

Appendix 5

How to Stop Worrying and Love the Computer: Program generating Figures 5.1 and 5.2 using MPS/GE

This appendix gives the code for the central-case model of Chapter 5. This is the “four-firm-type” model that was joint work with Tony Venables, published in the JIE 1998. The calibration of this model is somewhat different from the original, so don’t puzzle over it if you notice that the results do not quite match those in the JIE article. They do match the results in Chapter 5 exactly. I am going to assume in writing this appendix that you read appendix 3 discussing the basic syntax and structure of MPS/GE.

National (type-d) and horizontal multinationals (type-h) are permitted, but vertical firms (type-v) are not. Thus there are four possible firm types: d_i , d_j , h_i , and h_j . A second simplifying assumption is that the X sector uses only one factor, skilled labor. The other factor, referred to as the “composite factor” or “unskilled labor” is used only in the Y sector. This helps “convexify” the model,

in that the X sector must draw skilled labor from the Y sector at an increasing cost in terms of Y.

The initial calibration of the model is to the center of the world Edgeworth box, where the solution is symmetric and all firms are type-h. The top of the file gives the accounting matrix of Table 5.1, except that I have added together a couple of columns here exploiting symmetry at the point of initial calibration. Because a type- h_i and type- h_j firm have the same technologies and output in both countries, I need only specify one multinational agent, ENTM, but two technologies for producing fixed costs. The equilibrium activity levels for these technologies corresponds to the number of type- h_i and type- h_j firms active in equilibrium.

Here are definitions of sectors (activities), commodities (prices), and consumers (agents), and auxiliary variables (markups).

<u>Sector</u>	<u>Activity level gives:</u>
YI	Output of Y in country i
YJ	Output of Y in country j
WI	Welfare of country i (output of utility)
WJ	Welfare of country j (output of utility)
XMI	Output of X by multinational firms in country i
XMJ	Output of X by multinational firms in country j
XDI	Output of X by national firms headquartered in i
XDII	Supply of a type- d_i firm to market i
XDIJ	Supply of a type- d_i firm to market j
XDJ	Output of X by national firms headquartered in j
XDJI	Supply of a type- d_j firm to market i
XDJJ	Supply of a type- d_j firm to market j
NMI	Output of fixed costs for type- h_i firms
NMJ	Output of fixed costs for type- h_j firms
NI	Output of fixed costs for type- d_i firms
NJ	Output of fixed costs for type- d_j firms

<u>Commodity</u>	<u>Variable gives:</u>
CY	Price of good Y
UTILI	Price of a unit of utility in country i
UTILJ	Price of a unit of utility in country j
SI	Price of skilled labor in country i
SJ	Price of skilled labor in country j
LI	Price of unskilled labor in country i
LJ	Price of unskilled labor in country j
CXI	Consumer price of X in country i
CXJ	Consumer price of X in country j
CXDI	Marginal cost of X in country i
CXDJ	Marginal cost of X in country j
FCM	Price of fixed costs for multinational firms
FCI	Price of fixed costs for national firms in country i
FCJ	Price of fixed costs for national firms in country j

Consumer Variables gives:

CONSI	Income of representative consumer in country i
CONSJ	Income of representative consumer in country j
ENTM	Income (markup revenue) of multinational “owner”
ENTI	Income (markup revenue) of owners of type- d_i firms
ENTJ	Income (markup revenue) of owners of type- d_j firms

Auxiliary Variable gives:

NMIT Number of type- h_i firms (explained below)

NMJT Number of type- h_j firms (explained below)

NIT Number of type- d_i firms (explained below)

NJT Number of type- d_j firms (explained below)

MARKMI Markup of type- h_i and h_j firms in country i

MARKMJ Markup of type- h_i and h_j firms in country j

MARKDII Markup of a type- d_i firm in market i

MARKDIJ Markup of a type- d_i firm in market j

MARKDJI Markup of a type- d_j firm in market i

MARKDJJ Markup of a type- d_j firm in market j

I will explain the variables NMIT, NMJT, NIT, and NJT at the end of this appendix for anyone interested. They are basically there to avoid a potential zero/zero division problem in the constraint equations which is disliked by the solver. They take on the values of NMI, NMJ, NI, and NJ except when one of the latter is zero, in which case the auxiliary variables are bounded slightly above zero to avoid a 0/0 expression.

Near the top of the GAMS files, we see

```
SET C /1*19/;
```

```
SET R /1*19/;
```

```
ALIAS (R,RR);
```

```
SCALAR UP /0/;
```

The sets C and R stand for column and row, and we are using these to generate the Edgeworth box, moving in 5% steps, from 0.05 to 0.95, which is where the number 19 comes from (19 steps). I will not go into detail about the alias statement and the scalar “UP” (set initially at $UP = 0$), but will mention their use below.

After the MPS/GE block of the problem, we set starting values of the variables with the .L (for level) syntax. These help the solver find the first solution. Then we come to the loop statement that repeatedly solves the model $19 \times 19 = 361$ times to generate the Edgeworth box. I will explain this complicated loop statement at the end of the appendix; basically, it causes the program to “snake” through the Edgeworth box rather than jumping from the bottom of one column to the top of the next column. We could replace it with `LOOP (C,`

```
LOOP (R,
```

```
) ;
```

```
) ;
```

but the solver has trouble with the big jumps that result.

Inside the loop we see four factor endowment parameters (ENDOW) being set for each point in the Edgeworth box::

```
ROW = ORD (R) ;
```

```
COL = ORD (C) ;
```

```
ENDOWJS = (ROW) * .1 ;
```

```
ENDOWJL = (2 - .1*COL) ;
```

```
ENDOWIS = (2 - .1*ROW) ;
```

```
ENDOWIL = (COL) * .1 ;
```

where “ORD” stands for the ordinal value of the set and converts the set index into an actual number. The ENDOW parameters are seen in the consumer demand blocks

\$DEMAND:CONSI

D:UTILI

E:SI Q: (150.*ENDOWIS)

E:LI Q: (50.*ENDOWIL)

\$DEMAND:CONSJ

D:UTILJ

E:SJ Q: (150.*ENDOWJS)

E:LJ Q: (50.*ENDOWJL)

Note that when $R = C = 1$, the first value of the loop for example, the endowments of the two countries will be

$$SI = 150 * 1.90 = 285 \quad (95\% \text{ of the world endowment of S})$$

$$LI = 50 * 0.1 = 5 \quad (5\% \text{ of the world endowment of L})$$

$$SJ = 150 * 0.1 = 15 \quad (5\% \text{ of the world endowment of S})$$

$$LJ = 50 * 1.90 = 95 \quad (95\% \text{ of the world endowment of L})$$

This is therefore the top left-hand (northwest) corner of the Edgeworth box.

Please ignore the “.L” statements that follow, they are to avoid a certain divide-by-zero problem that can arise when an agent has zero income. This will be discussed briefly at the end of the program.

Following the SOLVE statement, we manipulate the output before going on to the next element of the box. We see the statements

$$\text{MNIE}(R, C) \$ (\text{NMI.L GE } 0.3) = 10.;$$

$$\text{MNJE}(R, C) \$ (\text{NMJ.L GE } 0.3) = 1.0;$$

$$\text{NEI}(R, C) \$ (\text{NI.L GE } 0.3) = .1;$$

$$\text{NEJ}(R, C) \$ (\text{NJ.L GE } 0.3) = .01;$$

$$\text{REGIME}(R, C) = \text{MNIE}(R, C) + \text{MNJE}(R, C) + \text{NEI}(R, C) + \text{NEJ}(R, C);$$

These statements generate an “indicator function” that tells us what types of firms are active in equilibrium. (Please ignore the \$ conditionals, they are also discussed at the end of the appendix.) There are various ways to do this, but I just assign a 10 if type- h_i firms are active, and so forth. The parameter matrix REGIME(R,C) is then the sum of these assignments. A REGIME value of 10.1,

for example, tells us that type- h_i and type- d_i firms are active in equilibrium at that point in the world Edgeworth box. The values of REGIME in the central case of Chapter 5 are shown in Table 5.2.

Finally, we calculate the volume of affiliate sales (VAS), shown in Figure 5.2.

$$MI(R, C) = NMI.L\$ (NMI.L GE 0.3) ;$$

$$MJ(R, C) = NMJ.L\$ (NMJ.L GE 0.3) ;$$

$$DI(R, C) = NI.L\$ (NI.L GE 0.3) ;$$

$$DJ(R, C) = NJ.L\$ (NJ.L GE 0.3) ;$$

$$SHAREJM(R, C) \$ (MI(R, C) + MJ(R, C)) = MJ(R, C) / (MI(R, C) + MJ(R, C)) ;$$

$$SHAREIM(R, C) \$ (MI(R, C) + MJ(R, C)) = MI(R, C) / (MI(R, C) + MJ(R, C)) ;$$

$$VAS(R, C) = (CXI.L * XMI.L * SHAREJM(R, C) + CXJ.L * XMJ.L * SHAREIM(R, C)) / CY.L ;$$

SHAREJM is the share of multinational firms (if any) headquartered in country j . Of the total output of multinationals in country i , only the portion produced by type- h_j firms counts as affiliate sales, as discussed in Chapter 5. Thus, out of the value of output of type- h firms in country i , only

$$CXI . L * XMI . L * SHAREJM (R, C)$$

counts as affiliate output (CXI is price, XMI is the quantity (activity level), and CY is the price of good Y, used here as numeraire). At the center of the box in Figures 5.1 and 5.2, the share parameters are both $1/2$, so exactly half of all world output of X is affiliate output.

Near the end of the program you will see the statement

$$MODELSTAT (R, C) = BOX . MODELSTAT - 1 ;$$

Filename (BOX in the this case) follow by a period and the keyword MODELSTAT produces a statistic that indicates if the model solved properly. It takes a value of 5 if the model does not solve and a value of one if it does solve (don't ask me why). By subtracting one and storing all values in a matrix, I will

have a matrix where the only non-zero values indicate a failure to solve that element. GAMS will only display non-zero values of matrices and parameters, which is very beneficial in this case. If a solution is found every time, it will just report “ALL ZERO” instead of 361 zeros.

That is about it. I realize that this is too much for some of you, and not enough for others. As I have said before in these appendices, my two goals here are (1) to give you a feel for how the numerical models actually work, and (2) to give you enough understanding to make a decision as to whether or not you might be interested in learning more about these techniques.

Here is the actual program that generates Table 5.2, Figure 5.1 and Figure 5.2 using the initial calibration of Table 5.1. Note that the numbers in the accounting matrix are values. In many cases the initial price is chosen as one, so they are also physical quantities. The exceptions are X_i and X_j . Note that the 100 units in the initial data are interpreted as 80 physical units at a price of 1.25 (marginal cost is 1, and the markup is 0.2: $1.25(1 - 0.2) = 1$). This is where the “80s” come from in the MPS/GE program. Note that the price 1.25 is used in calibrating the utility functions, activities WI and WJ.

- * EXAMPLE 5
- * this produces Figure 5.1 of Chapter 5
- * it uses mpsge

- * four firm types, no type-v firms
- * Oligopoly model of Markusen and Venables, JIE 1998
- * calibrated to the center of the edgeworth box

\$ONTEXT

	YI	YJ	XMI	XMJ	NMI	NMJ	WI	WJ	CONSI	CONSJ	EHTMH
CYI	100						-100				
CYJ		100						-100			
CXI			100				-100				
CXJ				100				-100			
FCM					20	20					-40
LI	-50								50		
SI	-50		-80		-15	-5			150		
LJ		-50								50	
SJ		-50		-80	-5	-15				150	
UTILI							200		-200		
UTILJ								200		-200	
MKI			-10	-10							20
MKJ			-10	-10							20

\$OFFTEXT

SET C /1*19/;
 SET R /1*19/;

ALIAS (R,RR) ;
SCALAR UP /0/ ;

PARAMETERS

FMI

FMJ

ENDOWIS

ENDOWJS

ENDOWIL

ENDOWJL

FDI

FDJ

TCOST

ROW

COL

SCALE (R)

TMC (R, C)

MNIE (R, C)

MNJE (R, C)

NEI (R, C)
NEJ (R, C)
MI (R, C)
MJ (R, C)
DI (R, C)
DJ (R, C)
VAS (R, C)
REGIME (R, C)
SHAREIM (R, C)
SHAREJM (R, C)
WELFAREI (R, C)
WELFAREJ (R, C)
MODELSTAT (R, C) ;

FMI = 8 ;
FMJ = 8 ;
FDI = 5.5 ;
FDJ = 5.5 ;
TCOST = .15 ;

**** now control passes to the MPS/GE compiler ****

\$ONTEXT

\$MODEL:BOX

\$SECTORS:

YI YJ

WI WJ

XMI XMJ

XDI

XDII

XDIJ

XDJ

XDJI

XDJJ

NMI NMJ

NI NJ

\$COMMODITIES:

CY

UTILI UTILJ

SI SJ

LI LJ

CXI CXJ

CXDI

CXDJ

FCM

FCI FCJ

\$CONSUMERS:

CONSI CONSJ

ENTM

ENTI ENTJ

\$AUXILIARY:

NMIT

NMJT

NIT

NJT

MARKMI

MARKMJ

MARKDII

MARKDIJ

MARKDJI

MARKDJJ

\$PROD:YI s:5.0

O:CY Q:100.0

I:SI Q:50.0

I:LI Q:50.0

\$PROD:YJ s:5.0

O:CY Q:100.0

I:SJ Q:50.0

I:LJ Q:50.0

\$PROD:XMI

O: CXI Q: 80. A: ENTM N: MARKMI
I: SI Q: 80.

\$PROD:XMJ

O: CXJ Q: 80. A: ENTM N: MARKMJ
I: SJ Q: 80.

\$PROD:XDI

O: CXDI Q: 80.
I: SI Q: 80.

\$PROD:XDII

O: CXI Q: 80. A: ENTI N: MARKDII
I: CXDI Q: 80.

\$PROD:XDIJ s:0.0

O: CXJ Q: 80. A: ENTI N: MARKDIJ

I: CXDI Q: 80.

I: SI Q: (80.*TCOST)

\$PROD:XDJ

O: CXDJ Q: 80.

I: SJ Q: 80.

\$PROD:XDJI

O: CXI Q: 80. A: ENTJ N: MARKDJI

I: CXDJ Q: 80.

I: SJ Q: (80.*TCOST)

\$PROD:XDJJ s:0.0

O: CXJ Q: 80. A: ENTJ N: MARKDJJ

I: CXDJ Q: 80.

\$PROD:NMI s:0.0

O:FCM

I:SI Q:(FMI*3/4)

I:SJ Q:(FMI/4)

\$PROD:NMJ s:0.0

O:FCM

I:SJ Q:(FMJ*3/4)

I:SI Q:(FMJ/4)

\$PROD:NI

O:FCI

I:SI Q:FDI

\$PROD:NJ

O:FCJ

I:SJ Q:FDJ

\$PROD:WI s:1.0

O:UTILI Q:200.

I: CXI Q: 80. P: 1.25

I: CY Q: 100.

\$PROD: WJ s: 1.0

O: UTILJ Q: 200.

I: CXJ Q: 80. P: 1.25

I: CY Q: 100.

\$DEMAND:CONSI

D:UTILI

E:SI Q:(150.*ENDOWIS)

E:LI Q:(50.*ENDOWIL)

\$DEMAND:CONSJ

D:UTILJ

E:SJ Q:(150.*ENDOWJS)

E:LJ Q:(50.*ENDOWJL)

\$DEMAND:ENTM

D:FCM

\$DEMAND:ENTI

D:FCI

\$DEMAND:ENTJ

D:FCJ

\$CONSTRAINT:NMIT

NMIT =G= NMI;

\$CONSTRAINT:NMJT

NMJT =G= NMJ;

\$CONSTRAINT:NIT

NIT =G= NI;

\$CONSTRAINT:NJT

NJT =G= NJ;

\$CONSTRAINT:MARKMI

MARKMI * (NMIT+NMJT) * (XMI + XDII + XDJI) =G= XMI;

\$CONSTRAINT:MARKMJ

MARKMJ * (NMIT+NMJT) * (XMJ + XDIJ + XDJJ) =G= XMJ;

\$CONSTRAINT:MARKDII

MARKDII*(NIT)*(XMI + XDII + XDJI) =G= XDII;

\$CONSTRAINT:MARKDIJ

MARKDIJ*(NIT)*(XMJ + XDIJ + XDJJ) =G= XDIJ;

\$CONSTRAINT:MARKDJI

MARKDJI*(NJT)*(XMI + XDII + XDJI) =G= XDJI;

\$CONSTRAINT:MARKDJJ

MARKDJJ*(NJT)*(XMJ + XDIJ + XDJJ) =G= XDJJ;

**** now control goes back to GAMS ****

\$OFFTEXT

\$SYSINCLUDE MPSGESET BOX

**** set starting values (these are guesses) ****

NMI.L = 0;
NMJ.L = 0;
NMIT.L = 0;
NMJT.L = 0;

NI.L = 2.0;
NJ.L = 2.0;
NIT.L = 2.0;
NJT.L = 2.0;

CXI.L = 1.25;
CXJ.L = 1.25;

MARKMI.L = .2;
MARKMJ.L = .2;

MARKDII.L = .2;
MARKDIJ.L = .2;
MARKDJI.L = .2;

MARKDJJ.L = .2;

XMI.L = .0;

XMJ.L = 0.;

XDI.L = 1.0;

XDII.L = 1.0;

XDIJ.L = 1.0;

XDJ.L = 1.0;

XDJI.L = 1.0;

XDJJ.L = 1.0;

CY.FX = 1.0;

NMIT.LO = 0.001;

NMJT.LO = 0.001;

NIT.LO = 0.001;

NJT.LO = 0.001;

**** now do some stuff, mostly to reduce the size of
the listing file

```
BOX.ITERLIM = 5000;  
OPTION MCP=MILES;  
OPTION SOLPRINT=OFF;  
OPTION LIMROW=0;  
OPTION LIMCOL=0;  
$OFFSYMLIST OFFSYMREF OFFUELLIST OFFUELXREF
```

**** now comes the loops that repeatedly solve the
model over the

**** world Edgeworth box ****

```
LOOP(C,
```

```
    LOOP(RR,
```

* If we are going down a column, then $R=RR$.

* If we are going up a column, then $R = NR - RR + 1$

```

      LOOP (R$( (1-UP)$ (ORD(R) EQ ORD(RR)           ) +
                UP$(ORD(R) EQ CARD(R)-ORD(RR)+1)   ) ,

```

```

ROW = ORD(R) ;

```

```

COL = ORD(C) ;

```

```

ENDOWJS = (ROW) * .1 ;

```

```

ENDOWJL = (2 - .1*COL) ;

```

```

ENDOWIS = (2 - .1*ROW) ;

```

```

ENDOWIL = (COL) * .1 ;

```

```

UTILI.L = MAX(UTILI.L, 1.E-4) ;

```

```

UTILJ.L = MAX(UTILJ.L, 1.E-4) ;

```

```

FCM.L = MAX(FCM.L, 1.E-4) ;

```


FCI.L = MAX (FCI.L, 1.E-4) ;

FCJ.L = MAX (FCJ.L, 1.E-4) ;

\$INCLUDE BOX.GEN

SOLVE BOX USING MCP ;

MODELSTAT (R,C) = BOX.MODELSTAT - 1. ;

MNIE (R,C) \$ (NMI.L GE 0.3) = 10. ;

MNJE (R,C) \$ (NMJ.L GE 0.3) = 1.0 ;

NEI (R,C) \$ (NI.L GE 0.3) = .1 ;

NEJ (R,C) \$ (NJ.L GE 0.3) = .01 ;

REGIME (R,C) = MNIE (R,C) + MNJE (R,C) + NEI (R,C) +
NEJ (R,C) ;

MI (R,C) = NMI.L \$ (NMI.L GE 0.3) ;

MJ (R,C) = NMJ.L \$ (NMJ.L GE 0.3) ;

$$DI(R, C) = NI.L \$(NI.L \text{ GE } 0.3);$$

$$DJ(R, C) = NJ.L \$(NJ.L \text{ GE } 0.3);$$

$$SHAREJM(R, C) \$(MI(R, C) + MJ(R, C)) = MJ(R, C) / (MI(R, C) + MJ(R, C));$$

$$SHAREIM(R, C) \$(MI(R, C) + MJ(R, C)) = MI(R, C) / (MI(R, C) + MJ(R, C));$$

$$VAS(R, C) = (CXI.L * XMI.L * SHAREJM(R, C) + CXJ.L * XMJ.L * SHAREIM(R, C)) / CY.L;$$

$$WELFAREI(R, C) = WI.L;$$

$$WELFAREJ(R, C) = WJ.L;$$

);

);

* We have finished a column, so the next column
 * changes direction:

```
IF (UP, UP = 0; ELSE UP = 1; );  
  
);  
  
DISPLAY WELFAREI, WELFAREJ;  
DISPLAY MI, MJ, DI, DJ;  
DISPLAY VAS;  
  
DISPLAY REGIME;  
DISPLAY MODELSTAT;  
  
$LIBINCLUDE SSDUMP REGIME FIG1REG.XLS  
$LIBINCLUDE SSDUMP VAS FIG2VAS.XLS
```

Final notes for the truly interested and strong-willed:

The variables NMIT, NMJT, NIT, and NJT need a few words of explanation but you might want to just skip this paragraph if you are not interested in the fine points. The difficulty with the markup equations is that the sales of a given firm type in a market are total output of that group of firms divided by the number of those firms. That is, the X variables above are total output of a firm type, not the output of an individual firm. If that firm type is not active, then this is a ratio of zero/zero. The computer doesn't like this much, and the solver can crash when asked to evaluate 0/0. So I have a second set of variables for the number of firms, those ending in "T", which are set in constraint equations to be greater than or equal to the actual number of firms, but have minimum values constrained at 0.001 to avoid the division-by-zero problem. So in the MPS/GE block you will see, for example,

```
$CONSTRAINT :NIT
  NIT =G= NI;
```

where NI is the activity level for production of fixed costs for type- d_i firms. The auxiliary variables NIT appears in the markup equation, such as

```
$CONSTRAINT:MARKDII
  MARKDII*(NIT)*(XMI + XDII + XDJI) =G= XDII;
```

and NIT is constrained with a minimum value by a later statement,

```
NIT.LO = 0.001;
```

where “LO” stands for “lower bound on the variable named”. Thus when firm type- d_i is not active in equilibrium, $NI = 0$, $XDII/NIT = 0/0.001$ instead of $0/0$.

Now a few comments on the Loop statements. These are given by

```

LOOP (C,
    LOOP (RR,
        LOOP (R$ ( (1-UP) $ (ORD (R) EQ ORD (RR) ) +
                    UP$ (ORD (R) EQ CARD (R) - ORD (RR) + 1) ) ,
            ) ;
        ) ;
        IF (UP, UP = 0 ; ELSE UP = 1 ; ) ;
    ) ;

```

I will not go into this in detail. Basically, if I just said

```

LOOP (C,
LOOP (R,

```

```
) ;  
) :
```

the program would go down the first column and jump back to the top of the second column and go down that column. The problem with that is that the solver will begin looking for the solution at the top of the second column using the solution values at the bottom of the first column as starting values. But these starting values are way off from the new solution values. What the complicated loop statement above does is to “snake” out way through the Edgeworth box. When we reach the bottom of the first column, it just moves over to the bottom element of the second column and then works its way up. This avoids the big “jumps” that sometime cause the solver to fail.

Following the specification of endowments, we see the statements:

```
UTILI.L = MAX(UTILI.L, 1.E-4) ;  
UTILJ.L = MAX(UTILJ.L, 1.E-4) ;  
FCM.L = MAX(FCM.L, 1.E-4) ;
```

$$\begin{aligned} \text{FCI.L} &= \text{MAX}(\text{FCI.L}, 1.E-4); \\ \text{FCJ.L} &= \text{MAX}(\text{FCJ.L}, 1.E-4); \end{aligned}$$

These avoid another possible division-by-zero problem. When there is a good, such as fixed costs, which is only demanded by a single agent and that agent has zero income (e.g., an inactive firm type), the price of that good will be zero. (Note that the left-hand-side variables are all prices here.) But this price can appear in a denominator of some equation in the solution algorithm, so an initial assignment of a minimum starting value for each of these goods avoids a divide-by-zero problem.

One unexpected consequence of the use of the auxiliary variables such as NIT discussed above, it that the solver can leave little pieces of firms around. With reference to the determination of the equilibrium regimes (REGIME), the conditionals such as \$(NI.L GE 0.3) says don't make an assignment unless there is at least 0.3 of a firm in equilibrium. This is a somewhat sloppy way of ignoring a firm type when there is only a small fraction of one firm in equilibrium.

Appendix 4

Fun with Sets and Conditionals: GAMS program generating the Nash Equilibria in Figures 3.6 and 3.7

GAMS has a number of features that make high-dimension problems or looping over parameter values quite easy. The set notation in GAMS is clear and simple in my opinion. I have also made extensive use of conditional statements that set values of parameters or variables depending on whether or not some condition is true or false.

This appendix illustrates the use of sets and conditionals, using the policy experiment in Figures 3.6 and 3.7 as an example. Set notation can also be used in MPS/GE, but I won't go into that here. I can tell you that it is immensely useful in large-dimension problems. Again, there are many complications, subtleties, and options and I cannot provide a general tutorial here. As in the case of the other appendices, I want to illustrate the use of GAMS and to allow you to decide if it is something you might like to pursue further.

The program that generates the numbers for Figures 3.6 and 3.7 begins with the following statement.

```
SETS
R /0*2/
C /0*2/
S /1*10/ ;

ALIAS (R, RR) ;
ALIAS (C, CC) ;
```

We are going to use three sets, R (for row), C (for column), and S (for scenario, namely different trade costs). There are many ways to order and specify the elements of a set, using numbers and/or letters, and I just use one simple one here. /0*2/ means that the set has three elements (0, 1, 2). (0, 1, 2) are the three strategies each firm can adopt: zero plants, one plant, or two plants. The rows and columns comprise a 3x3 payoff matrix as in (29) of Chapter 3. Set S has 10 elements, (1,...,10) which will be different levels of tc_j in this example.

“Alias” allows a set to have a second name and we will see why this is useful shortly.

The parameters of this model are:

$PI(R,C,S)$	profits of firm i when i chooses strategy R , firm j chooses strategy C , and the trade cost is S .
$PJ(R,C,S)$	profits of firm j
$WI(R,C,S)$	welfare of country i
$WJ(R,C,S)$	welfare of country j
$ROWMAX(R,C,S)$	largest element of row R , for trade cost S
$COLMAX(R,C,S)$	largest element of column C , for cost S
$NE(R,C,S)$	nash equilibrium: an element of matrix $R \times C$ will be assigned a value of 1 if it is a Nash equilibrium, zero otherwise.
$PINE(S)$	Nash equilibrium profits of firm i for trade costs S
$PJNE(S)$	Nash equilibrium profits of firm j
$WINE(S)$	Nash equilibrium welfare of country i
$WJNE(S)$	Nash equilibrium welfare of country j
$ALPHA$	demand intercept, see chapter 3

G	plant fixed cost
F	firm fixed cost
T	cost of exporting to country i
TJ	cost of exporting to country j
TCJ(S)	vector of different values of TJ

Now the program loops over set S. Each iteration of the loop increases the trade cost TJ by 0.25, beginning with $TJ = 1$. ORD(S) is the ordinal value of the current element of S, and allows the element to be written as a number (i.e., element 1 of the set S becomes the number 1).

```

LOOP (S,
TJ = 0.75 + 0.25*ORD(S);
TCJ(S) = TJ;

```

TCJ(S) just stores the values of TJ as a vector for display later in the listing file. Then the program sets the values of profits for each element of the 3x3 payoff matrix, for each level of the trade cost TJ. These values are found in the text of Chapter 3, equations (23)-(28). The loop is close with the syntax “);”

Next comes three lines of code that find all the pure-strategy Nash equilibria for each value of S . These use the conditional operator “\$”. Consider a couple of simple examples

$$X = 1 \$ Y ; \qquad X = 1 \$ (Y \text{ GE } 4) ;$$

The first is read “set the value of X equal to 1 if Y is non-zero, otherwise set $X = 0$ ”. The second is read “set the value of X equal to 1 if Y is greater than or equal to 4, otherwise set $X = 0$ ”. Note that when the conditional is on the right-hand side, it will override all previous assignments of values to X , in this case setting it at either 1 or 0. In our program, we have:

$$\begin{aligned} \text{ROWMAX}(R, C, S) &= 1 \$ (\text{PI}(R, C, S) \text{ EQ } \text{SMAX}(RR, \text{PI}(RR, C, S))) ; \\ \text{COLMAX}(R, C, S) &= 1 \$ (\text{PJ}(R, C, S) \text{ EQ } \text{SMAX}(CC, \text{PJ}(R, CC, S))) ; \\ \text{NE}(R, C, S) &= \text{ROWMAX}(R, C, S) * \text{COLMAX}(R, C, S) ; \end{aligned}$$

The first line says “set $\text{ROWMAX}(R, C, S) = 1$ if $\text{PI}(R, C, S)$ is the largest element (over the rows) of column C for trade cost S ”, other wise set it to zero”. SMAX stands for “set max”. We see the use of “alias” here to distinguish what we are searching over from what we are assigning. ROWMAX is finding firm i ’s best

response to strategy C played by firm j.

COLMAX similarly finds firm j's best response to a strategy R played by firm i. A best response gets a value of one, and all other values in the row get zeros.

NE(R,C,S) multiplies together the corresponding elements of the matrices ROWMAX and COLMAX. For a given value of S, NE(R,C,S) will be a matrix of zeros and ones, with a one indicating a Nash equilibrium. There will ten of these matrices 3x3 payoff matrices, one for each element of S (each level of tc_j).

Now we find the Nash equilibrium levels of profits and welfare. Here we have a conditional on the left-hand side of the equation. This means something different.

$$X\$Y = 1;$$

means "set X equal to 1 if Y is non-zero, otherwise leave the current value of X unchanged". When the conditional is on the left-hand side, it does not override the current value of X (i.e., set it to zero) when the condition is false, the

statement is just ignored when the condition is false.

```

LOOP (R,
LOOP (C,

PINE (S) $NE (R, C, S) = PI (R, C, S) ;
PJNE (S) $NE (R, C, S) = PJ (R, C, S) ;
WINE (S) $NE (R, C, S) = WI (R, C, S) ;
WJNE (S) $NE (R, C, S) = WJ (R, C, S) ;

) ;
) ;

```

This statement loops over the payoff matrix for a given S , and give $PINE(S)$ the value of firm i 's profits when it hits the Nash equilibrium and so forth for the other variables. This is a bit sloppy, it only works if the Nash equilibrium for a given value of S is unique, but that holds in this problem ($NE(R,C,S)$ is printed out so I can check). If there is more than one Nash equilibrium, the parameters will be assigned the values from the last Nash equilibrium encountered over the

payoff matrix.

It is important that the conditional appear on the left, otherwise GAMS will override the correct assignment with a zero assignment when it moves on to the next element after the Nash equilibrium.

The following statement causes the output to be displayed in the listing file.

```
DISPLAY TCJ, NE, PINE, PJNE, WINE, WJNE;
```

The set designators are not used in the display statement.

A great feature of GAMS is that the output in a listing file can be dumped into several types of graphics packages. I use EXCEL (but I wrote the book in WordPerfect I might add). The statements to dump to EXCEL sheets are as follows.


```
$LIBINCLUDE XLDUMP PINE NASH.XLS SHEET1!A2  
$LIBINCLUDE XLDUMP PJNE NASH.XLS SHEET1!A4  
$LIBINCLUDE XLDUMP WINE NASH.XLS SHEET1!A6  
$LIBINCLUDE XLDUMP WJNE NASH.XLS SHEET1!A8
```

```
*EXAMPLE 4
* this is the code for Figures 3.6 and 3.7
* it shows some of the set features of GAMS
* finding Nash equilibria over a 3x3 payoff matrix,
for different
* values of trade costs

* REGIME CHANGES AS COUNTRY J RAISES ITS TRADE COSTS.
```

```
SETS
```

```
R /0*2/
```

```
C /0*2/
```

```
S /1*10/;
```

```
ALIAS (R,RR);
```

```
ALIAS (C,CC);
```

```
PARAMETERS
```

```
PI (R,C,S)
```

```
PJ (R,C,S)
```

```
WI (R, C, S)
WJ (R, C, S)
ROWMAX (R, C, S)
COLMAX (R, C, S)
NE (R, C, S)
PINE (S)
PJNE (S)
WINE (S)
WJNE (S)
ALPHA
G
F
T
TJ
TCJ (S) ;

ALPHA = 12 ;
T = 1 ;
F = 10 ;
G = 12 ;
```

**** here is the loop over values of trade costs

LOOP(S,

TJ = 0.75 + 0.25*ORD(S);

TCJ(S) = TJ;

PI('2','2',S) = 2*(ALPHA/3)**2 - 2*G - F;

PI('1','2',S) = (ALPHA/3)**2 + ((ALPHA - 2*TJ)/3)**2
- G - F;

PI('0','2',S) = 0;

PI('2','1',S) = (ALPHA/3)**2 + ((ALPHA + T)/3)**2 -
2*G - F;

PI('1','1',S) = ((ALPHA + T)/3)**2 + ((ALPHA -
2*TJ)/3)**2 - G - F;

PI('0','1',S) = 0;

PI('2','0',S) = 2*(ALPHA/2)**2 - 2*G - F;

$$PI('1', '0', S) = (\text{ALPHA}/2)^2 + ((\text{ALPHA} - T_J)/2)^2 - G - F;$$

$$PI('0', '0', S) = 0;$$

$$WI('2', '2', S) = (1/2) * (2 * \text{ALPHA}/3)^2 + PI('2', '2', S);$$

$$WI('1', '2', S) = (1/2) * (2 * \text{ALPHA}/3)^2 + PI('1', '2', S);$$

$$WI('0', '2', S) = (1/2) * (\text{ALPHA}/2)^2;$$

$$WI('2', '1', S) = (1/2) * ((\text{ALPHA} + T + \text{ALPHA} - 2 * T)/3)^2 + PI('2', '1', S);$$

$$WI('1', '1', S) = (1/2) * ((\text{ALPHA} + T + \text{ALPHA} - 2 * T)/3)^2 + PI('1', '1', S);$$

$$WI('0', '1', S) = (1/2) * ((\text{ALPHA} - T)/2)^2;$$

$$WI('2', '0', S) = (1/2) * (\text{ALPHA}/2)^2 + PI('2', '0', S);$$

$$WI('1', '0', S) = (1/2) * (\text{ALPHA}/2)^2 + PI('1', '0', S);$$

$$WI('0', '0', S) = 0;$$

*

$$PJ('2', '2', S) = 2 * (\text{ALPHA}/3)^2 - 2 * G - F;$$

$$PJ('2', '1', S) = (\text{ALPHA}/3)^2 + ((\text{ALPHA} - 2 * T)/3)^2 -$$

$$G - F;$$

$$PJ('2', '0', S) = 0;$$

$$PJ('1', '2', S) = (ALPHA/3)**2 + ((ALPHA + TJ)/3)**2 - 2*G - F;$$

$$PJ('1', '1', S) = ((ALPHA + TJ)/3)**2 + ((ALPHA - 2*T)/3)**2 - G - F;$$

$$PJ('1', '0', S) = 0;$$

$$PJ('0', '2', S) = 2*(ALPHA/2)**2 - 2*G - F;$$

$$PJ('0', '1', S) = (ALPHA/2)**2 + ((ALPHA - T)/2)**2 - G - F;$$

$$PJ('0', '0', S) = 0;$$

$$WJ('2', '2', S) = (1/2)*(2*ALPHA/3)**2 + PJ('2', '2', S);$$

$$WJ('2', '1', S) = (1/2)*(2*ALPHA/3)**2 + PJ('2', '1', S);$$

$$WJ('2', '0', S) = (1/2)*(ALPHA/2)**2;$$

$$WJ('1', '2', S) = (1/2)*((ALPHA + TJ + ALPHA - 2*TJ)/3)**2 + PJ('1', '2', S);$$

$$WJ('1', '1', S) = (1/2)*((ALPHA + TJ + ALPHA - 2*TJ)/3)**2 + PJ('1', '1', S);$$

$$WJ('1', '0', S) = (1/2)*((ALPHA - TJ)/2)**2;$$

$$WJ('0', '2', S) = (1/2) * (ALPHA/2) ** 2 + PJ('0', '2', S);$$

$$WJ('0', '1', S) = (1/2) * (ALPHA/2) ** 2 + PJ('0', '1', S);$$

$$WJ('0', '0', S) = 0;$$

);

**** now we find the Nash equilibria for each value
of trade costs ****

$$\text{ROWMAX}(R, C, S) = 1\$ (\text{PI}(R, C, S) \text{ EQ } \text{SMAX}(\text{RR}, \text{PI}(\text{RR}, C, S))) ;$$

$$\text{COLMAX}(R, C, S) = 1\$ (\text{PJ}(R, C, S) \text{ EQ } \text{SMAX}(\text{CC}, \text{PJ}(R, \text{CC}, S))) ;$$

$$\text{NE}(R, C, S) = \text{ROWMAX}(R, C, S) * \text{COLMAX}(R, C, S) ;$$

**** set profits and welfare at the Nash eq value for
each trade cost ***

LOOP(R,
LOOP(C,

$$\text{PINE}(S) \$ \text{NE}(R, C, S) = \text{PI}(R, C, S) ;$$


```
PJNE (S) $NE (R, C, S) = PJ (R, C, S) ;  
WINE (S) $NE (R, C, S) = WI (R, C, S) ;  
WJNE (S) $NE (R, C, S) = WJ (R, C, S) ;
```

```
) ;
```

```
) ;
```

```
DISPLAY TCJ, NE, PINE, PJNE, WINE, WJNE;
```

```
**** dump the output to be plotted to EXCEL  
spreadsheets ****
```

```
$LIBINCLUDE SSDUMP PINE PROFI.XLS  
$LIBINCLUDE SSDUMP PJNE PROFJ.XLS  
$LIBINCLUDE SSDUMP WINE WELI.XLS  
$LIBINCLUDE SSDUMP WJNE WELJ.XLS
```

The listing file will give the following solution values. To save space in high-dimension problems, zero rows or columns of a matrix or zero values of scalars are not listed by GAMS.

---- 120 PARAMETER TCJ

1	1.000,	2	1.250,	3	1.500,	4	1.750,	5	2.000,	6	2.250
7	2.500,	8	2.750,	9	3.000,	10	3.250				

---- 120 PARAMETER NE

	1	2	3	4	5	6
1.1	1.000	1.000	1.000	1.000	1.000	1.000
+	7	8	9	10		
1.1	1.000	1.000				
2.1			1.000	1.000		

---- 120 PARAMETER PINE

1	7.889,	2	6.806,	3	5.778,	4	4.806,	5	3.889,	6	3.028
7	2.222,	8	1.472,	9	0.778,	10	0.778				

---- 120 PARAMETER PJNE

1	7.889,	2	8.618,	3	9.361,	4	10.118,	5	10.889,	6	11.674
7	12.472,	8	13.285,	9	5.111,	10	5.111				

---- 120 PARAMETER WINE

1	37.278,	2	36.194,	3	35.167,	4	34.194,	5	33.278,	6	32.417
7	31.611,	8	30.861,	9	30.167,	10	30.167				

---- 120 PARAMETER WJNE

1	37.278,	2	37.372,	3	37.486,	4	37.622,	5	37.778,	6	37.955
7	38.153,	8	38.372,	9	37.111,	10	37.111				

Note that there is no instance of multiple equilibria over the ten elements of set S. The (unique) Nash equilibria change from (1,1) to (2,1) when we hit $S = 9$ ($tc_j = 3$). The Nash equilibrium profit and welfare levels are dumped to EXCEL and do not have to be retyped.