now proceed to normal GAMS coding beginning with set declaration (box 3).

Box 3: Declaration of sets (IO Accounts)

```
SET
     n          Numeric Labels                  /        1*20        /

     i          Sectors                         /  AGRI          Agriculture, Fishery and Forestry
                                                   MNQG          Mining and Quarrying
                                                   MFNG          Manufacturing
                                                   CONs          Construction
                                                   EGWA          Electricity, Gas and Water
                                                   TRCS          Transport and Communications
                                                   WRTR          Wholsale and Retail Trade
                                                   FINA          Finance
                                                   REAL          Real Estate
                                                   PSER          Private Services
                                                   GSER          Government Services                 /

     fd         Final Demand
                                                /  PCE           Personal Consumption Expenditure
                                                   GGCE          General Government Consumption Expenditure
                                                   GFCF          Gross Fixed Capital Formation
                                                   CHST          Changes in Stocks
                                                   EXPO          Exports
                                                   IMPO          Imports                             /

     va         Value Added
                                                /  COMP          Compensation
                                                   DEPR          Depreciation
                                                   ITXN          Indirect  Taxes-Subsidies
                                                   OPSU          Operating Surplus                   /

  mapi(n,i)     mapping numeric labels to sectors
                                                /  1. AGRI
                                                   2. MNQG
                                                   3. MFNG
                                                   4. CONS
                                                   5. EGWA
                                                   6. TRCS
                                                   7. WRTR
                                                   8. FINA
                                                   9. REAL
                                                   10. PSER
                                                   11. GSER    /

mapva(n,va)     mapping numeric lables to value added
                                                /  13. COMP
                                                   14. DEPR
                                                   15. ITXN
                                                   16. OPSU  /

mapfd(n,fd)     mapping numeric lables to final demands
                                                /  13. PCE
                                                   14. GGCE
                                                   15. GFCF
                                                   16. CHST
                                                   17. EXPO
                                                   18. IMPO   /
```

2.1 Multi-dimensional Sets "Tuples"

Rutherford (2003) show that using "tuples" or ordered sets, which are multi-dimensional sets associated with a numeric index and a text index allow for an efficient way of mapping data into labeled matrices.

The use of multi-dimensional sets was therefore adopted is this exercise. Initially, a set of numeric index ranging from ***1 to 20*** was declared (box 3). This range is not arbitrary and is based on the data as the numeric indices actually correspond to each row and/or column in the table as shown in table 1. One should note that the numeric index must be inserted in each column and row account of table 1 before the data is imported from Excel.

Then the sets I, fd, va (sectors, final demand and value added respectively) were declared. Afterward, the set ***mapi(n,i)*** was declared where, as shown in box 1, each sector was mapped with a numeric index. Finally, the sets ***mapva(n,va), mapfd(n,fd)*** were declared to map the columns which contains the value added and final demand data.

2.2 Data handling

The Input-Output data was obtained using the procedure shown in box 4. Then, a column sum and row sum check using the parameters colchk and colrow was undertaken to verify whether the IO table is balanced. Colchk is the sum of total inputs to production (intermediate and primary) less total output, while the colrow is supply less demand.

Box 4: Declaration

```
Parameter
iod        input-output demand
vad        value added demand
fin        final demand
imptx      import taxes;
colchk     check the column sums in the IO table
colrow     check the supply-demand balance in the IO table;

iod(i,j) = sum((n,m)$(mapi(n,i)   and mapi(m,j)), iodata(n,m));
vad(va,i)= sum((n,m)$(mapva(n,va) and mapi(m,i)), iodata(n,m));
fin(i,fd)= sum((n,m)$(mapi(n,i)   and mapfd(m,fd)),iodata(n,m));
imptx(va,fd) = sum((n,m)$(mapva(n,va) and mapfd(m,fd)),iodata(n,m));

colchk(i)  =  sum(j,iod(j,i)) + sum(va,vad(va,i)) -   sum(mapi(n,i),iodata("18",n));
colrow(i)  =  sum(j,iod(i,j)) + sum(fd,fin(i,fd))   - (sum(j,iod(j,i)) + sum(va,vad(va,i)));
```

Table 1: Hypothetical Input-Output table

DESCRIPTION		AFF	M&Q	MFG	CONS	EGW	TCS	W&RT	FIN	REAL	PSER	GSER	Total Intermediate Demand	Personal Consumption Expenditure	General Government Consumption Expenditure	Gross Fixed Capital Formation	Changes in Stocks	Exports	Imports	Total Final Demand	Total Output
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Agriculture, Fishery and Forestry (AFF)	1	25,235	22	182,648	7	0	128	7,084	-	-	8,564	1,655	225,343.9	97,652.1	-	14,219.0	8,609.9	14,721.0	(17,173.5)	118,028.6	343,372
Mining and Quarrying (M&G)	2	-	199	81,407	3,777	6,642	11	2,504	-	69	76	8	94,693.6	1,299.0	-	-	261.7	2,844.4	(80,204.9)	(75,799.8)	18,894
Manufacturing (MFG)	3	44,178	3,147	536,863	40,712	10,569	85,380	56,030	10,552	3,410	84,680	23,489	899,009.8	460,264.4	-	133,595.7	24,163.2	787,715.3	(655,178.8)	750,559.8	1,649,570
Construction (CONS)	4	495	242	541	870	124	279	76	516	1,255	-	5,538	9,936.9	-	-	135,339.8	-	2,143.5	(2,740.1)	134,743.1	144,680
Electricity, Gas and Water (EGW)	5	3,240	1,031	34,238	348	7,498	1,120	1,918	2,250	217	9,440	2,717	64,017.3	34,334.7	-	-	-	-	-	34,334.7	98,352
Transport and Communications (TCS)	6	1,411	256	12,238	13,203	1,013	6,489	45,381	8,772	476	4,749	4,624	98,612.8	194,654.9	-	10,364.4	-	25,980.1	(72,567.0)	158,432.5	257,045
Wholsale and Retail Trade (W&RT)	7	2,703	134	131,480	1,777	1,953	2,594	527	187	157	3,356	1,018	145,885.2	231,586.5	-	9,895.3	-	20,115.6	(5,723.3)	255,874.1	401,759
Finance (FIN)	8	2,859	332	11,439	2,340	182	8,098	13,600	28	4,585	7,561	6,693	57,717.5	103,285.0	-	-	-	3,226.2	(13,943.6)	92,567.6	150,285
Real Estate (REAL)	9	155	28	1,924	985	2	2,338	2,438	4,603	121	2,837	1,897	17,327.6	128,052.6	-	-	-	-	-	128,052.6	145,380
Private Services (PSER)	10	5,192	1,431	11,372	2,991	2,211	11,279	6,298	24,835	4,980	32,759	13,527	116,875.9	205,697.4	-	-	-	55,499.5	(48,106.6)	213,090.2	329,966
Government Services (GSER)	11	-	-	-	-	-	-	-	-	-	-	-	-	1,019.9	219,562.4	-	-	817.6	(397.8)	221,002.2	221,002
Total Intermediate Inputs	12	85,468	6,822	1,004,149	67,009	30,196	117,717	135,856	51,742	15,270	154,022	61,168	1,729,421	1,457,847	219,562	303,414	33,035	913,063	(896,036)	2,030,886	3,760,306.262
Compensation	13	90,802	2,519	162,733	35,183	11,911	33,904	70,496	20,996	3,624	85,044	151,892	669,105	-	-	-	-	-	-	-	669,105
Depreciation	14	21,503	800	35,409	10,428	15,359	47,590	24,649	5,632	2,671	24,635	7,334	196,009	-	-	-	-	-	-	-	196,009
Indirect Taxes Net of Subsidies	15	8,121	639	49,748	1,969	3,079	3,518	4,318	8,271	1,885	7,891	608	90,046	-	-	-	-	-	47,623	47,623	137,669
Operating Surplus	16	137,479	8,114	397,530	30,091	37,807	54,315	166,441	63,644	121,931	58,373	-	1,075,725	-	-	-	-	-	-	-	1,075,725
Total Primary Inputs	17	257,905	12,072	645,420	77,671	68,156	139,328	265,904	98,543	130,110	175,944	159,834	2,030,886	-	-	-	-	-	47,623	47,623	2,078,508
Total Inputs	18	343,372	18,894	1,649,570	144,680	98,352	257,045	401,759	150,285	145,380	329,966	221,002	3,760,306.262	1,457,847	219,562	303,414	33,035	913,063	(848,413)	2,078,508	-

2.3 Aggregation Mapping

Aggregation mapping involves matching the sectoral set i defined in box 3, with a new set of sectoral classification (***set g***) in order to effectively carry out the desired aggregation process. As shown in box 5, the set **g** containing the 4 major sectors in the economy was declared followed by the set ***Map***(*,*) which matches each set that belongs to I to that of g. Essentially, this means aggregating all the agricultural, manufacturing, private services and government services sub-sectors in I into the four major sectors in the economy (i.e, AGR, MFG, PSE and GSE).

Box 5: Declaration of sets (Aggregation Maps)

```
SET
  G        Sectoral Classifications          /  AGR        Agriculture
                                                MFG        Manufacturing
                                                PSE        Private Services
                                                GSE        Government or Public Services
                                                                                    /

map(*,*)   mapping individual industry to sectoral classifications
                                             /  AGRI.AGR
                                                MNQG.MFG
                                                MFNG.MFG
                                                CONS.PSE
                                                EGWA.PSE
                                                TRCS.PSE
                                                WRTR.PSE
                                                FINA.PSE
                                                REAL.PSE
                                                PSER.PSE
                                                GSER.GSE   /
```

2.4 Data Aggregation

The IO table was then aggregated into 4 major sectors as shown in box 6.

Box 6: Computing for Weighted Tariff Rates

```
Parameter
iodat          aggregated inter-industry matrix
vadat           aggregated value added
finde          aggregated final demand
imptax          aggregated import tax-tariff
;

iodat(g,gg)      = sum((i,j)$(map(i,g) and map(j,gg)), iod(i,j));
vadat(va,g)      = sum(i$map(i,g),vad(va,i));
finde(g,fd)      = sum(i$map(i,g),fin(i,fd));
imptax(va,fd)     = imptx(va,fd)          ;
```

2.5 Verifying aggregated IO table

It is essential to verify the new IO table after aggregation. One can check whether there exist a negative value or empty rows and columns in the data (Rutherford 2003). This was undertaken in box 7.

Box 7: Data Check

```
set negval(*,*)  Check for negative balance
set empty(g,*)   Check for empty rows and columns;

negval(g,gg)        = yes$(iod(g,gg)<0);
empty(g,"row")      = 1$(sum(gg,iod(g,gg))=0);
empty(gg,"column")  = 1$(sum(g,iod(g,gg))=0);
```

3. Writing (Exporting) Data to an Excel Spreadsheet

The resulting computations were then exported from GAMS to Excel (box 8). Initially, the ***execute_unload*** statement was written to denote (replace) a new (or existing) GDX file[6]. This statement allows the user to unload the results to a GDX file—which occurs during the execution phase. This statement is followed by ***io_table.gdx'*** which is the file name of the GDX file, then followed by the list of parameters being unloaded

Box 8: Exporting the Data

```
execute_unload 'io_table.gdx', iodat vadat finde imptax
execute 'gdxxrw.exe io_table.gdx o=aggregated_IO.xls par=iodat rng=a1
                    par=vadat rng=a6 par=finde rng=H1 par=imptax rng=a12'
```

The resulting computations were then written from GDX to Excel. To begin with, the directive ***execute*** was used, followed by ***'gdxxrw.exe io_table.gdx o=aggregated_IO.xls par=iodat rng=a1 par=vadat rng=a6 par=finde rng=h1par=imptax rng=a12 '*** where *io_table.gdx* is the GDX file used, with the corresponding output file *io_table.xls* written in Excel (.xls) format. The *par=iodat* was then declared to instruct GAMS/GDX to write all data belonging to iodat in Excel. Then, the *rng=a1* was declared to instruct GDX to write the iodat results starting at range a1[7]. The same principle was applied to all other remaining parameters.

[6] The execute_unload directive does not add or replace symbols in an existing GDX file (GAMS, 2006)
[7] It is also possible to choose the Excel worksheet that the user wants the results to be written. See GAMS (2006) for details.

References

Corong, Erwin (2007). "*Input_Output_1.gms:* The accompanying GAMS code for the note— Aggregating Input-Output Table in GAMS". Poverty and Economic Policy (PEP) Research Network.
Available at www.pep-net.org

GAMS Development Corporation (2006). *"GAMS GDX facilities and tools"*. Washington, D.C., USA.
GAMS website: www.gams.com

Jensen, Jesper (2005). *"Lecture on: Aggregating input-output tables using GDX"*. Introduction to CGE modeling using GAMS and MPS/GE. Essen, Germany
Jensen Jesper's website: www.tecatraining.dk

Rutherford, Thomas (2003). *"A GAMS program which maps social accounting data into labeled submatrices"*. University of Colorado, Boulder Colorado, USA.
Available at www.mpsge.org