Essays on Local Sensitivity Analysis and its Application in Economics

By

Huiming Wang

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy
(Economics)

at the
UNIVERSITY OF WISCONSIN-MADISON
2017

Date of final oral examination: 08/11/2017

The dissertation is approved by the following members of the Final Oral Committee:

Thomas F. Rutherford, Professor, Agriculture and Applied Economics
Michael Ferris, Professor, Computer Sciences
Corbett A. Grainger, Associate Professor, Agriculture and Applied Economics
Xiaodong Du, Associate Professor, Agriculture and Applied Economics
Chao Fu, Associate Professor, Economics

Contents

Acknowledgments

I am indebted to my advisor, Thomas Rutherford, for his invaluable guidance and encouragement at various stages of this research. I am very grateful to Michael Ferris and Corbett Grainger for their insightful discussions, comments and suggestions. I thank Xiaodong Du, Chao Fu, Bruno Lanz, Erwin Kalvelagen and Alex Meeraus for their discussions and useful comments. Finally, I would like to thank my parents and my wife, Xiaoxia Shi, for continuous support and encouragement.

Abstract

The three chapters of my dissertation introduce Local Sensitivity Analysis (LSA) tool, explore its role in economic modeling, especially its application in trade, econometrics and energy economy models.

Chapter 1 shows global consequences of U.S. protectionism in trade. It provides an improved welfare decomposition based on the Central Path Decomposition (CPD) method suggested by Harrison et al. (2000). New calculation tool called Local Sensitivity Analysis (LSA) is applied in Computable General Equilibrium (CGE) models to attribute welfare variation to discrete changes of policy instruments. It also helps to bring down computational costs of CPD.

Chapter 2 is jointly authored with Tom Rutherford. In this chapter, we show that although the General Algebraic Modeling System (GAMS) has a long history of applications in operations research and computable general equilibrium modeling, it is at present rarely employed for econometric estimation, in part because of the lack of tools required for calculating confidence intervals and other test statistics at a solution. This chapter documents new tools which facilitate the use of GAMS to solve a range of econometrics problems. Like Kalvelagen (2007b) and Kalvelagen (2007a), we take a pedagogic perspective and illustrate the advantages of algebraic modeling languages with concrete applications.

Chapter 3 is jointly authored with Tom Rutherford. In this chapter, we apply `LSA` to block decomposition approach and its application in *top-down/bottom-up* fashion energy-economy modeling. We show an improved decomposition algorithm which is not only more clear since `LSA` derives local gradients which relieves modelers from employing numerical differentiation, but also more robust over the *diagonalization* decomposition in many cases. Furthermore, other than the imposed own-price elasticities suggested by Böhringer and Rutherford (2009), block decomposition with `LSA` captures the full own- and cross-price demand elasticities matrix in the *bottom-up* partial equilibrium model which eliminates the decomposition procedure's dependence on exogenous demand elasticities.

Chapter 1

Local Sensitivity Analysis and Central Path Decomposition

1.1 Introduction

As Thomas L. Friedman, the author of "The World is Flat" (Friedman (2005)), a best selling book in the early 21^{st} century comments on the trend of globalization, he does not think major players involved in the globalization such as India or China would go self-isolation one day when they have been so deeply interlinked with the rest of the world. The name of the United States does not appear then for good reasons. More than a decade later, both India and China are still active in the global market, while U.S., once the leader of globalization who promotes free international factor movements, is in retreat from its long standing endorsement of free trade due to a series of political and economic reasons. It seems that a good potion of leaders of the current administration believe that previous commitments on free trade is hurting the U.S. economy thus it is necessary and potentially beneficial to raise trade barriers such as tariffs on goods that are not "Made in USA". Is this a wise move economically? How could this move affect the welfare of U.S. as well as its trading parters'? In this paper, we attribute welfare effects to some protectionist policy instruments, mainly import tariffs, in some Computable General Equilibrium (CGE) models. Our goal is to measure which policy instrument is responsible for how much welfare changes. We apply Local Sensitivity Analysis (LSA) on Central Path Decomposition (CPD) method suggested by Harrison et al. (2000) as main analytic tool. In general, policy simulations using general equilibrium models often involve multiple instruments. If the contributions attributed to those instruments add exactly to the overall simulation result, we call this a decomposition of the simulation. As discussed by Harrison et al. (2000), CPD presents an order-independent welfare decomposition along a straight line, starting from the original point with no changes in policy instruments toward their final value post-change, so called "natural path". Harrison et al. (2000) also explain why an order-independent way of calculating contributions seems desirable and we demonstrate those through an example later.

Instead of numerical differentiation method, We rely on `LSA` to calculate policy impacts on welfare along the "natural path" to prepare for CPD. We introduce `LSA` and how it works in section 3. In this paper, taking advantage of automatic differentiation techniques which are available in the General Algebraic Modeling System (GAMS), `LSA` improves computational efficiency which contributes to a faster CPD by rapidly retrieving first order derivatives locally. On the other hand, policy impact on welfare evaluated at discrete policy instrument point is a basic measure in policy evaluation. In order to perform more complete sensitivity analysis along the path to achieve precise decomposition, traditional procedure usually involves numerical integration: first solve the model multiple times at different policy points, then apply numerical differentiation at each fixed point to estimate its welfare effects. However, this numerical integration procedure could be computational intensive especially when the model is large with many policy instruments to evaluate, a large amount of discrete policy change points for each instrument does not help either. Our by-pass of this is to approximate local sensitivity results through quadratic interpolation along the decomposition path. Only two explicit solutions are required: one at the starting point and the other at the ending point when exogenous policy variations fully implemented. Though `LSA`, we now have easy access to both first and second order derivatives in many CGE models. We calibrate a quadratic marginal effects function based on these results. And we show that in many cases, our approximation method is not only accurate, but also fast comparing to the traditional numerical integration approach.

We demonstrate our decomposition/interpolation method based on simulations in multi-regional trade applications from the Global Trade Analysis Project, version 7 (henceforth "GTAP7") database. For example, suppose one region in the model raises its tariffs on imports from other regions and the others may option to retaliate, it requires a general equilibrium model to explain welfare consequences and we apply our method for policy evaluation. Key insights of this paper are as follows:

- Spillover effects are common when raising tariff on imports which neutralize the policy effects. And "trade war" between major economic powers would leads to welfare deterioration for each side as well as the world economy.

- With the help of `LSA`, accurate approximation of tariff impacts on welfare could be achieved by a weighted sum of two boundary (starting and ending) solutions along the decomposition path.

- In case of accurate decomposition, *i.e.*, when the required number of policy impacts evaluation is large enough, our decomposition procedure has speed advantage over the numerical integration procedure.

The remainder of this paper is organized as follows: Section 2 provides an overview of the HHP decomposition technique. Section 3 lays out the Local Sensitivity Analysis (`LSA`) tool. Section 4 presents the two point policy impacts estimation using `LSA`. Section 5 we predicts the effectiveness of some protectionist tariff interventions through CPD and in section 6 we conclude.

1.2 The HHP Decomposition Technique

General equilibrium provides an established micro-consistent approach for evaluating the impacts of public policy on resource allocation (efficiency) and the associated changes in income for economic agents ("equity"). It has been, and still is, widely used in analytical work for assessing policy measures, such as tax reforms, where market interactions potentially play an important role. However, for the sake of tractability, analytical approaches are typically rather simple and not sufficiently complex for applied policy analysis. Therefore, numerical models are commonly used to accommodate the systematic analysis of economic problems where analytical solutions are either not available or do not provide adequate information.

The main virtue of complex CGE models, *i.e.* the comprehensive and consistent quantification of direct and indirect policy impacts, constitutes also the major challenge for their use. As various partial effects, which may work in opposite directions, contribute to the overall effect, it can get very difficult to explain in depth the aggregate policy outcome. Numerical applications inherit some ambiguity in the interpretation of the results as long as it is not possible to make transparent the sign and the magnitude of individual effects. Therefore, procedures which allow the decomposition of general equilibrium effects in a meaningful way are very helpful for the understanding and interpretation of policy simulations. A deliberate decomposition not only facilitates analysis of the various sources of the total effects but also assures a more rigorous check for the correct numerical implementation of policy questions.

In the context of multilateral policy appraisal, Böhringer and Rutherford (2002) apply a decomposition that splits the overall economic effect into a domestic market effect keeping international prices constant, and an international market effect as a result of changes in international prices (terms of trade effect). In other words, the decomposition allows separation of the primary effect of domestic policy action from the secondary burden or benefit transmitted via changes in international prices. Yet, the procedure is not suited for quantifying how much of the total economic impact for one specific region is due to its own action and what is contributed by the individual actions of other regions.

In response to the need for accurate method of attributing welfare change to exogenous policy instruments, Harrison et al. (2000) propose a linear decomposition methodology called Central Path Decomposition (CPD). Imaging exogenous space as an n-dimensional ($n = 3$ here) cube with the starting point at the origin, when tariff instruments are set at 0 for all, $T(0, 0, 0)$, and the ending point to the diagonally opposite vertex (indexed by $T(\alpha, \beta, \gamma)$, where α, β, γ measure policy instrument change in each dimension). Decomposition along this "central" path inside the cube, from $T(0,0,0) \to T(\alpha,\beta,\gamma)$, is thus called CPD. To draw a parallel, if only one instrument at a time was changing, *i.e.*, policy instrument travels only along the edges of the cube, we call these policy change paths "edgewise" paths. For example, the highlighted path in the graph from $T(0,0,0) \to T(\alpha,0,0) \to T(\alpha,\beta,0) \to T(\alpha,\beta,\gamma)$ is an edgewise path. Unlike CPD, sequential decomposition along edgewise paths are order-dependent by construction. The large number of possible policy orderings (which equals $n!$), combined with the potential sensitivity of decomposition results to the order used to calculate contributions, are disadvantages of the sequential method of result decomposition.

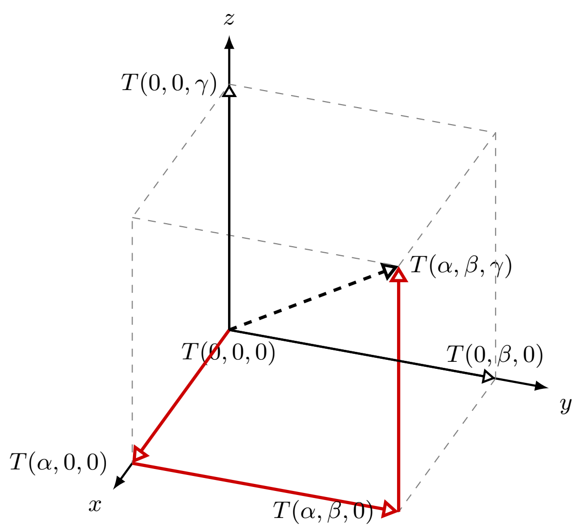

Figure 1.2.1: Central Path and Edgewise Path

Next, we are going to show how sensitive are estimates of the contributions to the order in which instruments are applied, and we compare CPD with these decomposition though edgewise paths in a North American Free Trade Agreement (NAFTA) renegotiation example. This example is based on GTAP7 with 8 regions indexed by r:

Regions	Description
CHN	China and Hong Kong
CAN	Canada
USA	United States
MEX	Mexico

ERU	European Union
OEC	Other OECD countries
LIC	Other low income countries
MIC	Other middle income countries

Table 1.1: Description of a Subset of Regions in 2004 GTAP Dataset

and 5 goods indexed by i.

Goods	Description
ENR	Energy goods
BAS	Basic material
AGR	Agriculture goods
MFR	Manufactures
SER	Services

Table 1.2: Description of a Subset of 2004 GTAP Dataset which Contains 5 Goods

Suppose U.S. pulls out from the NAFTA and raises its import tariffs from both Canada (CAN) and Mexico (MEX) by 45% and the two affected countries launch retaliatory tariffs against U.S.. And we could decompose welfare change in U.S. due to those 3 instruments:

[α] Raise of U.S. import tariffs from CAN and MEX by 45%.

[β] Raise of CAN import tariffs from U.S. by 45%.

[γ] Raise of MEX import tariffs from U.S. by 45%.

The first column of the following table shows the effect of applying instrument [α] to the equilibrium. The current equilibrium could be the original one if instrument [α] is the first to be applied, for example when the "Path" column of the table has value "Order $\alpha\beta\gamma$" or "Order $\alpha\gamma\beta$". The equilibrium could also be one that has already absorbed the effects of other instruments. For example, the effect of applying shock [α] could be more precisely labeled [$\alpha|\beta$] ([α] given [β]) when we apply shocks [α] to the post-simulation database produced by an equilibrium which observed instrument [β]. Effect of applying shock [α] now equals effect of applying both instruments [α] and [β] minus effect of applying instrument [β] alone to the original equilibrium. Similarly, when instrument [β] and [γ] are imposed before instrument [α], effect of [α] could be labeled as [$\alpha|\beta\gamma$] (α given β and γ) and calculated as effect of applying [α],[β],[γ] minus effect caused by instrument [β] and [γ] together. And tariff change path with this order is from $T(0,0,0) \rightarrow T(0,\beta,0) \rightarrow T(0,\beta,\gamma) \rightarrow T(\alpha,\beta,\gamma)$.

The first six rows of the table (1.3) document welfare effect of each policy instrument along different paths. Variation of effect between these rows shows that order-dependent welfare decomposition is sensitive to the policy change path. While in the 8^{th} row, we decompose welfare using CPD. It is not affected by the order in which instruments are implemented thus it attributes welfare change to policy instruments with much less noise. According to the row of "CPD", We find that tariff on imports from Canada and Mexico reduces U.S. consumption slightly, and as expected, retaliatory measures from these two could damage U.S. welfare more.

Path	Effect of [α]	Effect of [β]	Effect of [γ]	Total
Order $\alpha\beta\gamma$	-0.548	-2.576	-1.439	-4.563
Order $\alpha\gamma\beta$	-0.548	-1.366	-2.649	-4.563
Order $\beta\alpha\gamma$	-2.964	-0.160	-1.439	-4.562
Order $\beta\gamma\alpha$	-2.964	-1.737	0.139	-4.563
Order $\gamma\alpha\beta$	-1.672	-0.242	-2.649	-4.563
Order $\gamma\beta\alpha$	-1.672	-3.030	0.139	-4.563
Average	-1.728	-1.518	-1.316	-4.562
CPD	-0.067	-2.855	-1.596	-4.518

Table 1.3: Contributions of the 3 Instruments to US Welfare Change ($US million)

Meanwhile, raise of U.S. import tariff on the other two will damage welfare of Mexico with larger magnitude. In the following table about Mexico welfare decomposition, it is also clear that increased costs of exporting to the U.S. contributes the most of the welfare loss of Mexico:

Path	Effect of $[\alpha]$	Effect of $[\beta]$	Effect of $[\gamma]$	Total
Order $\alpha\beta\gamma$	-39.17	3.436	-8.033	-43.767
Order $\alpha\gamma\beta$	-39.17	-7.791	3.193	-43.768
Order $\beta\alpha\gamma$	2.738	-38.472	-8.033	-43.767
Order $\beta\gamma\alpha$	2.738	-9.244	-37.261	-43.767
Order $\gamma\alpha\beta$	-8.877	-38.083	3.193	-43.767
Order $\gamma\beta\alpha$	-8.877	2.371	-37.261	-43.767
Average	-15.103	-14.631	-14.034	-43.768
CPD	-38.594	3.078	-7.817	-43.333

Table 1.4: Contributions of the 3 Instruments to Mexico Welfare Change ($US million)

And it seems that some regions such as China could be potentially benefited from the "NAFTA renegotiation" which is a sign of so called "spillover" effects. Intuition behind this is straight forward: when prices of region A's imports from B are higher due to raised tariffs, consumers in region A would purchase substitutes from a third region C which is not involved in this tariff war. Overall, it is easy to see that spillover effects are not large enough to cover the initial damages caused by raising tariffs and the total welfare of the world is lower post-simulation.

Region	Total Welfare Effect
CHN	12.428
CAN	-43.266
USA	-4.562
MEX	-43.767
EUR	0.947
OEC	1.861
LIC	-1.788
MIC	1.878

Total	-76.269

Table 1.5: Total Welfare Change of the 3 Instruments ($US million)

Next, we revisit the CPD approach to present it in more details. CPD may be best explained when an endogenous variable Z be expressed as an explicit function of a vector t of exogenous variables (policy instruments), and t moves through a straight line from its starting point t_0 to its ending point t_1. Suppose changes of consumption level in region r with respect to U.S. import tariff change, Δc^r can be expressed as a function F of policy variables τ_s:

$$\Delta c^r = F(\tau_s),$$

in which tariff τ_s is associated with good i imported from region s, $s \in (s_1, s_2, \ldots, s_n)$ (n is the total number of source regions) into the region U.S.. Suppose we only consider one good, for example, manufacture good. We could then skip the good index i for clearance henceforth. Suppose that the tariff vector moves along some path beginning at $\tau_{s^0} = (\tau_{s_1^0}, \tau_{s_2^0}, \ldots, \tau_{s_n^0})$ and ending at $\tau_{s^1} = (\tau_{s_1^1}, \tau_{s_2^1}, \ldots, \tau_{s_n^1})$. Now suppose we divide import tariff instrument into 100 installments. Grand total change in consumption level of region r could then be approximated:

$$\Delta c^r = \Sigma_s \frac{\partial c^r}{\partial \tau_s} d\tau_s \tag{1.2.1}$$

in which $d\tau_s$ is one installment of total tariff change, $d\tau_s = \frac{\tau_{s^1} - \tau_{s^0}}{100}$ and $\frac{\partial c^r}{\partial \tau_s}$ is the consumption variation of region r caused by U.S. import tariff on manufacture good from region s. In this way, we distribute total welfare change Δc^r among the n policy instruments.

Next, we introduce basic ideas behind our computational tool which calculates local dependence of endogenous variables on policy instruments.

1.3 Local Sensitivity Analysis

In order to find derivatives such as $\frac{\partial c^r}{\partial \tau_s}$ in equation (1.2.1) at its solution point in an economic models, basic calculus about finite difference gives us some ideas. For example, when value change u approaching 0, *i.e.*, a disturbance of value, $u \to 0$, we approximate first and second order derivatives of function $f(x, y)$ evaluated at point (x_0, y_0) as follows:

$$f'(x,y)|_{x_0,y_0} \approx \frac{f(x_0+u,y_0) - f(x_0,y_0)}{u},$$

$$f''_{xx}(x,y)|_{x_0,y_0} \approx \frac{f(x_0+u,y_0) - 2f(x_0,y_0) + f(x_0-u,y_0)}{u^2},$$

We also find second order derivatives of function $f(x, y)$ with respect to both arguments (x, y) as perturbation $u \to 0, v \to 0$:

$$f''_{xy}(x,y)|_{x_0,y_0} \approx \frac{f(x_0+u,y_0+v) - f(x_0+u,y_0-v) - f(x_0-u,y_0+v) + f(x_0-u,y_0-v)}{4uv}.$$

With so called numerical differentiation method, we solve economic models to find perturbed values of function $f(x, y)$ as above and we approximate derivatives based on those values. With the help of automatic differentiation techniques in GAMS, we put together a calculation tool called Local Sensitivity Analysis (`LSA`) through which we retrieve first/second order derivatives in a package. We introduce some under hood mathematics of this `LSA` tool here.

An economic model may be formulated as a system of equations

$$F_i(z;t) = 0 \quad i = 1, \ldots, n$$

in which $z \in \mathrm{R^n}$ is a vector of equilibrium variables and $t \in \mathrm{R^m}$ is a vector of strategic policy instruments. The dependence of z_j $(j = 1, \ldots, n)$ on t_k $(k = 1, \ldots, m)$, $\frac{\mathrm{dz_j}}{\mathrm{dt_k}}$, can be deduced from the implicit function theorem which identifies changes in the endogenous variables (z_j) which maintain local feasibility of the system of equations. In other words, for each instrument k, the values of $\frac{\mathrm{dz_j}}{\mathrm{dt_k}}$ satisfy the system of equations

$$\frac{\mathrm{d}}{\mathrm{dt_k}} F_i(z;t) = \sum_j \frac{\partial F_i}{\partial z_j} \frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \frac{\mathrm{dF_i}}{\mathrm{dt_k}} = 0 \quad \forall i. \tag{1.3.1}$$

This is a system of n equations ($i = 1, \ldots, n$) in n unknowns ($j = 1, \ldots n$). we seek solutions of the system of equations:

$$\sum_j \left(\frac{\partial F_i}{\partial z_j}\right) \frac{\mathrm{dz_j}}{\mathrm{dt_k}} = -\frac{\mathrm{dF_i}}{\mathrm{dt_k}}.$$

In matrix notation, this can be written as

$$\left(\frac{\mathrm{dz.}}{\mathrm{dt_k}}\right) = -\left(\nabla_z F(z;t)\right)^{-1} \ \nabla_{t_k} F(z;t) \tag{1.3.2}$$

if the Jacobian matrix $\nabla_z F(z;t)$ is invertible. In a range of applications we need to determine the local dependence of z on t, i.e.

$$\frac{\mathrm{dz_i}}{\mathrm{dt_k}} \approx \left.\frac{\mathrm{dz_i}}{\mathrm{dt_k}}\right|_{\bar{t}} + \sum_\ell \left.\frac{\mathrm{d^2z_i}}{\mathrm{dt_k dt_\ell}}\right|_{\bar{t}} (t_\ell - \bar{t}_\ell),$$

in which higher order derivatives are obtained by differentiation of (1.3.1) with respect to t_ℓ:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{dt_\ell}}\frac{\mathrm{d}}{\mathrm{dt_k}} F_i(z;t) &= \frac{\mathrm{d}}{\mathrm{dt_\ell}} \left(\sum_j \frac{\partial F_i}{\partial z_j}\frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \frac{\mathrm{dF_i}}{\mathrm{dt_k}} \right) \\
&= \sum_j \frac{\partial^2 F_i}{\partial z_j \mathrm{dt_\ell}}\frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \sum_{jj'} \frac{\partial^2 F_i}{\partial z_j \partial z_{j'}}\frac{\mathrm{dz_{j'}}}{\mathrm{dt_\ell}}\frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \sum_j \frac{\partial F_i}{\partial z_j}\frac{\mathrm{d^2z_j}}{\mathrm{dt_k dt_\ell}} + \sum_j \left(\frac{\partial^2 F_i}{\partial z_j \mathrm{dt_k}}\frac{\mathrm{dz_j}}{\mathrm{dt_\ell}} + \frac{\mathrm{d^2F_i}}{\mathrm{dt_k dt_\ell}} \right) \\
&= 0.
\end{aligned}$$

The unknowns in this equation are the second derivatives $\zeta_j^{k\ell} = \frac{\mathrm{d^2z_j}}{\mathrm{dt_k dt_\ell}}$. Fixing k and ℓ, then the n-vector ζ solves the n-equation linear system of equations:

$$\sum_j \left(\frac{\partial F_i}{\partial z_j}\right) \ \zeta_j^{k\ell} = -g_i^{k\ell}$$

where the right-hand side vector can be computed from the Hessian matrix and the first-order derivatives already computed using (1.3.2):

$$g_i^{k\ell} = \sum_j \left(\frac{\partial^2 F_i}{\partial z_j \mathrm{dt_\ell}}\frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \frac{\partial^2 F_i}{\partial z_j \mathrm{dt_k}}\frac{\mathrm{dz_j}}{\mathrm{dt_\ell}} \right) + \sum_{jj'} \frac{\partial^2 F_i}{\partial z_j \partial z_{j'}}\frac{\mathrm{dz_{j'}}}{\mathrm{dt_\ell}}\frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \frac{\mathrm{d^2F_i}}{\mathrm{dt_k dt_\ell}}.$$

In matrix notation, we have:

$$\frac{\mathrm{d^2 z_.}}{\mathrm{dt_k dt_\ell}} = -\nabla_z F(z;t)^{-1} \left(\sum_j \left(\nabla_{z_j t_\ell} F\right) \frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \sum_j \left(\nabla_{z_j t_k} F\right) \frac{\mathrm{dz_j}}{\mathrm{dt_\ell}} + \sum_{jj'} \left(\nabla_{z_j z_{j'}} F\right) \frac{\mathrm{dz_{j'}}}{\mathrm{dt_\ell}} \frac{\mathrm{dz_j}}{\mathrm{dt_k}} + \nabla_{t_k t_\ell} F \right). \tag{1.3.3}$$

Dual Formulation

Some may raise concerns over the double summation term in the previous formula since in practice, it slows the calculation down. We borrow some ideas from the dual formulation of the model to by-pass this. If we solve the linear program:

$$\max z_\ell$$

subject to:

$$F(z;t) = 0,$$

the Lagrange multiplier on constraint i, $p_{\ell i}$, represents the total derivative:

$$p_{\ell i} = \frac{\mathrm{d}z_\ell}{\mathrm{d}F_i}.$$

The first order condition for t_k implies:

$$\frac{\mathrm{d}z_\ell}{\mathrm{d}t_k} = -\sum_i \left(\frac{\partial F_i}{\partial t_k}\right) p_{\ell i}$$

The second derivatives are then

$$\frac{\mathrm{d}^2 z_\ell}{\mathrm{d}t_1 \mathrm{d}t_2} = \sum_i p_{\ell i} \frac{\partial^2 F_i}{\partial t_1 \partial t_2} + \sum_i p_{\ell i} \sum_j \frac{\partial^2 F_i}{\partial t_1 \partial z_j} \frac{\mathrm{d}z_j}{\mathrm{d}t_2} + \sum_i \frac{\partial F_i}{\partial t_1} \frac{\mathrm{d}p_{\ell i}}{\mathrm{d}t_2} \tag{1.3.4}$$

The right-hand-side of the linear program is vacuous, so the dual program consists of a linear system of equations:

$$\sum_i \frac{\partial F_i}{\partial z_j} p_{\ell i} = \begin{cases} 1 & j = \ell \\ 0 & j \neq \ell \end{cases}$$

Taking the total derivative, we have:

$$\sum_i p_{\ell i} \left(\frac{\partial^2 F_i}{\partial z_j \partial t} + \sum_{j'} \frac{\partial^2 F_i}{\partial z_j \partial z_{j'}} \frac{\mathrm{d} z_{j'}}{\mathrm{d}t} \right) + \sum_i \frac{\partial F_i}{\partial z_j} \frac{\mathrm{d} p_{\ell i}}{\mathrm{d}t} = 0,$$

and hence:

$$\frac{\mathrm{d} p_{\ell i}}{\mathrm{d}t} = - \left(\frac{\partial F_i}{\partial z_j}^T \right)^{-1} \left(\sum_i p_{\ell i} \left(\frac{\partial^2 F_i}{\partial z_j \partial t} + \sum_{j'} \frac{\partial^2 F_i}{\partial z_j \partial z_{j'}} \frac{\mathrm{d} z_{j'}}{\mathrm{d}t} \right) \right),$$

And we plug this back to equation (1.3.4) to find $\frac{\mathrm{d}^2 z_\ell}{\mathrm{d}t_1 \mathrm{d}t_2}$ without computational expensive double summation term in equation (1.3.3). We document the syntax of deriving first and second order derivatives using `LSA` in the Appendix. Next, we introduce our interpolation method with the help of `LSA` to approximate tariff impacts on the decomposition path.

1.4 LSA Approximation of the Central Path

Some natural questions related to welfare effects caused by policy instruments are, how does the welfare of some region vary along the policy change path? Is there an optimal tariff rate for some region, such as U.S. to implement? If yes, what is that rate? In principal, setting up a CGE model before imposing different tariff rates and solving the model multiple times sounds like a straight forward strategy for these questions, which is called the numerical integration method. However, in many occasions, this solution is expensive in calculation especially when we have large amount of policy points to evaluate. Unfortunately, as we discussed earlier, a precise CPD requires us to evaluate policy impacts on large number of points. Good news is now `LSA` helps us in addressing these questions in a less costly way. To keep track of the policy impacts, we interpolate first order derivatives at fixed policy points on decomposition path though a quadratic approximation using first/second order derivatives reported by `LSA` at both end of the "natural path". In order to do that, we solve a convex least squares problem to calibrate unknown coefficients in our assumed quadratic tariff impacts function. Our interpolation method guarantees that once we find first/second derivatives at the starting and ending points of instrument change path, additional time spent on interpolation is trivial, thus total computational time will not increase significantly with the increased number of discrete instrument points as it would in numerical integration. Note that we choose quadratic approximation of first order derivative because we are now equipped with up to second order derivatives.

The following section describes a small multi-regional trade example within the GTAP7. This project's objectives include the provision of a documented, publicly available, global, general equilibrium data base, and to conduct seminars on a regular basis to inform the research community about how to use the data in applied economic analysis. GTAP has lead to the establishment of a global network of researchers who share a common interest of multi-region trade analysis and related issues. In this example, model structure as well as equilibrium conditions (market clearance, zero-profit and budget balance) are the same as in the GTAP7inGAMS model in Rutherford (2010). The data characterize intermediate demand and bilateral trade in 2004, including tax rates on imports and exports and other indirect taxes are as stated in Rutherford (2010). The core GTAP model is a static, multi-regional model which tracks the production and distribution of goods in the global economy. In GTAP the world is divided into regions (typically representing individual countries), and each region's final demand structure is composed of public and private expenditure across goods. The model is based on optimizing behavior. Consumers maximize welfare subject to budget constraint with fixed levels of investment and public output. Producers combine intermediate inputs, and primary factors (skilled and unskilled labor, land, resources and physical capital) at least cost subject for given technology. The dataset includes a full set of bilateral trade flows with associated transport costs, export taxes and tariffs.

And here lists some endogenous variables typically covered in the model:

Variables	Description

C(r)	Consumption in region r
Y(g,r)	Supply of good g in region r
M(i,r)	Imports of good i into region r
YT(j)	Transportation services required by good j
FT(f,r)	Specific factor transformation
P(g,r)	Domestic output price of good g from region r
PM(j,r)	Import price of good j from region r
PT(j)	Transportation services
PF(f,r)	Primary factors rent
PS(f,g,r)	Sector-specific primary factors
RA(r)	Representative agent in region r
TAU(i,s,r)	Tariff schedule on good i from region s into r

Table 1.6: Description of a Subset of Variables in 2004 GTAP Dataset

We start with a simple example in which we implement a 50% increase in the U.S. import tariff on all commodities from all other regions in the model given the protectionist tendency of the current administration. We apply CPD to attribute welfare variation in one region to policy instrument by regions, also we describe welfare change along decomposition path graphically. After that, We compare results across different weights on the marginal effects of the starting point to pick a best weight for each region. To begin with our analysis, it seems that U.S. keeps low rates from most of the regions for most of her imports according to the GTAP data (2004) and a 50% raise is significant.

Goods	Original Region	Import Tariff (before)	Import Tariff (after)
ENR	CHN	0.001	0.501
ENR	EUR	0.011	0.511
ENR	OEC	0.006	0.506
ENR	LIC	1.2E-6	0.500

ENR	MIC	0.001	0.501
BAS	CHN	0.018	0.518
BAS	MEX	1.7E-4	0.500
BAS	EUR	0.013	0.513
BAS	OEC	0.014	0.514
BAS	LIC	3.9E-4	0.500
BAS	MIC	0.007	0.507
AGR	CHN	0.013	0.513
AGR	MEX	0.005	0.505
AGR	EUR	0.028	0.528
AGR	OEC	0.042	0.542
AGR	LIC	0.054	0.554
AGR	MIC	0.027	0.527
MFR	CHN	0.031	0.531
MFR	CAN	0.002	0.502
MFR	MEX	2.6E-4	0.500
MFR	EUR	0.014	0.514
MFR	OEC	0.012	0.512
MFR	LIC	0.095	0.595
MFR	MIC	0.023	0.523

Table 1.7: Proposed USA Import Tariff Changes Based on 2004 GTAP Dataset

Denote the first order derivative of consumption of region r with respect to U.S. import tariff change on good i from region s as:

$$d(t) = a + b_1 t + b_2 t^2, \tag{1.4.1}$$

in which for each region r, when U.S. raises import tariff on good i from region s, a, b_1, b_2 are unknown scalar coefficients, t is a scalar tariff between 0 and Δt. Only two solutions at both ends of "natural path" are required in this approximation approach. We estimate tariff impacts on welfare of region r evaluated at each point t, $0 < t < \Delta t$ through interpolation. With the help of `LSA`, as introduced in the Appendix, we derivate first/second order derivatives *c_tau* and *c_tau_tau* at both ends of change path. We then have $d(\Delta t) = \left(\frac{dc_r}{d\tau_{is}}\right)|_{\tau^1_{is}}$, $d(0) = \left(\frac{dc_r}{d\tau_{is}}\right)|_{\tau^0_{is}}$, $d'(\Delta t) = \Sigma_{i'j}\left(\frac{d^2c_r}{d\tau_{is}d\tau_{i'j}}\right)|_{\{\tau^1_{is}\tau^1_{i'j}\}}$ and $d'(0) = \Sigma_{i'j}\left(\frac{d^2c_r}{d\tau_{is}d\tau_{i'j}}\right)|_{\{\tau^0_{i,s}\tau^0_{i'j}\}}$[1] ready for calibration. According to equation (1.4.1), we first evaluate it at the starting point $t = 0$ to get

$$a = d(0),$$

Plug this into equation (1.4.1) and evaluate it at $t = \Delta t$, we have constraint on coefficient b_1 and b_2 as

$$b_1\Delta t + b_2(\Delta t)^2 = d(\Delta t) - d(0). \tag{1.4.2}$$

Suppose w is our weight index on tariff impacts of the starting point $t = 0$. When $w = 1$, we match the first order derivatives of the right hand side of function (1.4.1) at starting point, we then have

$$b_1 = d'(0),$$

On the other hand, if we assign no weight on sensitivity results of the starting point, $w = 0$, we match our first order derivatives of the quadratic function when Δt is applied, *i.e.*,

$$b_1 + 2b_2\Delta t = d'(\Delta t).$$

When weight index is between 0 and 1, $0 < w < 1$, optimal coefficients could be calculated by minimizing a weighted sum of squares of these two extreme matches, we solve:

$$\min_{b_1,b_2} w\left(b_1 - d'(0)\right)^2 + (1-w)\left(b_1 + 2b_2\Delta t - d'(\Delta t)\right)^2$$

[1] Here τ^0_{is} stands for tariff at the starting point and $\tau^1_{is} = \tau^0_{is} + \Delta t$ stands for tariff at the ending point, j is another name of s, i' is an alias name of i.

subject to the constraint 1.4.2. It leads us to calibrated coefficients such as:

$$b_1 = wd'(0) + (1-w)\left(\frac{2\,(d(\Delta t) - d(0))}{\Delta t} - d'(\Delta t)\right),$$

and

$$b_2 = \frac{(2w-1)\,(d(\Delta t) - d(0))}{\Delta t^2} - \frac{(1-w)d'(\Delta t) - wd'(0)}{\Delta t}.$$

As a result, we thus have a calibrated first order derivative function in quadratic form and we could do interpolation of tariff impacts and policy simulation based on this function. For example, when we assign the values of the array $(0, 0.1, 0.2, \ldots, 1)$ to weight index w, we have a set of tariff impacts on welfare with different weights on that of the original equilibrium.

After we interpolate these first order derivatives, we measure how good is our approximation. Based on the equation (1.2.1), we compare the error between the change in welfare levels $(c^r_{s1} - c^r_{s0})$ due to tariff instruments in regions s and the summation of tariff impacts times partial tariff changes, $\Sigma \frac{\partial c^r}{\partial \tau_{i,s}} d\tau_{i,s}$. The following list is for small GTAP7 dataset with 5 goods and 100 installments of tariff instruments, we find that the approximation is quite accurate when change of tariff $d\tau^h$ were small enough. Note that the column indexed by "exact" contains tariff impacts calculated though numerical integration, the "linear" column is a linear interpolation between boundary points. We can see that the "exact" column produces very small approximation error for each region, while the "linear" column is not so good an approximation as a whole. For each region r, we could pick the best weights to put on the original equilibrium solutions by minimizing approximation errors across weights.

```
----     Measure of approximation (Unit: trillion)
           exact       linear        0%          10%         20%         30%         40%


CHN -7.40787E-4        0.003       -0.002       -0.002       -0.002       -0.001       -0.001
CAN 4.059057E-4        0.013       -0.002       -0.001 -8.86740E-4 -4.56563E-4 -2.63853E-5
USA 5.091879E-5       -0.009        0.005        0.004        0.004        0.003        0.002
MEX 2.598213E-4        0.012       -0.002       -0.001 -8.88350E-4 -4.97726E-4 -1.07103E-4
```

EUR	-1.68874E-5	6.731216E-4	5.012815E-5	5.278997E-5	5.545179E-5	5.811360E-5	6.077542E-5
OEC	-9.28876E-5	6.615233E-4	2.494470E-4	2.359047E-4	2.223624E-4	2.088201E-4	1.952779E-4
LIC	-1.16365E-4	0.001	6.214717E-4	5.235491E-4	4.256265E-4	3.277038E-4	2.297812E-4
MIC	-2.69562E-4	6.947913E-4	-1.92578E-4	-1.65508E-4	-1.38438E-4	-1.11368E-4	-8.42979E-5

+	50%	60%	70%	80%	90%	100%
CHN	-9.49960E-4	-7.09302E-4	-4.68645E-4	-2.27988E-4	1.266983E-5	2.533273E-4
CAN	4.037922E-4	8.339696E-4	0.001	0.002	0.002	0.003
USA	0.001	5.257474E-4	-2.47195E-4	-0.001	-0.002	-0.003
MEX	2.835209E-4	6.741444E-4	0.001	0.001	0.002	0.002
EUR	6.343723E-5	6.609905E-5	6.876087E-5	7.142268E-5	7.408450E-5	7.674631E-5
OEC	1.817356E-4	1.681933E-4	1.546510E-4	1.411088E-4	1.275665E-4	1.140242E-4
LIC	1.318586E-4	3.393602E-5	-6.39866E-5	-1.61909E-4	-2.59832E-4	-3.57754E-4
MIC	-5.72279E-5	-3.01579E-5	-3.08787E-6	2.398215E-5	5.105217E-5	7.812220E-5

Next, we apply our interpolation and decomposition method in a larger subset of GTAP7 dataset with the following 23 goods:

Goods	Description
DWE	Dwellings
OIL	Refined oil products
GAS	Natural gas works
OMN	Mining (MINING)
LUM	Wood and wood-products (WOODPRO)
PPP	Paper-pulp-print (PAPERPRO)
CRP	Chemical industry (CHEMICAL)
NMM	Non-metallic minerals (NONMET)
I_S	Iron and steel industry (IRONSTL)
NFM	Non-ferrous metals (NONFERR)

ELE	Electricity and heat
OME	Other machinery (MACHINE)
OMF	Other manufacturing (INONSPEC)
CNS	Construction (CONSTRUC)
ATP	Air transport
COL	Coal transformation
CRU	Crude oil
TEQ	Transport equipment (TRANSEQ)
FPR	Food products (FOODPRO)
TWL	Textiles-wearing apparel-leather (TEXTILES)
AGR	Agricultural products
TRN	Transport
SER	Commercial and public services

Table 1.8: Description of a Subset of 2004 GTAP Dataset which Contains 23 Goods

This new version of the GTAP dataset could consume more computational resources when solving for derivatives. We run our decomposition method on this larger model, and compare the total computational time in this dataset with that in the small dataset with 5 goods. It seems that in a model with smaller dataset, quadratic approximation method dominate numerical integration in speed while in a larger model, it would cost about 230 seconds per point to report those second order derivatives. However, one benefit of our interpolation method is we have flat computational cost for different number of fixed instruments points while computational time consumed in numerical integration is increasing with the number of points. Thus, when we have enough policy points to work on which is required by the accuracy of CPD approximation, we have advantage in computational cost through implementing quadratic approximation over numerical integration. Note that in first column of the following table, Q.A. stands for Quadratic Approximation and N.I. stands for Numerical Integration. Thus Q.A.10 means we use quadratic approximation when there is 10 policy points to evaluate along the path, *etc.*

Model with:	5 goods	23 goods
Method and Interior Points	Elapsed Time (s)	Elapsed Time (s)
Q.A.10	14	463
Q.A.20	14	465
Q.A.40	14	475
N.I.10	17	147
N.I.20	22	274
N.I.40	39	525

Table 1.9: Speed Comparison between Quadratic Approximation and Numerical Integration

Similar to those in the smaller dataset with 5 goods, our interpolation method works well in a larger dataset in approximation accuracy. The following is a list of closeness measure of our approximation in a model with 23 goods and 100 tariff instruments.

```
----    Measure of approximation (Unit: trillion)
            exact       linear         0%          10%          20%          30%          40%


CHN -7.11661E-4        0.002 -6.70782E-4 -5.98597E-4 -5.26411E-4 -4.54226E-4 -3.82040E-4
CAN 5.452290E-4        0.008       0.002       0.002       0.002       0.002       0.001
USA -2.50500E-5       -0.009       0.005       0.004       0.004       0.003       0.002
MEX 5.327103E-4        0.008 -2.66409E-5 1.669053E-4 3.604516E-4 5.539978E-4 7.475440E-4
EUR -3.74113E-5 4.930886E-4        0.001       0.001       0.001       0.001 8.734630E-4
OEC -9.98802E-5 7.236190E-4 9.201365E-4 8.367346E-4 7.533328E-4 6.699309E-4 5.865291E-4
LIC 8.435324E-5        0.002       0.001       0.001 9.372267E-4 8.080218E-4 6.788169E-4
MIC -1.68618E-4        0.002 -4.77985E-4 -3.98577E-4 -3.19169E-4 -2.39761E-4 -1.60354E-4


  +       50%          60%          70%          80%          90%         100%
```

CHN	-3.09855E-4	-2.37669E-4	-1.65484E-4	-9.32983E-5	-2.11128E-5	5.107264E-5
CAN	0.001	0.001	0.001	0.001	0.001	0.001
USA	0.001	7.031826E-4	-1.89007E-5	-7.40984E-4	-0.001	-0.002
MEX	9.410903E-4	0.001	0.001	0.002	0.002	0.002
EUR	7.381290E-4	6.027950E-4	4.674610E-4	3.321270E-4	1.967930E-4	6.145896E-5
OEC	5.031273E-4	4.197254E-4	3.363236E-4	2.529217E-4	1.695199E-4	8.611804E-5
LIC	5.496120E-4	4.204071E-4	2.912021E-4	1.619972E-4	3.279231E-5	-9.64126E-5
MIC	-8.09457E-5	-1.53781E-6	7.787005E-5	1.572779E-4	2.366858E-4	3.160936E-4

Next, we apply our decomposition method with different tariff proposals to discuss related welfare effects.

1.5 Tariff Instruments and Welfare

U.S. Raises Import Tariff from all Other Regions

In the previous section, we suppose that the U.S. raises its import tariffs on all imports from other regions. We decompose total welfare change in region r according to policy instruments absorbed in region s, and we have change of consumption level in region r due to tariff change in region s, Δc_s^r and we draw a graph of policy impacts along the change path for each combination of r, s.

We first check its policy consequences in the region of China. Imposing import tariffs on Chinese goods could probably cause less competitive imports from China, but who is going to befit from that? From the graphic first order derivatives of welfare change in China with respect to U.S. import tariff, we find that small amount of increase of U.S. import tariffs on all manufacture goods from China could affect Chinese welfare negatively. However, when import tariffs from U.S. keep going up, its damage on Chinese welfare would go smaller:

At the same time, welfare effects of rising U.S. tariffs against imports from China at home (for region U.S.A) does increase U.S. welfare initially when tariff raise are relatively small, but this positive impacts would go to zero if U.S. imposes a large enough tariff raise (0.5 in

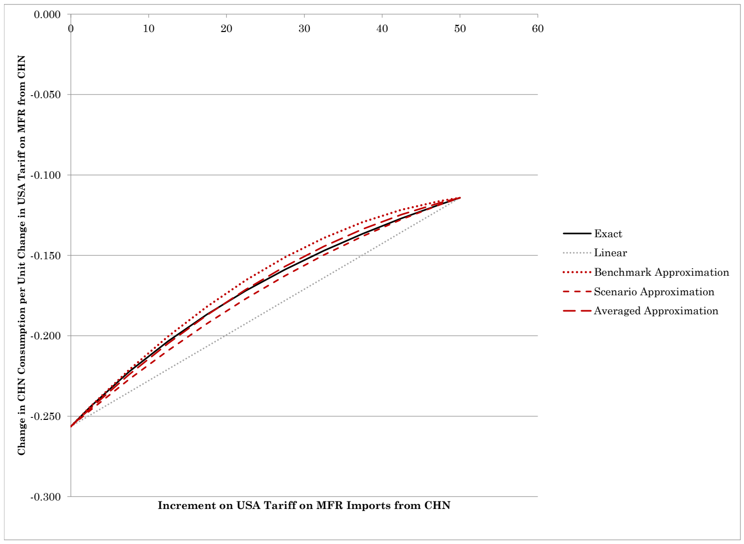

Figure 1.5.1: Welfare Change in China by US import tariff on Chinese Manufactures

this case).

An interesting marginal welfare change with respect to U.S. raising import tariff against Chinese goods in the region of Canada is shown here. We find that Canadian welfare benefits from U.S.'s punitive trade measures against China, it seems to indicate tariff increase against one region would result in importing more from another.

And similar to the case of China, raising U.S. import tariffs against Canadian goods negatively affects welfare of Canada, it keeps that way when U.S. implements a larger tariff raise.

Trade War between the U.S. and China

In this section, we introduce a more realistic application of our decomposition method in a trade war scenario. Tariff policy evaluation is not complete if we suggest imposing punitive tariffs on some regions without potential retaliation from those regions in mind. In this scenario, assume U.S. raises tariffs on imports from China by 45% and Chinese government fights back with the same retaliatory tariffs. It is commonly agreed that this is not a good

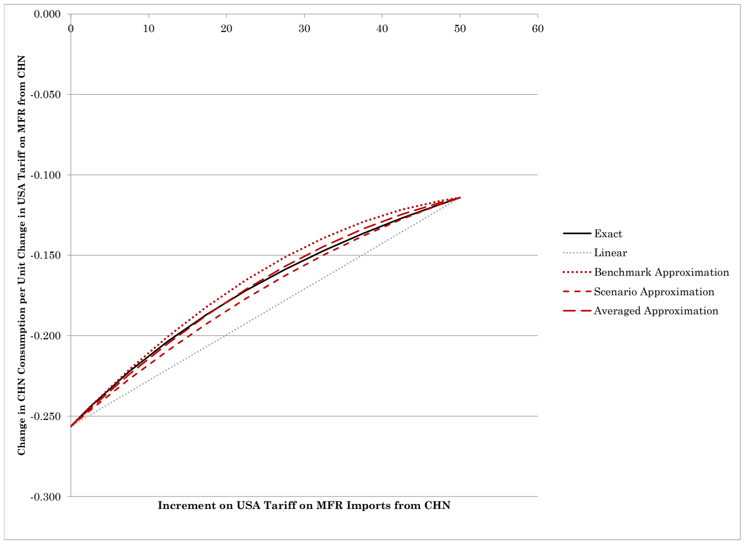

Figure 1.5.2: Welfare Change in China by US import tariff on Chinese Manufactures

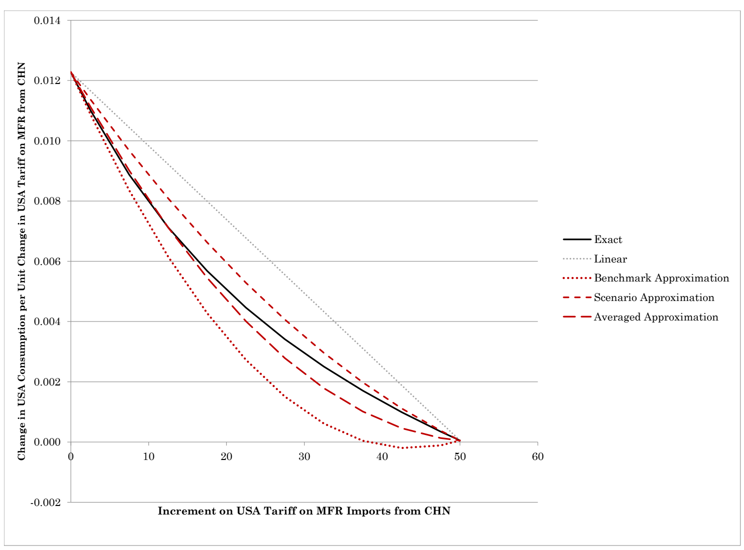

Figure 1.5.3: Welfare Change in US by US import tariff on Chinese Manufactures

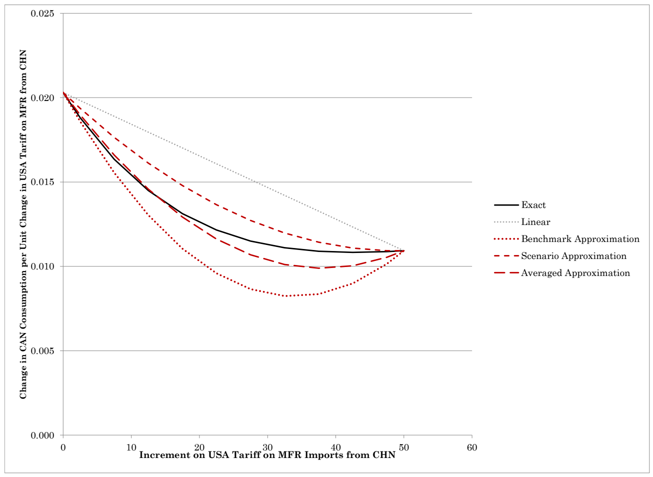

Figure 1.5.4: Welfare Change in Canada by US import tariff on Chinese Manufactures

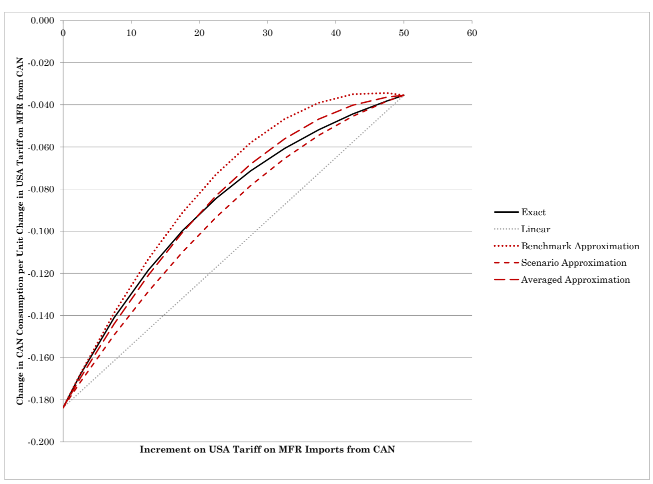

Figure 1.5.5: Welfare Change in Canada by US import tariff on Canadian Manufactures

idea for both regions, we are going to apply our calibrated tariff impacts function to check if that is the case. In this case, we decompose welfare changes in each region r according to two groups of tariff instruments, one is U.S.'s import tariffs on goods from China and the other is China's retaliatory measures against the U.S..

Similar to the graph when the U.S. imposes tariffs on all goods from other regions, tariff impacts on welfare of China go down as import tariff on its exports to U.S. goes up, however, the slope of this gradient curve is negative.

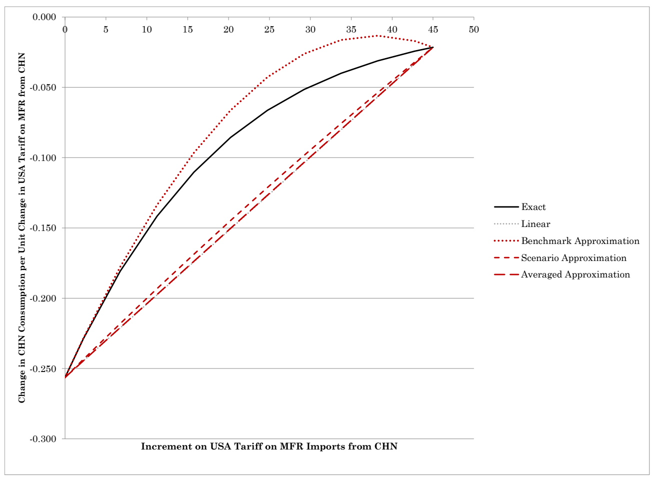

Figure 1.5.6: Welfare Change in China by US import tariff on Chinese Manufactures

Next, we find that welfare of China could be benefited from small tariff increase by Chinese retaliation. This positive impacts would go to zero quickly if we were talking about a large enough tariff raise from both sides.

On the US side: raising import tariffs against China initially increases U.S. welfare, however, higher tariff rates (more than 10% in this model) may hurt its welfare. The result is less ambiguous when we consider tariff responses from China, tariffs from both side would negatively affect U.S. consumption as follows:

For some substitution goods providers, such as Canada. It seems that spillover effect of trade war between the US and China works to their interest. Canada welfare is positively

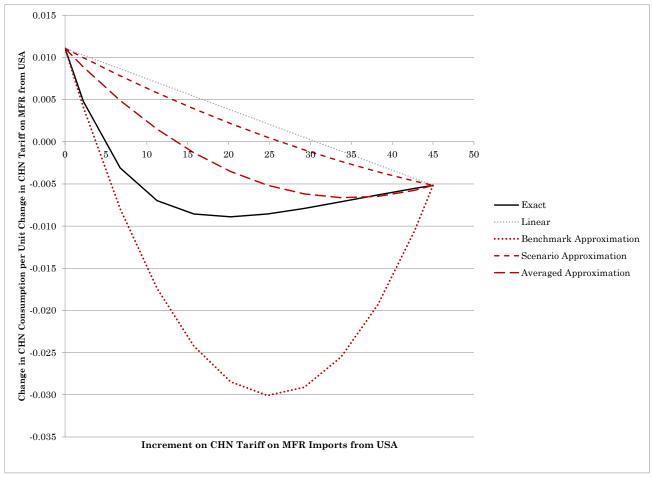

Figure 1.5.7: Welfare Change in China by China import tariff on US Manufactures

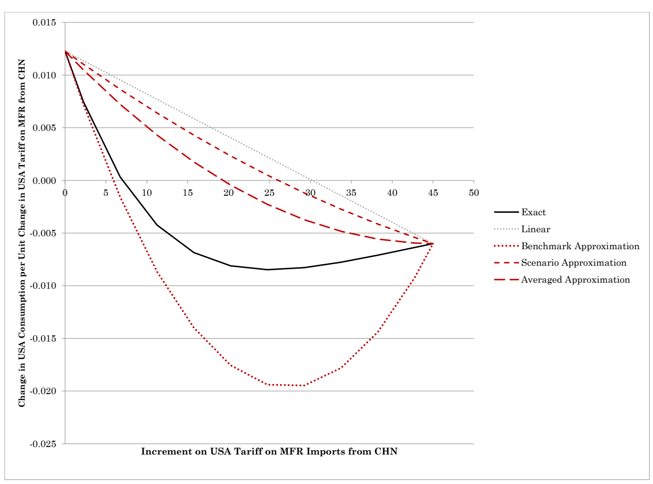

Figure 1.5.8: Welfare Change in US by US import tariff on Chinese Manufactures

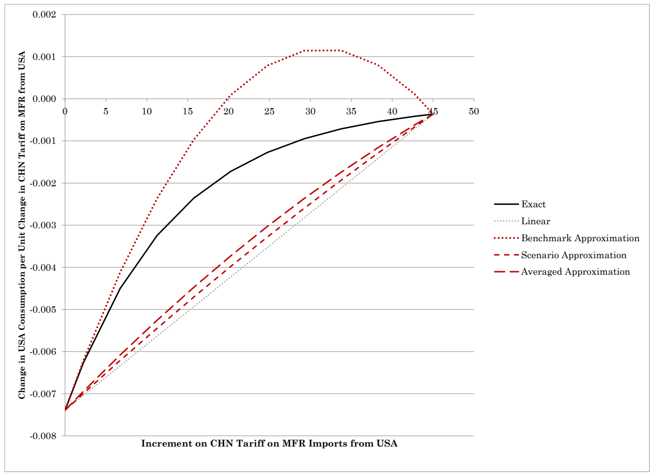

Figure 1.5.9: Welfare Change in US by China import tariff on US Manufactures

affected in both cases: either US raising tariffs on Chinese goods, or China implements its retaliatory measures against U.S..

1.6 Conclusion

In this paper, we show a modified Central Path Decomposition of Harrison et al. (2000) without numerical integration, and we apply that in a range of tariff interventions which have been discussed recently in Washington, DC. It is worth noting that although we only work on order-independent decomposition with very small number of instruments, in which case it seems that order-dependent approaches such as decomposition along edgewise paths works fine as well, imagining if we were discussing some trade agreements in the World Trade Organization (WTO) when all its members committed to reduce their trade barriers at the same time? Too many policy instruments makes order-dependent decomposition not feasible. Another related point is that when the model is large, it may appears that it costs `LSA` some time in reporting second order derivatives at solution points. However, in cases when CPD is the reasonable choice left for us to do welfare decomposition as in the WTO case above, we have to solve the model enough times to insure a precise decomposition. It is almost surely

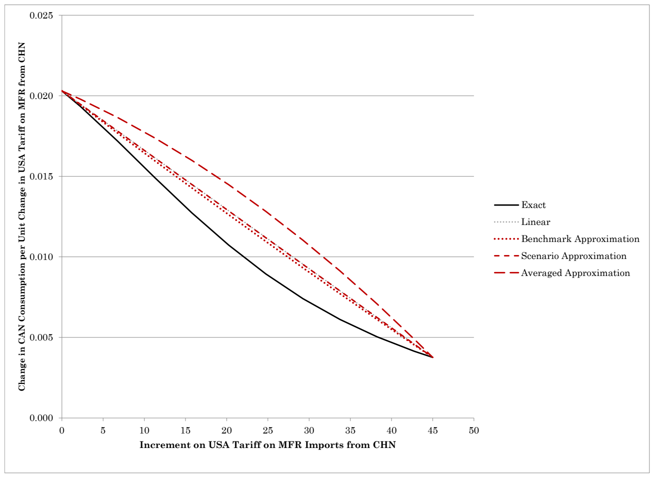

Figure 1.5.10: Welfare Change in Canada by US import tariff on Chinese Manufactures

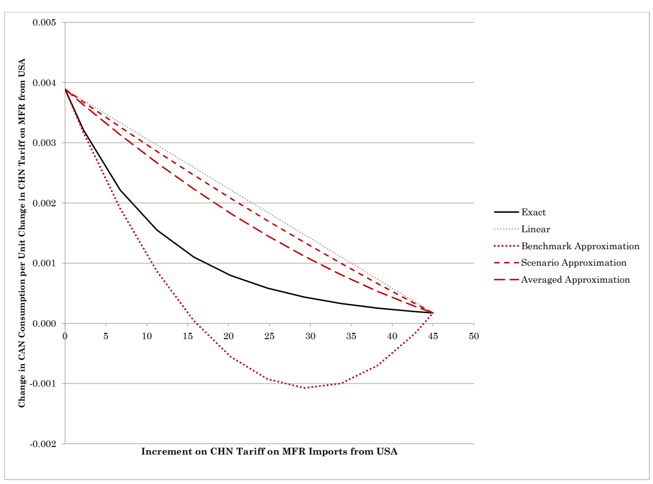

Figure 1.5.11: Welfare Change in Canada by China import tariff on US Manufactures

more expensive in calculation through numerical integration. After all, we only require two explicit solutions in our interpolation method.

Chapter 2

Maximum Likelihood and Least Squares Estimation: New Tools for Solving Econometrics Problems with GAMS

2.1 Introduction

GAMS, the "Generalized Algebraic Modeling System", is a modeling language which was originally developed for linear, nonlinear and integer programming. One of the main objectives of GAMS is to facilitate rapid prototyping of mathematical optimization and structural equilibrium models. Kalvelagen (2007a) points out that because GAMS is an excellent tool for applied optimization, it is well suited for econometric applications such as the Maximum Likelihood and Least Squares estimation, applications which are optimization problems. It has been less apparent, particularly for non-GAMS programmers, is that with a few new tools, GAMS is also able to perform all of the tasks involved in estimation and statistical testing. In this paper, we describe such tools and we illustrate how these methods permit the calculation of confidence intervals and test statistics for estimation problems formulated in GAMS.

Our exposition highlights the virtues of algebraic modeling languages for econometric model specification. A GAMS program provides a explicit framework for model specification. Furthermore, the logical steps in an estimation problem can be translated from `MATLAB/STATA/R` into GAMS, and GAMS then offers a level of transparency, even for readers who are not proficient in the programming rules. We show this by examples.

Together with the paper we provide a library of worked examples through which graduate students in applied economics can more easily solve several types of classical econometric problems. For professionals currently using `MATLAB/STATA/R`, we offer code fragments which cover most of the mundane tasks involved in implementing estimation models in GAMS: reading data from different sources, calling include library routines, even simple GAMS syntax for working with datasets.

As for the expected *audience* for this paper, we have three groups of people in mind: existing GAMS users who are interested in our tools for econometric analysis with GAMS; second, `MATLAB/STATA/R` programmers who may occasionally encounter estimation problems which are not readily solved with their standard tools. By providing cleanly documented "worked examples", we make it easy for non-GAMS programmers to use GAMS when needed. Third, we are writing for graduate students in applied economics who can benefit from the clarity and precision of using algebraic modeling tools for estimation. We are particularly thinking about students who want to do applied equilibrium analysis and would benefit from using a single programming framework for their thesis work.

The examples covered in this paper can be grouped into two broad categories: maximum likelihood (MLE) and nonlinear least squares (NLS). MLE models include a binomial *probit* model, a binomial *logit* model, a conditional *logit* model with multiple choice, an ordered *logit* with multiple choice. NLS model is about parameter estimation and statistic tests based on a CES production function. In order to solve a parametric estimation example as listed above, we set up a GAMS model as a Nonlinear Program (`NLP`), estimate the unknown parameters and report statistics for estimation and testing purposes. Before getting to the estimation, we begin with an introduction to GAMS library routines we have created to calculate estimation statistics.

The remainder of this paper is organized as follows: Section 2 introduce how to retrieve Jacobian/Hessian matrices. Section 3 presents MLE models and section 4 presents a NLS model and in section 5 we conclude.

2.2 Retrieving Indexed Jacobian and Hessian Matrices

Consider first a generic model based on a set of equations $eq(m)$, in which m is the (potentially multidimensional) domain of equations. The model involves variables $var(n)$ in which n is the domain. Let nn represent an alias for variable domain in GAMS. In order to compute estimation statistics, we need to have access to the Jacobian and Hessian matrices at the solution, $J(m, n)$ and $H(m, n, nn)$. For this purpose, we have produce a *local sensitivity analysis* library routine which retrieves submatrices of J and H.

`LSA` routine works with `NLP` and `MCP` models in GAMS. They are invoked using the conventional `$libinclude` syntax. Here is an example in which we solve an optimization model and then retrieve Jacobian/Hessian submatrices. Following the solve statement for the `NLP` model, the libinclude call LSA requests specific submatrices at the solution point. After returning from LSA, the matrices are read from a GDX file in the scratch directory:

```
solve <model_name> using nlp maximizing <obj>;
$libinclude lsadata <model_name> "using nlp maximizing <obj>"
"<eq>_<var>,<eq>_<var1>_<var2>"
execute_load '%gams.scrdir%lsadata.gdx',<Jacobian>=<eq>_<var>,
<Hessian>=<eq>_<var1>_<var2>;
```

In this pseudo-code obj is the objective value in the model, optimand in an `NLP` model takes the form of ”using nlp maximizing/minimizing obj”. For a mixed complementarity problem (`MCP`) model, we instead use the commands:

```
solve <model_name> using mcp;
$libinclude lsadata <model_name>
execute_load '%gams.scrdir%lsadata.gdx',<Jacobian>=<eq>_<var>,
<Hessian>=<eq>_<var1>_<var2>;
```

Here is an example of how Jacobian and Hessian matrices are retrieved following log-likelihood estimation in example application `beach`:

```
solve nc_beach maximizing LOGLIK using nlp;

parameter
        Jac(i,k)        Jacobian of constraints wrt unknowns,
        H(i,k,kk)        Hessian of constraints wrt unknowns;

$libinclude lsadata nc_beach "using nlp maximizing LOGLIK" "llf0_beta,llf0_beta_beta"
execute_load '%gams.scrdir%lsadata.gdx',Jac=llf0_beta, H=llf0_beta_beta;
```

Here is an example of how a Jacobian matrix is retrieved following solution of a system of equations (cast as a mixed complementarity problem) in application `milk`:

```
solve diff using mcp;

parameter
        grad(i)         Gradients retrieved in statistical testing;

$libinclude lsadata diff
execute_load '%gams.scrdir%lsadata.gdx',grad=def_diff_beta;
```

2.3 Maximum Likelihood Estimation (MLE)

Maximum Likelihood as an estimation strategy which is widely applied in econometrics, especially in models based on Limited Dependent Variables (LDV). In this type of model, a "limited" dependent variable y is one which takes a "limited" set of values. For example, a binary dependent variables mode in which dependent variable $y \in \{0, 1\}$, a multinomial model in which $y \in \{0, 1, 2, \dots, k\}$, a count data model whose dependent variables $y \in \{0, 1, 2, \dots\}$, and a model censored from below at 0 in which $y \in \{\mathbb{R}^+ \cup 0\}$.

2.3.1 Binary choice

We first introduce two binary choice models in *probit* and *logit* formulation. Both are regression models where dependent variable satisfies $y_t \in \{0, 1\}$ (which represents a Yes/No outcome) given explanatory variables x_t ($I \times 1$, where I is the total number of exogenous variables). To be more specific, both binary *probit* and *logit* describe the conditional distribution of $\Pr(y_t = 1|x_t)$, *i.e.*, the probability that an observation with particular characteristics will fall into a positive outcome. However, these two models take different distribution assumptions on error terms, which leads to different functional form assumptions of this conditional probability.

We assume for each observation t, the net utility gained U_t^*, which is not observable, and related to a set of exogenous variables x_t. After introducing coefficients, $\beta(i)$ and error terms μ_t, we establish the following latent model:

$$U_t^* = x_t'\beta + \mu_t, \tag{2.3.1}$$

This latent model is equivalent to the binomial *probit*/*probit* model written as:

$$y_t = x_t'\beta + \mu_t, \tag{2.3.2}$$

When the relationship between latent utility variable U_t^* and the observable response (0/1) variable, y_t, satisfies:

$$y_t = \begin{cases} 1 & \text{if} \quad U_t^* > 0 \\ 0 & \text{otherwise.} \end{cases}$$

To further develop this regression model, in binomial *probit*, we assume *i.i.d* normally distributed error terms μ_t, and the conditional probability $\Pr(y_t = 1|x_t)$ takes the normal form:

$$\Pr(y_t = 1|x_t) = \Phi(x_t'\beta), \tag{2.3.3}$$

where $\Phi(\cdot)$ is the standard normal Cumulative Distribution Function (CDF).

On the other hand, in the binomial *logit* model, we assume that error terms μ_t follow an *i.i.d.* logistic distribution, and the conditional probability takes the logistic form:

$$\Pr(y_t = 1|x_t) = \frac{exp(x_t'\beta)}{1 + exp(x_t'\beta).} \tag{2.3.4}$$

Based on these functional form of conditional probabilities, standard statistical textbook such as Greene (2011) show estimator $\hat{\beta}$ in binomial *probit* through maximizing the following log-likelihood function: $\ln \mathcal{L}(\beta)$:

$$\hat{\beta} = \arg\max_{\beta} [\ln \mathcal{L}(\beta)] = \arg\max_{\beta} \left[\sum_t \left(y_t \ln \Phi(x_t'\beta) + (1 - y_t) \ln \left(1 - \Phi(x_t'\beta)\right)\right) \right], \tag{2.3.5}$$

while estimator in binomial *logit* is calculated as:

$$\hat{\beta} = \arg\max_{\beta} [\ln \mathcal{L}(\beta)] = \arg\max_{\beta} \left[\sum_t \left(y_t \ln \left(\frac{\exp(x_t'\beta)}{1 + \exp(x_t'\beta)} \right) + (1 - y_t) \ln \left(\frac{1}{1 + \exp(x_t'\beta)} \right) \right) \right]. \tag{2.3.6}$$

In order to report standard regression statistics such as t-statistic, p-value etc., we estimate the co-variance matrix of the estimator $\hat{\beta}$, i.e., $\hat{V}_{\hat{\beta}}$. This is based on the inverted Hessian matrix (see, e.g., Greene (2011)):

$$\hat{V}_{\hat{\beta}} = -(\hat{H}_{\hat{\beta}})^{-1}, \tag{2.3.7}$$

where $\hat{H}_{\hat{\beta}} = \nabla^2 \ln \mathcal{L}(\beta)_{|\hat{\beta}}$ is the estimated Hessian of the log-likelihood function $\ln \mathcal{L}(\beta)$ at the solution point $\hat{\beta}$.

2.3.1.1 Binomial Probit (`milk`)

A biannual survey of households the Encuesta Nacional de Ingresos y Gastos de los Hogares(ENIGH) undertaken by The national institute of statistics in Mexico has been used by many academics to study income inequality, returns to education, gender based income differences, consumer expenditure patterns and many other issues. We use this data to examine purchases of fluid milk by Mexican households as there is continuing concern as to the lack of an adequate intake of calcium especially by children. We motivate a *probit* model of fluid milk purchases with the following variable list (2.1):

Variables	Description	Units
fluidx	Expenditures on pasteurized fluid milk	Pesos
	Household Composition Variables	
perlt6	Percent of household members <6 years old	%
per6_11	Percent of household members between 6 and 11 years old	%
perge66	Percent of household members older than 65	%
num_yung	Number of household members <12 years of age	#
	Other Household Characteristics	
sm_city	Household is located in a town with 2,500-15,000 population	0/1
city_11	Household is located in a town with $>15,000$ population	0/1
rural	Household is located in a town with $<2,500$ population	0/1
refrig	Does the household own a refrigerator/freezer?	0/1
incomet	Quarterly household income	10,000 Pesos
perfafh	Percent of weekly household food expenditures spent on food purchased and consumed outside the home	%
regdf	Household located in the Federal District region of Mexico	0/1

Table 2.1: Description of a Subset of Variables in 2002 ENIGH Dataset

We first estimate the *probit* model, test the significance of exogenous variables in part (a) and part (b); Define and test the income elasticity in part (c); Then estimate and test the marginal effect of having a refrigerated space on milk purchasing probability in part (d).

For part (a), we estimate the *probit* model represented in equation (2.3.8) with the general formulation 2.3.5 in mind:

$$\Pr(fluidx > 0) = F(Constant, num\_yung, incomet, num\_yung{*}incomet, sm\_city, city, refrig, perfafh) \tag{2.3.8}$$

We first solve the estimation problem using MLE of a *probit* model, after that, we compare the model in equation (2.3.8) with a restricted model when only the constant term is included in the model to check whether such a "naive" model is enough to explain the pattern of observed values of the dependent variable.

According to *probit* formulation 2.3.5 in binary choice case, we set up log-likelihood as follows. Here y(t) is a (0/1) variable indicates "purchases of fluid milk" or not in time period t, x(t,i) are explanatory variable i in time period t. Detailed codes are included in Appendix:

```
*       Part (a)
*       Probit Model of Fluid Milk Purchases
variable        BETA(i)         Unknowns to be estimated,
                LOGLIK          Log-Likelihood;

equation        obj             Definition of log-likelihood for probit model;

obj..   LOGLIK =e= sum(t,   y(t) *log(    errorf(sum(i, x(t,i)*BETA(i))) +
                         (1-y(t))*log(1 - errorf(sum(i, x(t,i)*BETA(i)))));

BETA.LO(i) = -100;
BETA.UP(i) = 100;

model milk_probit /obj/;
solve milk_probit maximizing LOGLIK using nlp;
```

Maximize log-likelihood function LOGLIK leads to coefficients estimation $\widehat{\beta(i)}$. We then derive Hessian of the log-likelihood with respect to model coefficients at this point.

```
parameter       H(i,ii)         Hessian of log-likelihood wrt coefficients,
                ident(i,ii)     Identity matrix,
                cov(k,i,ii)     Covariance matrix of estimators in model k,
                cov_(i,ii)      Covariance matrix retrieved from lsa,
                stat(k,i,*)     Statistics reported in model k;


*        Retrieve the Hessian:
$libinclude lsadata milk_probit "using nlp maximizing LOGLIK" "obj_beta_beta"
execute_load '%gams.scrdir%lsadata.gdx',H=obj_beta_beta;


*        Convert the Hessian to a symmetric matrix:
H(i,ii) = H(i,ii) + H(ii,i)$(not sameas(i,ii));
```

After that, according to co-variance estimation formulation in MLE (2.3.7), we find inverse of the Hessian as the estimated variance matrix before report regression statistics on the full model with all explanatory variable x(t,i).

```
*        Find inverse of the Hessian
ident(i,i)$sum(ii$H(i,ii),1) = 1;
$libinclude lufactorsolve H ident cov_


*       Record estimated covariance of the unrestricted probit model
cov("ur",i,ii) = cov_(i,ii);


*       Report statistics for the unrestricted model
stat("ur", i, "estimator")  = BETA.L(i);
```

```
stat("ur", i, "std error")  = sqrt(cov("ur",i,i));

stat("ur", i, "t value")$cov("ur",i,i) = BETA.L(i)/sqrt(cov("ur",i,i));


*       Use the BETAREG function:

stat("ur", i, "P value")$cov("ur",i,i)

               = BETAREG( (card(t) - card(i))/

                          (card(t) - card(i) + sqr(stat("ur", i, "t value"))),

                          (card(t) - card(i))/2, 0.5 );
```

To check the significance of exogenous variables in this model, we repeat the estimation procedure on a model with only constant (unity) explanatory variable (x(t,"cons") = 1, 0 for all other variable i) and report the following statistics from the original model (called unrestricted model) and the restricted model:

```
----    413 PARAMETER stat  Statistics reported in model k
```

	estimator	std error	t value	P value
ur.cons	-0.919	0.035	-26.314	
ur.num_yung	-0.068	0.016	-4.362	1.2E-5
ur.incomet	0.186	0.018	10.221	
ur.num_yung_incomet	0.048	0.010	4.676	2.9E-6
ur.sm_city	0.285	0.037	7.598	
ur.city	0.697	0.028	25.283	
ur.refrig	0.558	0.027	20.376	
ur.perfafh	-0.448	0.057	-7.908	
r .cons	0.139	0.010	13.382	

We do hypothesis testing in this binomial *probit* model using the *Likelihood Ratio* (LR) test. The LR test is based on the difference in the log-likelihood functions for the unrestricted and restricted models. In general, the larger the fall of the log-likelihood (from the value of the unrestricted case to that of the restricted case), the more likely we want to reject the null hypothesis.

Let $\mathcal{L}_{ur}$ ($\mathcal{L}_{res}$) denote the maximized log-likelihood value for the unrestricted (restricted) model. Then the likelihood ratio statistic follows a *Chi-square* distribution

$$LR = 2(\mathcal{L}_{ur} - \mathcal{L}_{res}) \xrightarrow{\text{d}} \chi^2_q,$$

where q is the number of restrictions in a hypothesis.

```
*       Part (b):
*       Likelihood ratio test against H0: Restricted model is true
*       Find LR test statistic and p value based on H0


*       Statistical Testing
set     m       Test type
                /lr                     Likelihood ratio test,
                 z_ie                   Z test on income elasticity wrt unknowns,
                 z_diff                 Z test on probability difference wrt unknowns/;


parameter       alpha                   Type 1 error level   /0.05/,
                grad(i)                 Gradients retrieved in statistical testing,
                pval(m)                 P value of test type m,
                test_stat(m)            Test statistic based on H0 in test m;


test_stat("lr") = 2 * (ll("ur") - ll("r"));


*       Use the gammareg function
*       in with degrees of freedom of the LR test: card(i) - card(cn)
pval("lr") = 1 - gammareg(test_stat("lr")/2, (card(i) - card(cn))/2);
```

We construct a report of test results for all statistical tests performed in this example to avoid replicated work.

```
*       Make a report of test results
$onecho > %gams.scrdir%report.gms


parameter       test(m,*,*)             Test Results;
loop(m$(pval(m) lt alpha),
        test(m, "Reject H_0", "P value")   = pval(m);
        test(m, "Reject H_0", "Test Stat") = test_stat(m););


loop(m$(pval(m) ge alpha),
        test(m, "Fail to reject H_0", "P value")   = pval(m);
        test(m, "Fail to reject H_0", "Test Stat") = test_stat(m););


option test:3:2:1;


display "Test Report:", test;
$offecho


$include "%gams.scrdir%report"
```

And *LR* test shows that it is statistically safe to reject the null hypothesis that naive model is true.

(c) At the mean of the data, what is the income elasticity of the probability of purchasing fluid milk? From a statistical perspective is this elasticity significant?

We evaluate income elasticity of the milk purchasing probability at the mean of the data, then apply *z-test* to check whether it is significantly different from zero. A *z-test* assumes the distribution of the test statistic under the null hypothesis can be approximated by a normal distribution.

```
*       Part(c):
*       Z test on income elasticity wrt unknowns with
*       H0: Income elasticity = 0
parameter       m_x(i)          Mean of the data;
m_x(i) = sum(t, x(t,i))/card(t);

variable        I_ELS           Income elasticity evaluated at the mean of the data;
equation        def_ie          Definition of income elasticity;

$macro  norm(i) ( 1/sqrt(2*pi)*exp(-0.5*(sqr(sum(i, m_x(i)*BETA(i)))))\
                   /errorf(sum(i, m_x(i)*BETA(i))) )
def_ie..
   I_ELS =e=
    (BETA("incomet") + BETA("num_yung_incomet")*m_x("num_yung"))*m_x("incomet")*norm(i);

model ie /def_ie/;
BETA.FX(i) = b("ur",i);
solve ie using mcp;

*       Retrieve the gradients using lsadata:
$libinclude lsadata ie
execute_load '%gams.scrdir%lsadata.gdx',grad=def_ie_beta;

*       Find Z statistics against H0:
test_stat("z_ie") = I_ELS.L / sqrt(sum((i,ii), grad(i)*cov("ur",i,ii)*grad(ii)));
pval("z_ie")      = 1 - errorf(abs(test_stat("z_ie")));

$include "%gams.scrdir%report"
```

According to the *p-value*, we reject the null hypothesis claiming that the income elasticity of purchasing probability is statistically zero.

(d) Based on results obtained from estimating equation (2.3.8) what is the impact of having refrigerated storage in the household on the probability of purchasing fluid milk? Is this impact positive from a statistical perspective?

First of all, since *probit* model is nonlinear, β_{refrig} is not the coefficient of impact we are looking for. Analytically, marginal effects in changes in explanatory variable j in a *probit* model is

$$\frac{\partial \Pr(Y_t = 1|x_t)}{\partial x_{t,j}} = \beta_j \phi(x_t' \beta),$$

according to the *probit* model (2.3.3), where ϕ is the probability density function of a standard normal distribution.

In the following code, we address the marginal effects of having a refrigerated storage or not on the milk purchasing probability. First, dividing the data into two groups, one having a refrigerated storage and the other not, we then find the difference in probabilities of milk purchase between these two groups, a measurement of the marginal effect of having a refrigerated space. After that, we run a *z-test* to check whether this impact is significant or not.

```
*       Part(d):
*       Z test on probability difference wrt unknowns with
*       H0: Having refigerated storage does not have significant impact

parameter
        mx_ref(i)       Mean of x if household owning a refrig or freezer,
        mx_noref(i)     Mean of x if household not owning a refrig or freezer;

*       Generate dummy variable of having refrigerated storage or not
mx_ref(i) = m_x(i);
```

```
mx_ref("refrig") = 1;

mx_noref(i) = m_x(i);
mx_noref("refrig") = 0;

*       Define the probability difference in purchasing milk
*       caused by having refigerated storage or not
variable
        P_DIFF    Difference in milk purchasing probabilities with having refrig or not;
equation
        def_diff  Definition of the difference in probabilities;
def_diff..
        P_DIFF =e= errorf(sum(i, mx_ref(i)*BETA(i))) - errorf(sum(i, mx_noref(i)*BETA(i)));

model diff /def_diff/;
solve diff using mcp;

*        Retrieve gradients using lsadata:
$libinclude lsadata diff
execute_load '%gams.scrdir%lsadata.gdx',grad=def_diff_beta;

*       z test
*       Find Z statistics

test_stat("z_diff") = P_DIFF.L / sqrt(sum((i,ii), grad(i) * cov("ur",i,ii) * grad(ii)));
pval("z_diff")      = 1 - errorf(abs(test_stat("z_diff")));

$include "%gams.scrdir%report"
```

According to the *p-value* of a *z-test*, we reject the null hypothesis that having refigerated storage has zero impact on milk purchasing probability. Here is a summary of all three statistical tests whose *p-value* all close to zero:

```
----    735 Test Report:
----    735 PARAMETER test  Test Results
                        Test Stat
lr    .Reject H_0    2833.535
z_ie  .Reject H_0      17.066
z_diff.Reject H_0      20.851
```

2.3.2 Binomial Logit (`credit`)

Greene (1992) estimated a model of consumer behavior where he examines whether or not an individual has experienced a major negative derogatory report in his/her credit history. The data file contains information on the credit history of a sample of more than 1,000 individuals. The variables contained in this dataset include among others are the following:

Variable	Description
Majordrg	Number of major derogatory credit reports
Age	Age in yeas plus
Inc_per	Per Capita yearly income (divided by $10,000)
Avgexp	Average monthly credit card expenditures
Ownrent	person owns a home (=1)
Active	Number of active credit accounts

Table 2.2: Description of Variables in credit data file

The following discrete choice model is set up to examine the determinants of whether a credit card holder experiences a derogatory credit report:

$$\Pr(Majordrg > 0) = g(Age, Age\_sqr, Inc\_Per, Avgexp, Ownrent),$$

We then estimate the above model using the *logit* specification based on the characteristics of data before reporting typical statistics; After that, we calculate the elasticity impacts of a change in Age and Income on the probability of having a major derogatory report, then test whether these impacts are statistically 0; Finally in the last part, we test the above income elasticity is significantly different when we evaluate them at different income levels.

We start with the *logit* estimation: first introduce dependent variable y(t) and explanatory data x(t,i). We set up log-likelihood as model objective according to 2.3.6 (detailed codes are included in Appendix A):

```
*       Part(a)
*       Estimate logit model through loglikelihood maximization
variable        BETA(i)         Unknowns to be estimated,
                LOGLIK          Loglikelihood;

equation        obj             Objective for a unrestricted model;

obj..        LOGLIK =e= sum(a(t),
               y(t)*log(exp(sum(i, x(t,i)*BETA(i)))/(1 + exp(sum(i, x(t,i)*BETA(i))))) +
               (1-y(t))*log(1/(1 + exp(sum(i,x(t,i)*BETA(i))))));

model credit_logit  /obj/;
solve credit_logit maximizing LOGLIK using nlp;
```

Similar to the previous *probit* example, we first trigger lsadata file in retrieving Hessian H(i,ii):

```
*        Retrieve the Hessian matrix:
```

```
$libinclude lsadata credit_logit "using nlp maximizing LOGLIK" "obj_beta_beta"
execute_load '%gams.scrdir%lsadata.gdx',H=obj_beta_beta;
```

Then inverse this matrix to estimate the variance cov_(i,ii).

```
*        Convert the Hessian to a symmetric matrix:
H(i,ii) = H(i,ii) + H(ii,i)$(not sameas(i,ii));


*        Find inverse of the Hessian
ident(i,i)$sum(ii$H(i,ii),1) = 1;
$libinclude lufactorsolve H ident cov_
```

Finally, regression statistics report of this binomial *logit* model is built upon coeffient and variance estimates:

```
----    413 PARAMETER stat  Statistics reported in model k
             estimator   std error   t value     P value
cons         -3.923       0.826      -4.747       2.3E-6
age           0.146       0.044       3.286       0.001
age_sqr      -0.002      5.7E-4      -2.819       0.005
inc_per       0.098       0.055       1.776       0.076
avgexp       -0.001      3.8E-4      -3.585       3.5E-4
ownrent      -0.435       0.158      -2.757       0.006
```

(b) Based on the estimated Logit model, what are the elasticity impacts of a change in Age and Inc_per (individually) on the probability of having a major derogatory report? Are these effects different from 0 at the mean of the data?

In this part, we rely again on a *z-test* to check whether impacts of a change in Age and Income on the probability of having a major derogatory report are 0 or not. Both model set-up and estimation procedure are similar to *z-test* we preform in the *probit* example, therefore we skip the details and leave the codes in Appendix A. Test statistics confirm that it is safe to reject the null in both tests: neither the Age or Income elasticity of having major credit reports is statistically close to 0.

```
----    709 PARAMETER test  Test Results

                      P value     Test Stat

z_age.Reject H_0      5.0E-5       3.889

z_inc.Reject H_0       0.038       1.772
```

2.3.3 Multiple Choice Models

We turn now to examples of *logit* and *probit* models based on multiple choice data. Estimation of these models involves evaluating integrals of the normal distribution in some cases, and the *logit* model becomes more popular than the *probit* when dealing with multiple choice data. However, for those who has the needs, we put some codes in the appendix to show how to use GAMS in simulating multivariate normal CDF (Geweke, Hajivassiliou, Keane (GHK) simulator) to make the multiple integrals possible.

In multiple choice models with more than two alternatives, if explanatory variables contain only individual characteristics, the *multinomial logit* model is defined as

$$\Pr(y_i = j|x_i) = P_{ij}^m = \frac{\exp(x_i'\beta_j)}{\sum_{h=1}^{J}\exp(x_t'\beta_h)},$$

for $j, h = 1, 2, \ldots, J$, where y_i is the dependent variable that indicates the choice made, x_i is a vector of characteristics specific to the i^{th} individual, and β_j is a vector of coefficients specific to the j^{th} alternative. Thus, this model involves choice-specific coefficients and only individual specific regressors. For model identification, one often assumes that $\beta_1 = 0$. The log-likelihood function of the *multinomial logit* is

$$\ln \mathcal{L}(\beta) = \sum_{i=1}^{I}\sum_{j=1}^{J} d_{ij} \ln P_{ij}^m,$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual i chooses alternative j} \\ 0 & \text{otherwise,} \end{cases}$$

A closely related technique called *conditional logit*, developed by McFadden (1973), which is appropriate when choice-specific data are available. Assuming type I extreme-value distributed errors, the probability that individual i chooses alternative j from his choice set C_i is

$$\Pr(y_i = j|x_{ij}) = P^c_{ij} = \frac{\exp(x'_{ij}\beta)}{\sum_{h\in C_i}^{J}\exp(x'_{ih}\beta)},$$

where x_{ij} is a vector of attributes specific to the j^{th} alternative as perceived by the i^{th} individual. The log-likelihood function of the *conditional logit* is

$$\ln\mathcal{L}(\beta) = \sum_{i=1}^{I}\sum_{j\in C_i}^{J} d_{ij}\ln P^c_{ij},$$

Besides exogenous data characteristics, one obvious difference between *multinomial logit* and *conditional logit* is the former estimates $J-1$ sets of coefficients β_j (for $j = 2, 3, \ldots, J$), while the latter only estimates a single set β, which includes a coefficient for each explanatory variable. In next two examples, the dependent variable could either be unordered (`beach`) or ordered (`fish`). We start with the unordered, *conditional logit* problem as follows:

2.3.3.1 Multiple Choice in Conditional Logit (`beach`)

In this example, data file contains variables constructed using the southeastern NC beach visitation data. The matrices are defined as follows:

Variables	Description
income	Annual income for each respondent
price	Travel cost in dollars for each respondent to each of the beaches
trip	Number of times each respondent visited each beach
parking	Number of parking spots available at each beach (constant across people, 100's of spots)
miles	Length in miles of the beach(constant across people)

width	Width in feet of the beach (constant across people)

Table 2.3: Description of variables in the data file: beach

First, we observe that we have choice-specific explanatory variables, which facilitates application of the *conditional logit* model.

(a) Estimate a choice model including price, parking, miles, and width as explanatory variables. Report estimates and t-statistics for this model, using both inverse Hessian and robust standard errors.

For explanatory variable list k0 = price, parking, miles, width, we set up a *conditional logit* model with respondents i and choice option j as follows:

```
$title Conditional Logit Model in multiple choice model: NC Beach Visits
*       Beach choices data
parameter       income(i)       Annual income for each respondent (divided by 10000),
                trip(i,j)       Number of times each respondent visited each beach,
                data(i,j,*)     Explanatory variables of respondent i who chooses j;


*       Generate the term miles_inc
data(i,j,"miles_inc") = data(i,j,"miles") * income(i) ;


*       Part (a)
*       Estimate a baseline choice model including parking, price, miles and width
*       as explanatory variables
variable        BETA(k)         Coefficients of explanatory variables,
                LOGLIK          Sum of log-likelihood,
                LL(i)           Definition of log-likelihood for respondant n;


equation        obj             Sum of log-likelihood functions,
```

```
                llf0(i)         Log-likelihood for the baseline model;


$macro          agg(i,j)        ( exp(sum(k0, data(i,j,k0)*BETA(k0))) )


llf0(i)..       LL(i) =e=
                   sum(j, trip(i,j)*log(agg(i,j))) - sum(j,trip(i,j))*log(sum(j, agg(i,j)));
obj..           LOGLIK =e= sum(i, LL(i));


model nc_beach /obj, llf0/;
solve nc_beach maximizing LOGLIK using nlp;
```

Next, we retrieve Jacobian and Hessian at the estimated point using `lsadata`:

```
*       Define parameters
parameter       Jac(i,k)                Jacobian of constraints wrt unknowns,
                H(i,k,kk)               Hessian of constraints wrt unknowns;


*        Retrieve the Jacobian and Hessian:
$libinclude lsadata nc_beach "using nlp maximizing LOGLIK" "llf0_beta,llf0_beta_beta"
execute_load '%gams.scrdir%lsadata.gdx',Jac=llf0_beta, H=llf0_beta_beta;
```

From reported statistics, we find that adopting "Robust Standard Errors" method leads to significantly smaller *t value* than the "Inverted Hessian" method.

```
----    579 PARAMETER statistics  Statistics at the point
                                  estimator   std error     T value
inverse Hessian        .price        -0.084       0.001     -74.472
inverse Hessian        .miles         0.050       0.004      12.436
inverse Hessian        .width       7.1E-4       1.7E-4     4.221
inverse Hessian        .parking       0.075       0.002      31.969
robust estimate for cov.price        -0.084       0.010      -8.583
```

```
robust estimate for cov.miles          0.050        0.030        1.677
robust estimate for cov.width         7.1E-4       8.9E-4        0.829
robust estimate for cov.parking        0.075        0.012        6.097
```

(b) Estimate a second model parameterized to test the following notion: higher income beach goers care more about the length of the beach when making site choice decisions, all else equal. Report estimates and t-statistics (just use the inverse Hessian from here forward) for the model you use and briefly interpret your findings.

In this part, we estimate a choice model including income and miles product in the baseline model as an explanatory variable, all others stay the same. As a result, we redefine the log-likelihood function based on variable set k1 which includes a new variable miles_inc.

```
*       Part (b)
*       Estimate a choice model including income and miles product
*       in the baseline model as an explanatory variable
*       Test:
*       Is it true that higher income beach goers care more
*       about the length of the beach when making site choice decisions, all else equal.
equation        llf1(i)         Log-likelihood in model with miles and income cross;

$macro          agg1(i,j)       ( exp(sum(k1, data(i,j,k1)*BETA(k1))) )

llf1(i)..     LL(i) =e=
                sum(j, trip(i,j)*log(agg1(i,j))) - sum(j,trip(i,j))*log(sum(j, agg1(i,j)));

model nc_beach1 /obj, llf1/;
solve nc_beach1 maximizing LOGLIK using nlp;
```

We find that once we consider the term miles_inc, the coefficient on miles becomes very insignificant. It seems to support the claim that higher income beach goers care more about the length of the beach when making site choice decisions, all else equal.

```
----    811 PARAMETER statistics  Statistics at the point
                    estimator   std error     t value     P value
H_inv.price            -0.084       0.001     -74.308
H_inv.miles            -0.002       0.009      -0.217       0.828
H_inv.width            6.9E-4      1.7E-4       4.094      4.8E-5
H_inv.parking           0.076       0.002      32.216    2.7E-136
H_inv.miles_inc         0.009       0.001       6.323
```

(c) If you summarize the site-choice frequencies in the sample you'll see that site 9 (Wrightsville Beach) has 19% of the visits. We might want to include an alternative specific constant for site 9 because of this relatively high frequency. Consider again the specification in (a), but this time include an alternative specific constant for Wrightsville. Interpret your estimates, paying particular attention to the constant for site 9 and how estimates for the other coefficients change.

In this part, we replace variable miles_inc with a new one called site9 which indicates the choice of beach site 9 in the variable set based on the model of the last part.

```
*       Part (c)
*       Estimate a choice model including site9 in the baseline model
*       as an explanatory variable

*       In the sample we find site 9 (Wrightsville Beach) has 19% of the visits.
*       Thus, we include an alternative specific constant (ASC) for site 9
*       because of this relatively high frequency.
equation        llf2(i)         Log-likelihood definition for each respondant;
$macro          agg2(i,j)       ( exp(sum(k2, data(i,j,k2)*BETA(k2))) )

llf2(i)..    LL(i) =e=
```

```
        sum(j, trip(i,j)*log(agg2(i,j))) - sum(j,trip(i,j))*log(sum(j, agg2(i,j)));

BETA.L("miles_inc") = 0;
model nc_beach2 /obj, llf2/;
solve nc_beach2 maximizing LOGLIK using nlp;
```

Comparing the following statistics to the baseline model, we find in a model with variable site 9, the coefficient of miles goes up and that of parking goes down. This result is understandable, since we include a beach site which is relatively low in length (below average) and abundant in parking spots (ranked 17/17 in ascending order) explicitly in the model.

```
----   1165 PARAMETER statistics  Statistics at the point
                 estimator   std error     t value     P value
H_inv.price        -0.086       0.001     -74.496
H_inv.miles         0.021       0.005       4.439     1.1E-5
H_inv.width        8.0E-4     1.68E-4       4.777     2.2E-6
H_inv.parking       0.135       0.006      24.301     4.9E-93
H_inv.site9        -0.783       0.065     -11.958
```

2.3.3.2 Multiple Choice Model in Probit (fish)

Next, we introduce another multiple choice example in ordered *probit* formulation. The motivation of an ordered model is clear: for example, the management of the California Charter Boat Association points out that four fishing modes (beach, pier, private boat and charter boat) could be ranked from a fishing quality perspective with beach being the least productive and charter boat being the most productive (the rankings are shown in the following table). Thus, the estimation methodology should take into account this exogenous fishing mode ordering. As such, we decide to estimate an ordered *probit* model of fishing mode choice. Table 2.4 provides a summary of the variables contained in this data set that you would like to use in the *probit* model.

Variables	Description	Units
	Identification of type of fishing mode actually chosen (dependent variable):	
choice	1 = Beach 2 = Pier 3 = Private Boat 4 = Charter Boat	#
day_cost	Daily cost of fishing for mode actually used	$100
ctchrate	Hourly fishing catch rate for mode actually used	#
income	Monthly income	$10,000

Table 2.4: Description of variables in the data file

(a) Using the above data, estimate an ordered *probit* model of fishing mode choice which can be represented by the following latent variable regression:

$$\text{Choice}_i^* = x_i\beta + \epsilon_t, \tag{2.3.9}$$

where $\epsilon_t \sim N(0,1)$. Here $Choice_i^*$ indicates the net utility obtained from using a particular fishing mode. The relationship between the above latent variable and observed fishing modes (Choice) can be represented via the following:

$$\text{Choice}_i = \begin{cases} \text{Beach} & \text{if Choice}_i^* \leq 0 \\ \text{Pier} & \text{if } 0 < \text{Choice}_i^* \leq \mu_p \\ \text{Private Boat} & \text{if } \mu_p < \text{Choice}_i^* \leq \mu_{pb} \\ \text{Charter Boat} & \text{if } \mu_{pb} < \text{Choice}_i^* \ , \end{cases}$$

The exogenous variable matrix X is composed of the following exogenous variables: $Cons$,Day_Cost, $CtchRate$ and $Mnth\_Inc$. Present the typical regression results not only in terms of the estimated β coefficients but also with respect to the estimated μ_js, here μ_j is a threshold parameter vector that should be estimated along with coefficients vector β.

In this ordered *probit* model, given equation (2.3.9), and the normally distributed error terms, we have ordered *probit* model defined as

$$P_{ij} = \Pr(y_i = j|x) = \Pr(choice_i^*|x),$$

for $j = 1, \ldots, J$, y_i is the dependent variable indicates choice made, and β is a vector of coefficients for explanatory variables. We thus have conditional probability when individual i chooses beach mode ($j = 1$):

$$P_{i1} = \Pr(\epsilon_i \leq -x\beta|x) = \Phi(-x\beta|x),$$

where $\Phi(\cdot)$ is the CDF of a standard Normal distribution. Similarly, when other fishing modes are chosen, conditional probabilities are:

$$\begin{aligned}
P_{i2} &= \Phi(\mu_p - x\beta|x) - \Phi(-x\beta|x), \\
P_{i3} &= \Phi(\mu_{pb} - x\beta|x) - \Phi(\mu_p - x\beta|x), \\
P_{i4} &= 1 - \Phi(\mu_{pb} - x\beta|x);
\end{aligned}$$

Notice that the marginal effect of a change in X_i not β_i, for instance,

$$\frac{\partial P_{i1}}{\partial x} = -\phi(-x\beta)\beta,$$

where $\phi(\cdot)$ is the density function of a standard Normal distribution. And We have this model's log-likelihood defined as follows without multiple integration since alternatives are ordered:

$$L = \sum_{i=1}^{I}\sum_{j=1}^{J} d_{ij} \ln P_{ij},$$

where dummy variable

$$d_{ij} = \begin{cases} 1 & \text{if individual i chooses alternative j} \\ 0 & \text{otherwise.} \end{cases}$$

```
$title GAMS Solution of an Ordered Probit Model with Fishing Mode Choice Data

set     i       Observations            /1*1182/,

        j       Fishing modes

                  /beach          Fishing from beachs,

                   pier           Fishing from piers,

                   p_boat         Fishing on private boats,

                   c_boat         Fishing on charter boats/,

        k       Coefficients to be estimated

                  /cons           Constant term (unity),

                   day_cost       Daily cost of fishing,

                   ctchrate       Hourly fishing catch rate,

                   mnth_inc       Monthly income of fishermen,

                   mu_p           Threshold value of utility obtained from the pier mode,

                   mu_pb          Threshold value of utility obtained from the p_boat mode/;


alias (k,kk);


parameter       data(i,*)       Source data;


$call gdxxrw fish_ordered.xlsx par=data rng=A1:Q1183 cdim=1 rdim=1 checkDate

$gdxin fish_ordered.gdx

$loaddc data

$gdxin
```

```
parameter       y(i)            Indentification of chosen fishing mode,
                x(i,k)          Explanatory variables,
                dum(i,j)        Dummy variable of fishing mode choice (=1);

*       Y(t): takes value of 1 to 4 corresponding to each fishing mode
y(i) = data(i,"choice");

*       Exlanatory variables
x(i,k)      = data(i,k);
x(i,"cons") = 1;

*       Dummy variable = 1 if fishing mode k is chosen
loop( (i,j),
            dum(i,j) = 1$(y(i) = ord(j)) + 0$(not (y(i) = ord(j))); );

*       Ordered probit model of fishing mode choice
*       Through maximum likelihood estimation

variable        LOGLIK          Loglikelihood function for the probit model,
                COEF(k)         Coefficients to be estimated;

equation        obj             Objective in the maximum likelihood estimation,
                con_mu          Constraint that mu_pb > mu_p;

*       Mu is the threshold value of the unobserved net utility
*       obtained from using a particular fishing mode, choice_star,

*       For example, when choice_star < 0, choose "beach",
```

```
*       when 0<= choice_star < mu_p, choose "pier",
*       when mu_p < choice_star <= mu_pb,  choose "private boat",
*       when choice_star > mu_pb, choose "charter boat".

$macro log_b(i,k)  ( log(errorf(- sum(k,x(i,k) * COEF(k)))) )
$macro log_p(i,k)  ( log(errorf(COEF("mu_p")- sum(k,x(i,k)*COEF(k))) -\
                      errorf(- sum(k,x(i,k)*COEF(k)))) )
$macro log_pb(i,k) ( log(errorf(COEF("mu_pb")-sum(k,x(i,k)*COEF(k))) -\
                      errorf(COEF("mu_p") - sum(k,x(i,k)*COEF(k)))) )
$macro log_cb(i,k) ( log(1 - errorf(COEF("mu_pb") - sum(k,x(i,k) * COEF(k)))) )

obj..   LOGLIK =e= sum(i, dum(i,"beach") *log_b(i,k)  + dum(i,"pier")  *log_p(i,k) +
                               dum(i,"p_boat")*log_pb(i,k) + dum(i,"c_boat")*log_cb(i,k));

con_mu..     COEF("mu_pb") =g= COEF("mu_p") + 0.01;

*       Set bounds for coefficients

COEF.LO("mu_p")  = 0.01;
COEF.LO("mu_pb") = 0.02;
COEF.UP("mu_p")  = inf;
COEF.UP("mu_pb") = inf;

*       Assign initial values for mu
COEF.L("mu_p")  = 1;
COEF.L("mu_pb") = 2;

model ordered_probit /obj, con_mu/;
```

```
solve ordered_probit maximizing LOGLIK using nlp;

*       Report regression statistics
parameter       H(k,kk)         Hessian of log-likelihood wrt coefficients,
                ident(k,kk)     Identity matrix,
                cov(k,kk)       Covariance matrix of estimators,
                stat(k,*)       Statistics reported;

*       Report Hessian matrix
$libinclude lsadata ordered_probit "using nlp maximizing LOGLIK"
execute_load '%gams.scrdir%lsadata.gdx',H=obj_coef_coef;

*       Convert the Hessian to a symmetric matrix:
H(k,kk) = H(k,kk) + H(kk,k)$(not sameas(k,kk));

*       Find inverse of the Hessian
ident(k,k)$sum(kk$H(k,kk),1) = 1;
$libinclude lufactorsolve H ident cov

*       Report statistics for the probit model
stat(k, "estimator")  = COEF.L(k);
stat(k, "std error")  = sqrt(cov(k,k));
stat(k, "t value")$cov(k,k) = COEF.L(k)/sqrt(cov(k,k));

*       Use the BETAREG function:
stat(k, "P value")$cov(k,k)
                     = BETAREG( (card(i) - card(k))/
                                (card(i) - card(k) + sqr(stat(k, "t value"))),
```

```
                                        (card(i) - card(k))/2, 0.5 );

display stat;
```

We have typical regression statistics as follows:

```
----    294 PARAMETER stat  Statistics reported

           estimator   std error     t value     P value

cons           0.772       0.078       9.929

day_cost       0.763       0.072      10.663

ctchrate       0.999       0.092      10.805

mnth_inc      -0.350       0.139      -2.509       0.012

mu_p           0.602       0.041      14.531

mu_pb          1.695       0.057      29.744     2.0E-145
```

2.4 Non-linear least squares (NLS)

In the following non-linear least squares models, we minimize sum of squared errors to estimate unknown before reporting standard regression statistics. Some of these statistics are based on Jacobian/Hessian values derived from `LSA`.

2.4.1 A CES Production Function (`ces`)

In this example, we first use the Mizon (1977) data accessed in Example 1 to estimate unknowns of the following Constant Elasticity of Substitution (CES) production function with multiplicative 2.4.1 error terms:

$$q_t = \phi[\delta k_t^{\rho} + (1-\delta) l_t^{\rho}]^{\frac{1}{\rho}} \exp(\mu_t) \tag{2.4.1}$$

where:

$\phi \equiv$ scale parameter, $\phi > 0$

$\delta \equiv$ distribution parameter, $0 < \delta < 1$

$\rho \equiv$ the substitution parameter, $-\infty < \rho < 1$, $\rho \neq 0$.

Comparing to the previous example, the production function in this exercise takes a more general CES form (Cobb-Douglas production function is an extreme case of CES when $\rho = 0$). Similar to Example 1, we estimate the unknown vector of parameters θ by minimizing the Sum of Squared Errors (SSE) subject to the CES production function. We report estimator of unknown vector $\hat{\theta}$, estimated variances based on the diagonal elements of the covariance matrix $\hat{V}_{\hat{\theta}}$, and t-statistics as well as p-values as we did in the previous example.

```
$title GAMS Solution of an NLS problem with CES production function
*       Two forms of CES production functions


*       Multiplicative Disturbance:
*       Part(a)
*       Estimate unknown coeffcents in CES production function
set     t       Observations   /t1*t72/,
        i       Unknowns to be estimated
                /delta          Value share of capital,
                 rho            Substitution parameter,
                 phi            Scale factor/;

alias (i,ii), (t,tt);

parameter       data(t,*)       Source data;
$call gdxxrw mizon_1977.xlsx par=data rng=A1:D73 cdim=1 rdim=1 checkDate
$gdxin mizon_1977.gdx
$loaddc data
$gdxin
```

```
*       Define explanatory variables
parameter       q(t)            Quantity of production,
                k(t)            Capital input,
                lb(t)           Labor input;

*       Introduce data
q(t)    = data(t,"q");
k(t)    = data(t,"cap");
lb(t)   = data(t,"lab");

*       Multiplicative Disturbance model (MD)
variable        THETA(i)        Unknowns to be estimated,
                MU(t)           Error terms,
                SSE             Sum of squared errors;

equation        fitmd(t)        CES production in the MD model,
                obj             Objective function in the MD model;

$macro          agg(t)  ( THETA("delta")*k(t)**THETA("rho") +\
                         (1-THETA("delta"))*lb(t)**THETA("rho") )

obj..           SSE =e= sum(t,sqr(MU(t)));
fitmd(t)..      log(q(t)) =e= log(THETA("phi")) + 1/THETA("rho")*log(agg(t)) + MU(t);

*       Set bounds on "phi","kappa" and "delta"
THETA.LO("phi")   = 0.01;
THETA.UP("phi")   = inf;
THETA.LO("delta") = 0.01;
```

```
THETA.UP("delta") = 0.99;


MU.LO(t) = -1000;
MU.UP(t) = 1000;


model CES_MD /obj, fitmd/;


*       Optimize when rho is between 0 and 1 first
THETA.LO("rho")  = 0.01;
THETA.UP("rho")  = 0.99;
THETA.L ("rho")  = 0.5;
```

After solving this conditional optimization given $0 < \rho < 1$, use *savpoint* to save temporary results in the data file `ces_md_p.gdx` for future comparison.

```
*       Save current results in "ces_md_p.gdx"
CES_MD.SAVEPOINT = 1;
solve CES_MD minimzing SSE using nlp;
CES_MD.SAVEPOINT = 0;


parameter       sse_value       Sum of squares;
sse_value = SSE.L;


*       Solve the problem again when rho is negative
THETA.LO("rho")   = -inf;
THETA.UP("rho")   = -0.01;
THETA.L ("rho")   = -0.5;


solve CES_MD minimzing SSE using nlp;
```

```
*       If the solution value is made worse,
*       load the previously computed solution:
if (SSE.L > sse_value, execute_loadpoint 'ces_md_p.gdx';);
```

Next, we report typical regression statistics for the CES model. The process is similar to its counterpart of the previous example.

```
*       Report regression statistics
parameter
                J(t,i)                  Jacobian,
                J_sqr(i,ii)             Squared Jacobian,
                jsinv(i,ii)             Inverse of J_sqr,
                ident(i,ii)             Identity matrix,
                cov(i,ii)               Covariance matrix,
                statistics(i,*)         Statistics at the point;

*       Generate an LSA request to retrieve Jacobian:
$libinclude lsadata ces_md "using nlp minimizing sse"
execute_load '%gams.scrdir%lsadata.gdx',J=fitmd_theta;

*       Find squared Jacobian matrix
J_sqr(i,ii)  = sum(t, J(t,i)*J(t,ii));

*       Prepare for finding inverse of J_sqr
ident(i,i)$sum(ii$J_sqr(i,ii),1) = 1;

*       Find the inverse of squared Jacobian
$libinclude lufactorsolve J_sqr ident jsinv

*       Note: SSE.L / (card(t) - card(i)) is unbiased estimator of error variance
```

```
cov(i,ii)  = jsinv(i,ii) *  SSE.L / (card(t) - card(i));


statistics(i"estimator")  = THETA.L(i);

statistics(i,"std error")  = sqrt(cov(i,i));

statistics(i,"t value")$cov(i,i)    = THETA.L(i)/sqrt(cov(i,i));


*       Use the BETAREG function:

statistics(i,"P value")$cov(i,i)

                = BETAREG( (card(t) - card(i))/

                           (card(t) - card(i) + sqr(statistics(i,"t value"))),

                           (card(t) - card(i))/2, 0.5 );
```

Here is the list of regression statistics reported from GAMS:

```
----    353 PARAMETER statistics  Statistics at the point

        estimator   std error     t value     P value

delta       0.277       0.034       8.264     6.6E-12

rho        -0.344       0.206      -1.673       0.099

phi         1.740       0.053      32.678     1.0E-43
```

(b) Under the CES specification in (2.4.1) evaluate the marginal products of Capital and Labor when these inputs are at their mean values. Are these marginal products positive from a statistical point of view?

We first evaluate the marginal products of capital (MPK) when capital and labor are at their mean values.

```
*       Part (b)

set     m       Statistical tests

                /mpk            T test on marginal product of capital,

                 mpl            T test on marginal product of labor/;
```

```
parameter       kbar            Mean of capital,
                lbar            Mean of labor,
                grad(i)         Gradients retrieved from lsa,
                tval(m)         T value of t-test against H0 in test m,
                pval(m)         P value of t-test against H0 in test m,
                alpha           Type 1 error tolerance level     /0.05/;

*       Define average of capital and labor
kbar = sum(t,  k(t))/card(t);
lbar = sum(t, lb(t))/card(t);

*       Test whether marginal product of capital is zero
variable        MPK_MD          Marginal (mean) product of capital in the MD model;
equation        def_mpk         Definition of marignal product of capital in the MD model;

$macro          agg_bar         ( THETA("delta")*lbar**THETA("rho") +\
                                 (1-THETA("delta"))*kbar**THETA("rho") )

def_mpk..       MPK_MD =e= THETA("phi")*THETA("delta")*kbar**(-1 + THETA("rho"))*
                           agg_bar**(1/THETA("rho") - 1)*exp(sigma2_md/2);

model mp_k /def_mpk/;
THETA.FX(i) = stat("MD",i,"estimator");
solve mp_k using mcp;
```

In order to test whether MPK is statistically positive, we need gradients of the marginal product of capital, MPK_MD with respect to unknowns THETA(i), evaluated at point estimates' level THETA.L(i). `LSA` routine helps us in extracting gradients in an `MCP` model, and calculates the *T-test* statistic based on null hypothesis $\mathbb{H}_0 : MPK = 0$:

```
*       Generate an LSA request to retrieve Jacobian:
$libinclude lsadata mp_k
execute_load '%gams.scrdir%lsadata.gdx',grad=def_mpk_theta;

*       Find adjusted covariance, report t-value and P-value
tval("mpk")   = MPK_MD.L/sqrt(sum((i,ii), grad(i)*cov(i,ii)*grad(ii)));

*       Use betareg function with degrees of freedom = card(i);
pval("mpk")   = betareg( card(i)/(card(i) + sqr(tval("mpk"))), card(i)/2, 0.5 );

*       Make a report of test results
$onecho >%gams.scrdir%report.gms
parameter       test(m,*,*)             Test Results;
loop(m$(pval(m) lt alpha),
        test(m, "Reject H_0: marginal product of factor = 0", "P value")   = pval(m);
        test(m, "Reject H_0: marginal product of factor = 0", "t value")   = tval(m););

loop(m$(pval(m) ge alpha),
        test(m, "Fail to reject H_0: marginal product of factor = 0", "P value") = pval(m);
        test(m, "Fail to reject H_0: marginal product of factor = 0", "t value") = tval(m);)

option test:3:2:1;
display "Test Report:", test;
$offecho

$include "%gams.scrdir%report"
```

We then apply the same method to test whether null hypothesis $\mathbb{H}_0 : MPL = 0$ is true in the other *T-test*:

```
*       Test whether marginal product of labor is zero
variable        MPL_MD          Marginal (mean) product of labor in the MD model;
equation        def_mpl         Definition of marignal product of labor in the MD model;

def_mpl..
        MPL_MD =e= THETA("phi")*(1 - THETA("delta"))*lbar**(-1 + THETA("rho"))*
                   agg_bar**(1/THETA("rho") - 1)*exp(0.5*SSE.L/(card(t) - card(i)));

model mp_l /def_mpl/;
solve mp_l using mcp;

*       Generate an LSA request to retrieve Jacobian:
$libinclude lsadata mp_l
execute_load '%gams.scrdir%lsadata.gdx',grad=def_mpl_theta;

*       Find adjusted covariance
tval("mpl") = MPL_MD.L/sqrt(sum((i,ii), grad(i)*cov(i,ii)*grad(ii)));

*       Use Betareg function with degree of freedom: card(i)
pval("mpl")   = betareg( card(i)/(card(i) + sqr(tval("mpl"))), card(i)/2, 0.5 );
$include "%gams.scrdir%report"
```

Given test statistics, we have enough evidence to reject both null hypothesis: either MPK or MPL is different from zero, *i.e.*, positive:

```
----    887 PARAMETER test  Test Results

                                                    t value     P value

mpk.Reject H_0: marginal product of factor = 0      10.596       0.002
mpl.Reject H_0: marginal product of factor = 0      14.492      7.1E-4
```

2.5 Conclusion

In this paper, we combine existing GAMS resources in order to retrieve key statistics in econometrics study. Several classical parametric estimation problems, including nonlinear least squares model, limited dependent variable model and some other models could be solved using maximum likelihood estimation are listed as examples in the paper.

Although the amount of variety in solved problems is limited, our main goal in this paper is to demonstrate the advantage of algebraic modeling languages such as GAMS in applied optimization. More recent econometrics applications, say mixed *logit* and multivariate *probit* model will be studied in some follow-up work based on this.

Chapter 3

Local Sensitivity Analysis and Top-Down / Bottom-Up Decomposition

3.1 Introduction

The purpose of this paper is to demonstrate a new decomposition approach for energy-economy modeling called block decomposition with small scale applications. We aim to improve on the decomposition procedure proposed by Böhringer and Rutherford (2009) in a *top-down/bottom-up* fashion energy model with the help of our new calculation tool named Local Sensitivity Analysis (LSA), which provides us easy assess to accurate local gradients. Starting from simple numerical examples, we illustrate basic ideas of block Newton algorithm which is the key of block decomposition, and comparing it to related approaches such as the Newton method as well as the *diagonalization* method. And we test through numerical examples that in many cases, block newton method has advantage over the *diagonalization* approach in robustness.

In this paper, market equilibrium problems are conveniently formulated as Mixed Complementarity Problems (MCP), thereby permitting integration of *bottom-up* programming models of the energy-environment system within *top-down* general equilibrium models of the overall economy. Despite the logical appeal of the complementarity approach, Böhringer and Rutherford (2009) point out that dimensionality constraints pose a fundamental limitation for the MCP approach. The complementarity formulation encompasses both primal and dual variables, often doubling the number of equations and the scope for error. When a large number of the equilibrium conditions can be represented in the form of an optimization model, decomposition methods are often the sole feasible approach. We built our decomposition procedure on that of Böhringer and Rutherford (2009)'s. They present an *ad-hoc* decomposition algorithm for solving energy-economy models. One feature of their decomposition is they impose exogenous own-price demand elasticities to linearize *bottom-up* energy demand. The present paper proposes a more careful and general decomposition in which the partial-equilibrium (*bottom-up*) model use `LSA` to update a demand system with calculated own- and cross-price elasticities. Endogenous elasticity information could contribute to the model's overall performance in a sense that we don't have to worry about its sensitivity with respect to elasticity parameters any longer.

The remainder of this paper is organized as follows. In section 2, we review *top-down/bottom-up* modeling especially the Böhringer and Rutherford (2009) decomposition. In section 3, we introduce block decomposition as well as block Newton approach, and we also explain how `LSA` tools help in improving the decomposition procedure. In section 4, we lay out several numerical examples in which we demonstrate how block decomposition works in different context. We also compare block decomposition with some other approaches. In section 5, we focus on its application in the *top-down/bottom-up* energy model. In section 6, we conclude.

3.2 Top-down/Bottom-up Modeling and BR Decomposition

Top-down and *bottom-up* are modeling paradigms to represent interactions between the energy system and the economy. The terms "*top-down*" and "*bottom-up*" are shorthand for aggregate and disaggregated models. Models in the first category emphasize economy-wide relationships, while those in the second category feature sectoral and technological details.

The major differences between *top-down* and *bottom-up* models lie in their domains (economy-wide versus the energy system), solution concept (primal partial equilibrium versus primal-dual general equilibrium), scope for substitution (the demand influences of non-energy sectors and the supply influences of factor price changes versus discrete technology set), and inclusion of optimistic low-cost or negative-cost emission reduction possibilities(IPCC (2001)).

Top-down models examine the broader economy and incorporate feedback effects between different markets triggered by policy-induced changes in relative prices and incomes. They typically do not feature technological details of energy production or conversion. Energy sectors - like other non-energy sectors - are typically represented in an aggregate way by means of smooth production functions which capture substitution (transformation) possibilities via substitution (transformation) elasticities. For example, some typical *top-down* computable general equilibrium (CGE) models assume that each economic sector is represented by a price-taking firm producing a homogeneous good. As a consequence, conventional *top-down* models cannot readily incorporate different assumptions about how discrete energy technologies and costs evolve into the future. *Top-down* models may also violate fundamental physical restrictions such as the conservation of matter and energy. Well known examples of *top-down* models include aggregate growth model with an environmental sector such as the DICE or RICE model introduced by Nordhaus and Boyer (1999), and primal-dual CGE models such as Paltsev et al. (2005).

In contrast, *bottom-up* energy system models are partial equilibrium representations of the energy sector. They feature a large number of discrete energy technologies to capture substitution of energy carriers on the primary and final energy level, process substitution, or efficiency improvements. Such models often neglect the macroeconomic impact of energy policies. *Bottom-up* energy system models are typically cast as optimization problems which compute the least-cost combination of energy system activities to meet a given demand for final energy or energy services subject to technical restrictions and energy policy constraints.

Given specific strengths and weaknesses of the *bottom-up* and *top-down* approaches, there are various hybrid modeling efforts that aim at combining the technological explicitness of *bottom-up* models with the economic richness of *top-down* models. These efforts may be distinguished into three broader categories: First, independently developed *bottom-up* and *top-down* models can be linked. This approach has been adopted since the early 1970's, but it often challenges overall coherence due to inconsistencies in behavioral assumptions and accounting concepts across "soft-linked" models. For example, Jacobsen (1998) discuss theoretical and methodological problems for integrating existing models for Denmark; Drouet et al. (2005) list three types of "hybrid" attempts and they focus on the coupling a world computable general equilibrium model (CGEM) and of a bottom up energy-technology-environment model (ETEM). Second, one could focus on one model type -either *bottom-up* or *top-down*- and use "reduced form" representations of the other. Prominent examples include ETA-Macro (Manne (1977)) and its successor MERGE (Manne et al. (1995)) which link a *bottom-up* energy system model with a highly aggregate one-sector macroeconomic model of production and consumption within a single optimization framework. Other examples include Bosetti et al. (2006), Messner and Schrattenholzer (2000) and Strachan and Kannan (2008). The third approach provides completely integrated models based on developments of solution algorithms for Mixed Complementarity Problems (MCP) during the 1990's (Dirkse and Ferris (1995); Rutherford (1995)). In an earlier paper, Böhringer (1998) stresses the difference between *bottom-up* and *top-down* with respect to the characterization of technology options and associated input substitution possibilities in production. He describes this hybrid approach with a small, static, MCP formulated model with stationary technologies. Later, Frei et al. (2003) show how the complementarity format can be used in CGE modeling for a dynamic formulation of *bottom-up* and *top-down* approach merging models. More lately, Wing (2006) and Wing (2008) propose a structure and numerical algorithm to model electricity generation as a constant elasticity of substitution (CES) aggregate of multiple technologies; Fujimori et al. (2014) use the Logit functions to determine the share of electricity production from a particular generation technology. These papers highlight the importance of "true" technology-based activity analysis in evaluating policy induced structural change at the sectoral level.

Starting from the `Arrow-Debreu` equilibria, Bohringer and Rutherford (2008) show that complementarity is a feature of economic equilibrium rather than equilibrium condition, this permits the modeler to integrate *bottom-up* activity analysis directly within a *top-down* representation of the broader economy. Bohringer and Rutherford (2008) also provide numerical example of a hybrid *bottom-up/top-down* model, within both static and dynamic settings. Their followed up paper of Böhringer and Rutherford (2009) is motivated by the following findings: Despite the coherence of the integrated complementarity approach, dimensionality may impose limitations on its practical application. A large number of bounds on decision variables in *bottom-up* model introduce unavoidable complexity in the integrated complementarity formulation since they must be associated with explicit price variables in order to account for income effects. Böhringer and Rutherford (2009) propose a decomposition technique to overcome complexity and dimensionality restrictions. They employ an iterative solution procedure in a decomposition algorithm to solve *top-down* and *bottom-up* model components consistently. In more details, they propose iterative solution of the *top-down* general equilibrium model given net supplies from the *bottom-up* energy sector sub-model followed by the solution of the energy sector sub-model based on a locally calibrated set of demand functions for energy sector outputs. The key idea behind this is when price responsiveness of energy supply is ignored, *i.e.*, fixed energy supply and price, the *top-down* general equilibrium model can be solved as a complementarity problem, and a solution in the *bottom-up* energy model (which is represented as a quadratic programming problem) given equilibrium prices from the *top-down* model would update supply and price of energy goods in each iteration. However, some may question the assumption that it is valid to ignore energy supply's price responsiveness, or may also challenge the form of the locally calibrated set of demand functions with fixed demand elasticities. Before we revisit the *bottom-up/top-down* decomposition of Böhringer and Rutherford (2009) to address these concerns, we first introduce block decomposition algorithm and its application in energy-economy modeling.

3.3 Block Decomposition

Suppose that we have a nonlinear system of equations to solve which is[1]:

$$F(z) = 0, \tag{3.3.1}$$

where $F : R^n \to R^n$ and $z \in R^n$. Given the setting of Böhringer and Rutherford (2009), we want to exploit special structure in F wherein a subset of equations and variables constitute first-order conditions for a quadratic optimization problem[2]. We then produce a conformal partition F:

$$F(z) \equiv F\begin{pmatrix} x \\ y \end{pmatrix} \equiv \begin{pmatrix} G(x, y) \\ H(x, y) \end{pmatrix}$$

where $G : R^n \to R^m$ and $x \in R^m$.

The simplest decomposition method for finding a solution of the original system would involving solve first for x in terms of $\bar{y}$:

$$G(x, \bar{y}) = 0, \tag{3.3.2}$$

and then solving for y in terms of $\bar{x}$:

$$H(\bar{x}, y) = 0. \tag{3.3.3}$$

This is a *diagonalization* approach which only works well when the off-diagonal terms, $\nabla_y G$ and $\nabla_x H$ are "small". Otherwise, this approach is likely to produce linear convergence at best. We will discuss this later in a numerical example.

In order to motivate the Newton algorithm[3]for block decomposition, consider a local

[1] Nonlinear complementarity problem $F(z) \perp z \geq 0$ are initially considered here.

[2] This can be generalized to account for inequalities, but for the time being we focus on smooth equations in order to simplify notation.

[3] This method finds successively better approximations of roots $z : f(z) = 0$ of a real valued function. For example, in one variable Newton method, starting from initial value z_0, we solve $z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)}$ successively until a sufficiently accurate root $z^* : f(z^*) = 0$ been found.

sensitivity analysis of the two subsystems, or blocks, (3.3.2) and (3.3.3). At a given solution point the implicit function theorem provides a first order Taylor approximation of x as a function of $\bar{y}$ at a point x^* which solves $G(x^*, \bar{y}) = 0$:

$$L_x(y)|_{\bar{y}} = x^* - \nabla_x G^{-1} \nabla_y G(y - \bar{y}), \tag{3.3.4}$$

and likewise, we can express y as a function of $\bar{x}$ at a point y^* for which $H(\bar{x}, y^*) = 0$:

$$L_y(x)|_{\bar{x}} = y^* - \nabla_y H^{-1} \nabla_x H(x - \bar{x}). \tag{3.3.5}$$

The local approximation can then be employed to approximate the omitted equations when solving for x and y. That is, we solve

$$G(x, L_y(x)) = 0, \tag{3.3.6}$$

and

$$H(L_x(y), y) = 0. \tag{3.3.7}$$

Notice that the linear approximation of y, L_y, depends on both x and the local approximation point $\bar{x}$. Our algorithm must therefore be initiated from an initial point, $z^0 = \begin{pmatrix} x^0 \\ y^0 \end{pmatrix}$. Computations in iteration k of the algorithm then consists of the following steps:

1. Construct $L_x(y)|_{x^{k-1}}$,
2. Solve (3.3.6) for x^k,
3. Construct $L_y(x)|_{y^{k-1}}$,
4. Solve (3.3.7) for y^k,
5. Evaluate $\epsilon = ||F(z^k)||$ where $z^k = \begin{pmatrix} x^k \\ y^k \end{pmatrix}$,
6. If $\epsilon > \bar{\epsilon}$, $k \leftarrow k + 1$, else exit.

To recap, suppose we are solving a model which can be expressed as a nonlinear complementarity problem:

$$F(z) \perp z \geq 0,$$

as is the basic thrust of procedure in Böhringer and Rutherford (2009). When we partition the variables z into those related to the economy, x, and those related to the energy system, y, the Jacobian of $F(z)$ might then been accordingly partitioned:

$$\nabla F = \begin{bmatrix} \nabla_{xx} F & \nabla_{xy} F \\ \nabla_{yx} F & \nabla_{yy} F \end{bmatrix}.$$

When the off-diagonal elements of the Jacobian are small, then a very simple diagonalization procedure can solve the model:

Initialize:

- $k \leftarrow 0$,
- Assign initial values, x^0 and y^0,
- Choose a solution tolerance, ϵ^*,
- Evaluate $\epsilon = \mathcal{E}(z^k) = ||F(z^0) \perp z^k||$,

Repeat while $\epsilon > \epsilon^*$:

- Solve
$$F(x^*, y^k) \perp x^*, \tag{3.3.8}$$
- Solve
$$F(x^k, y^*) \perp y^*, \tag{3.3.9}$$
- Assign $x^{k+1} = x^*$, $y^{k+1} = y^*$,
- $k \leftarrow k + 1$,
- Evaluate $\epsilon = \mathcal{E}(z^k) = ||F(z^k) \perp z^k||$.

The problem with this approach is that it could result in poor rates of convergence or non-convergence when the off-diagonal terms ($\nabla_{xy}F$ and $\nabla_{yx}F$) are significant. Instead we rely on something more subtle, replacing the simple lagged values by approximating functions. That is, we might replace x by

$$x = h(y; \bar{x}, \bar{y}),$$

in solving for y^* and replace y by

$$y = g(x; \bar{x}, \bar{y}),$$

when solving for x^*. When functions we use provide a locally exact approximation of those responses, the convergence rate can be quadratic and relatively fewer steps are required.

The algorithm with calibrated approximation functions is then:

Initialize:

- $k \leftarrow 0$,
- Assign initial values, x^0 and y^0,
- Choose a solution tolerance, ϵ^*,
- Evaluate $\epsilon = \mathcal{E}(z^k) = ||F(z^0) \perp z^k||$,

Repeat while $\epsilon < \epsilon^*$:

- Solve

$$F(x^*, y = g(x^*; x^k, y^k)) \perp x^*, \tag{3.3.10}$$

- Solve

$$F(x = h(y^*; x^k, y^k), y^*) \perp y^*, \tag{3.3.11}$$

- Assign $x^{k+1} = x^*$, $y^{k+1} = y^*$,
- $k \leftarrow k + 1$,
- Terminate when $\epsilon = \mathcal{E}(z^k) = ||F(z^k) \perp z^k||$.

For example, we might use affine functions to approximate these responses:

$$y = g(x; x^k, y^k) \equiv y^k + A(x - x^k), \tag{3.3.12}$$

and

$$x = h(y; x^k, y^k) \equiv x^k + B(y - y^k), \tag{3.3.13}$$

where A characterizes local sensitivity of energy goods in the *bottom-up* subsystem, y, with respect to variables associated with the general economy, x, at the solution point y^* and B characterizes the local sensitivity of x with respect to y at x^*.

We aim to apply this version of block decomposition to the *top-down/bottom-up* energy model. However, due to the complicity of the energy example, before applying approximation function (3.3.12) and (3.3.13) to improve the *top-down/bottom-up* decomposition in Böhringer and Rutherford (2009), we first work our way up applying block decomposition ideas in some simple numerical examples in the next section.

3.4 Numerical Examples

We first introduce our calculation tool `LSA` through which we derive precise local sensitivity results. This tool helps to make block decomposition less costly and more convenient to use in practice.

3.4.1 Local Sensitivity Analysis and its Interface in GAMS

Similar to Rutherford and Wang (2014) and Wang (2017), we make use of the `LSA` calculation tools in retrieving derivatives without numerical differentiation.

To begin with, `LSA` works in an economic model formulated as a system of equations

$$F_i(z; t) = 0 \quad i = 1, \ldots, n$$

in which $z \in \mathrm{R}^\mathrm{n}$ is a vector of equilibrium variables and $t \in \mathrm{R}^\mathrm{m}$ is a vector of strategic policy variables. Suppose we are interested in the dependence of z_j ($j = 1, \dots, n$) on t_k ($k = 1, \dots, m$), $\frac{\mathrm{dz_j}}{\mathrm{dt_k}}$, in matrix notation, this can be written as

$$\left(\frac{\mathrm{dz.}}{\mathrm{dt_k}}\right) = -\left(\nabla_z F(z;t)\right)^{-1} \ \nabla_{t_k} F(z;t) \tag{3.4.1}$$

if the Jacobian matrix $\nabla_z F(z;t)$ is invertible. Traditional way of finding `Jacobian` matrix $\nabla_z F(z;t)$ and $\nabla_t F(z;t)$ is through numerical differentiation. For example, for the i^{th} equation $F_i(\cdot)$ and the j^{th} unknowns z_j, given a small disturbance of value, $u \to 0$, we evaluate the local derivative of $F_i(\cdot)$ with respect to z_j at point $z_j = z_{j0}$:

$$\left(\frac{\partial F_i}{\partial z_j}\right)|_{z_{j0}} \approx \frac{F_i(z_{j0}+u,t) - F_i(z_{j0},t)}{u}.$$

Similarly procedure is followed when we derive Jacobian $\nabla_t F(z;t)$.

In the worked example, Böhringer and Rutherford (2009) calculate `Jacobian` with the same technology in a partial equilibrium setting. For instance, a piece of code for the dependence of PE(i) (the energy prices for energy good i) with respect to ESUP(i) (the current supply for energy goods i in the *top-down* model, which is taken as given from the *bottom-up* model in the previous iteration) is documented as follows:

```
alias (i,id);
loop(id,
        ESUP.FX(id) = ESUP.L(id) + 0.001;
        solve topdown using mcp;
        ESUP.FX(id) = ESUP.L(id) - 0.001;
        dpdq(i,id) = (PE.L(i)-peref(i))/0.001;
);
```

in which id is another name of set of energy goods i, dpdq(i,id) is the dependence of interest, peref(i) is the reference energy goods prices and perturbation is $u = 0.001$.

With the help of `LSA`, we could instead pull out first order derivatives mentioned above as follows:

```
$libinclude lsa topdown PE ESUP
execute_load '%gams.scrdir%lsa.gdx', PE_ESUP;
```

3.4.2 Block Decomposition Examples with Quadratic Function

3.4.2.1 Quadratic Function with Two Goods in Two Blocks

To demonstrate basic ideas of block decomposition, we begin with a simple quadratic model with two goods (or blocks, with one good in each block): z_k, for $k \in \{1, 2\}$. Suppose we have a quadratic function $f(z) = 0$ with $f : R^2 \to R^2$ and $z \in R^2$:

$$f_k = \alpha_k^0 + \Sigma_i \alpha_{ki}^1 z_i + \Sigma_{ij} \alpha_{kij}^2 z_i z_j, \tag{3.4.2}$$

with coefficients α_k^0, α_{ki}^1, α_{kij}^2 (where i,j are alias names of k) satisfying

$$\alpha_k^0 = -\left(\Sigma_i \alpha_{ki}^1 + \Sigma_{ij} \alpha_{kij}^2\right), \tag{3.4.3}$$

to make sure that $z^* = (1, 1)^T$ is a solution for this nonlinear system. We choose this functional form not only for its simplicity, but also for the fact that some coefficients in the quadratic function has clear meanings: for example, in block k, the coefficient associated with the term $z_i z_j$, α_{kij}^2, equals the corresponding element in the `Hessian` by construction, $\frac{\partial f_k^2}{\partial z_i \partial z_j}$. Despite the fact that it is a non-convex function which limits our choice of initial values in root finding procedure, it serves our purpose well in presenting some fundamentals of the `block Newton` algorithm.

We compare two model solving approaches. The first one is to apply `Newton` method directly to the whole vector $z = (z_1, z_2)^T$ for roots $z^* : f(z^*) = 0$. We solve this nonlinear system iteratively through Newton steps. Note that `Newton` method is not applied here to decompose the model, and we rely on the block Newton method for this task. In each step, we define linear function over variable set k according to the following condition:

$$\bar{f}_k = \Sigma_j \frac{\partial f_k}{\partial z_j} \left(Z_j - \bar{z}_j\right), \tag{3.4.4}$$

in which $\frac{\partial f_k}{\partial z_j}$ is the local gradient of the function f_k with respect to variable z_j, $\bar{z}_j$ is the current value of variable z_j, and $\bar{f}_k$ is the current value of function f_k. For a simple model like this, we calculate $\frac{\partial f_k}{\partial z_j}$ explicitly (we could instead launch `LSA` to calculate $\frac{\partial f_k}{\partial z_j}$ as we would do later in other examples):

$$\frac{\partial f_k}{\partial z_j} = \alpha^1_{kj} + \Sigma_i(\alpha^2_{kij} + \alpha^2_{kji})\bar{z}_i, \tag{3.4.5}$$

We thus solve the model containing equations (3.4.4) to find $z = (1, 1)^T$ as a solution if we pick the an initial guess z_0 in a small enough neighbour around this solution.

We could otherwise apply the `Newton` method to partitions (blocks) of variables z as well as equations f, *i.e.*, `block newton` method. Consider a partitioned function with linear approximation equations (3.3.4) and (3.3.5) as follows:

$$f(z) \equiv f \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \equiv \begin{pmatrix} f_1(z_1, z_2) \\ f_2(z_1, z_2) \end{pmatrix} \equiv \begin{pmatrix} f_1\left(z_1, L_2(z_1)\right) \\ f_2\left(L_1(z_2), z_2\right) \end{pmatrix},$$

where $f_1 : R^2 \to R$, and $z_1 \in R$. And we find linearized variable z_1 as a function of the other variable z_2 with the help of gradients $\frac{\partial f_k}{\partial z_j}$ in the lower partition of the above matrix, *i.e.*, f_2, and *vice versa*[4]. Of course, z_1 stays the same in its own block, f_1.

Linearized variable LZ_j in this one-good-in-one-block model is defined as:

$$LZ_j = \begin{cases} z_j & j \in kb(j) \\ \bar{z}_j - \bar{f}_j / \frac{\partial f_j}{\partial z_j} - \left(\frac{\partial f_j}{\partial z_{j+1}} / \frac{\partial f_j}{\partial z_j}\right)(Z_{j+1} - \bar{z}_{j+1}) & j \notin kb(j) \end{cases} \tag{3.4.6}$$

in which $j \in kb(j)$ means variable z_j in block containing function f_j.

We next solve this block decomposition model as system of equations modeled by (3.3.6) and (3.3.7): we first linearize variable z_j which is not in its own block using equation (3.4.6), thus apply `Newton` method on following functions to find solution with respect to linearized

[4]Note that we refer z_i, $i \neq j$ as the circular lead of z_j (Z_{j+1}) in GAMS for we only have two variables z_1 and z_2 in the vector z. In a two variable case like this, z_1's circular lead (lag) is z_2 and *vice versa* and we use set kb(j) to indicate whether variable z_j is in its own block. For more information about the circular leads/lags, see https://www.gams.com/latest/docs/userguides/mccarl/circular_or_equilibrium_leads_and_lags_plus_plus_minus_minus.htm

variable LZ_j:

$$\bar{f}_k = \Sigma_j \frac{\partial f_k}{\partial z_j} \left(LZ_j - \bar{z}_j \right), k \in kb(k) \tag{3.4.7}$$

And if we start from an initial guess z_0 close enough to the solution point, we find two methods converge to $(1,1)^T$ as follows:

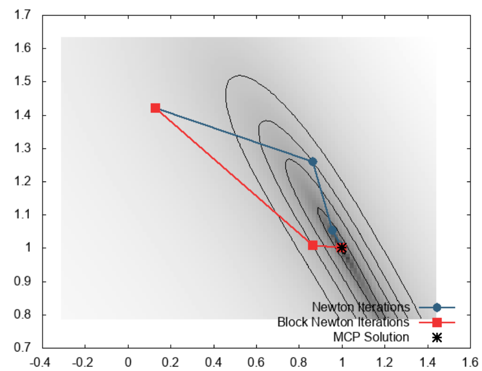

Figure 3.4.1: Newton and Block Newton Methods converge in Quadratic Model with Two Goods

However, as mentioned before, having a non-convex function means it is possible that different algorithm finds different local minimum if we pick bad initial guesses of z_0. And that is one reason why we work on a well defined market equilibrium model next:

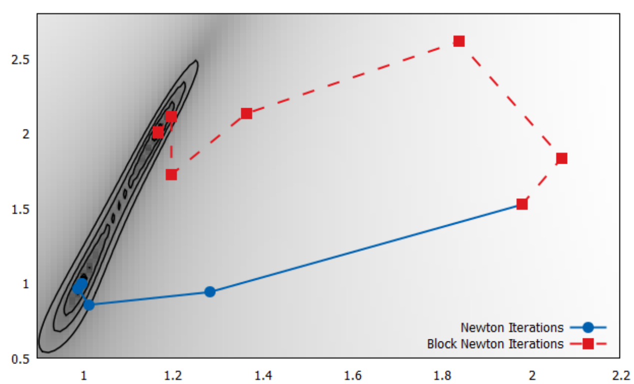

Figure 3.4.2: Non-convexity of Quadratic Model with Two Goods

Before moving to a partial equilibrium market model with two blocks, we further our explore in the quadratic model by introducing more goods (more elements in the vector z) includes some shared goods in both blocks. Remember that a key feature of *top-down/bottom-*

up decomposition is that quantities and prices of energy goods are updated in both sub-models in each iteration.

3.4.2.2 Quadratic Function with Multiple Goods including Two Shared Goods in Two Blocks

We pick up more features of the *top-down/bottom-up* modeling by expanding the previous one-good-in-one-block model. In this version of quadratic model, we divide functions as well as variables into two blocks, namely G and H, each containing 10 block specific goods and two shared goods. For example, we denote goods in block G as z_g with index $g \in \{zg1, ..., zg10, s1, s2\}$, and goods in the block H as z_h with $h \in \{zh1, ..., zh10, s1, s2\}$. Notice the bundle of shared goods $\{z_{s1}, z_{s2}\}$ exists in both blocks and the set of all goods $z \in R^{22}$ as the union of z_g and z_h. In details, variable z_k in this version is defined on the set $k \in \{zg1, ..., zg10, zh1, ..., zh10, s1, s2\}$. We hence have the following partition of function F:

$$F(z) \equiv F \begin{pmatrix} z_g \\ z_h \end{pmatrix} \equiv \begin{pmatrix} G(z_g, z_h) \\ H(z_g, z_h) \end{pmatrix}$$

where $G : R^{22} \to R^{12}$ and $z_g \in R^{12}$, while $H : R^{22} \to R^{12}$ and $z_h \in R^{12}$. Same as in the previous version, we have the quadratic functional form $f(z) = 0$ as (3.4.2) over set k. And the restriction (3.4.3) on coefficients stays the same to guarantee that $z^*_{22\times 1} = (1, \ldots, 1)^T$ is a solution.

According to instruction (3.3.10) of block decomposition, in each solving loop, we begin by solve the G system over z_g as we linearize variables z_h as a function of z_g according to Newton linearization equations (3.4.7), and we define these functions initially on set h (set kl(k) is used to indicate if a variable in in its own block). These steps successfully eliminate z_h. Having linearized z_h as a function of z_g, we then solve the nonlinear equations (3.4.2) for z_g as described in instruction (3.3.10).

Next, we reverse the order of linearization by following the instruction of (3.3.11) to solve the H system as follows: first linearize variables z_g with the help of gradients $\frac{\partial f_k}{\partial z_j}$ and variable

z_h, then solve nonlinear function $f(z) = 0$ to find optimal z_h.

We also test the `Newton` and diagonalization method on this quadratic model. Same as in the previous one, we apply the `Newton` method by solving the linearized model using equations (3.4.7) for all z_k simultaneously. On the other hand, unlike the `block Newton` approach, instead of linearization variables in the other block, in diagonalization procedure we initially take variables in the other block as fixed and solve $f(z) = 0$ to find roots in the current block, as summarized by the instructions (3.3.8) and (3.3.9). In the following graph, we compare different iteration steps of these three different approaches. With a small scale model a series parameters which are friendly to all three methods, they converge fast to find solutions.

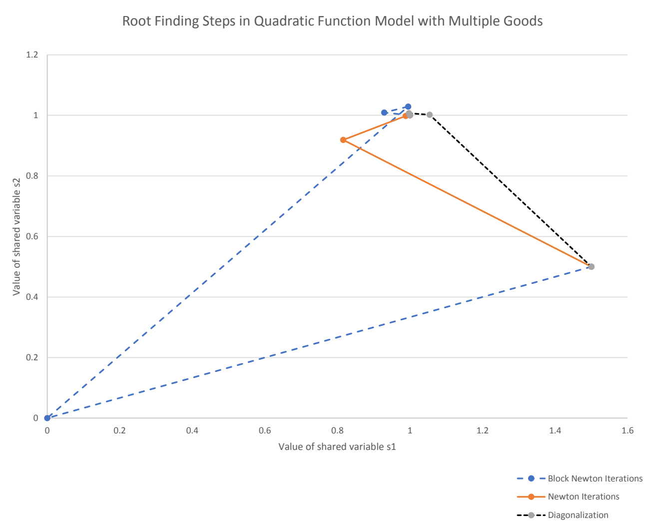

Figure 3.4.3: Quadratic Function with Two Blocks and Two shared Goods

3.4.3 Block Decomposition in a Partial Equilibrium Model

We next apply the `block Newton` method to a simple partial equilibrium model. Market partition is exactly the same as that of the previous model, we divide all goods and functions into two blocks (region), $r \in \{G, H\}$. Set of goods in each block, z_g and z_h, each contain 10 block specific goods and two shared goods (s_1, s_2). And k is the index for all goods. Other than a non-convex quadratic $f_k = 0$ condition for each good in the previous model, we set up market clearance condition based on demand and supply functions for each good k in this model and they are constructed to be convex.

On the supply side, suppose we have a linearized own-price supply function with constant supply elasticities $\eta_{kr} > 0$, we define supply of good k in block r, Y_{kr} as:

$$Y_{kr} = \bar{S}_{kr}(1 + \eta_{kr})(P_k - 1), \tag{3.4.8}$$

where $\bar{S}_{kr}$ is the reference supply of good k in region r, and market price of good k is P_k.

On the demand side, if consumption quantity is $D_{k,r}$, we set up a Constant Elasticity of Substitution (CES) demand function initially for goods in both blocks according to Rutherford (2008), which is a convex nonlinear function, with an elasticity of substitution for each region, σ_r as follows:

$$D_{kr} = \bar{D}_{kr} \left(\frac{P_r^u}{P_k}\right)^{\sigma_r} \frac{m_r}{P_r^u}, \tag{3.4.9}$$

in which unit cost function P_r^u is defined as

$$P_r^u = \Sigma_k \left(\theta_{kr} P_k^{(1-\sigma_r)}\right)^{\frac{1}{(1-\sigma_r)}},$$

with value share $\theta_{kr} = \frac{\bar{P}_k \bar{D}_{kr}}{\Sigma_{kr} \bar{P}_k \bar{D}_{kr}}$ assuming reference price $\bar{P}_k = 1$ for all good k, and m_r index of income in block r, which is currently set at 1.

Having both demand (3.4.9) and supply (3.5.19) functions, we update market clearance condition $f_k = 0$ on good g in block G or good h in H[5]:

$$f_g = \bar{S}_{gG}(1 + \eta_{gG})(p_g - 1) - \bar{D}_{gG} \left(\frac{P_G^u}{P_g}\right)^{\sigma_G} \frac{m_G}{P_G^u} = 0, \tag{3.4.10}$$

[5]Note that we have goods indexed by $g \in \{zg1, ..., zg10, s1, s2\}$ in block G, and goods $h \in \{zh1, ..., zh10, s1, s2\}$ in the block H. The set of all goods $z \in R^{22}$ as the union of z_g and z_h.

or

$$f_h = \bar{S}_{hH}(1+\eta_{hH})(P_h - 1) - \bar{D}_{hH}\left(\frac{P_H^u}{P_h}\right)^{\sigma_H}\frac{m_H}{P_H^u} = 0, \tag{3.4.11}$$

An partial equilibrium model is constructed on equation (3.4.9), (3.5.19), (3.4.10) and (3.4.11). Next, we follow similar block Newton procedure as that in the previous model to find solution z^*. We first linearize demand in block H, holding prices and demand fixed for variables $(zg_1, \ldots, zg_10)$. We then replace the demand function (3.4.9) with two new ones: the first is a linearized demand function indexed by h (shared goods included), and the other is still the same nonlinear demand function defined on goods $(zg_1, \ldots, zg_10)$, which are specific goods associated with block G. With this modified MCP model, we solve variables in H and G together.

Here the key point is how do we linearize demand function in the block H. We have two options:

One is to solve the model with block decomposition using the own-price elasticities. We define linearized demand as

$$D_{kr} = \bar{D}_{kr}(1-\sigma_r)\left(\frac{P_k}{\bar{P}_k} - 1\right), \tag{3.4.12}$$

And the other is using cross-price elasticities. One example is

$$D_{kr} = \bar{D}_{kr} - \Sigma_j \frac{\partial D_{kr}}{\partial P_j}\left(P_j - \bar{P}_j\right), \tag{3.4.13}$$

Both own- and cross- price elasticities could be reported by `LSA` and in a small model like this, both solve the model fast as we show in the following graph (3.4.4). We draw iteration steps of solving variables $(s_1, s_2)^T$ and they started from the initial point $s_0 = (1.9, 0.1)^T$ and converge to the optimal solution $(1,1)^T$ quickly. One thing worth noting is that we could regard the demand side of this partial equilibrium model as a simplest version of the *bottom-up* model and supply side as a *top-down* model, and shared goods $(s_1, s_2)^T$ as energy goods, in this sense, we are performing a simple *top-down/bottom-up* block decomposition.

However, block decomposition using diagonalization does not work well in this model. To

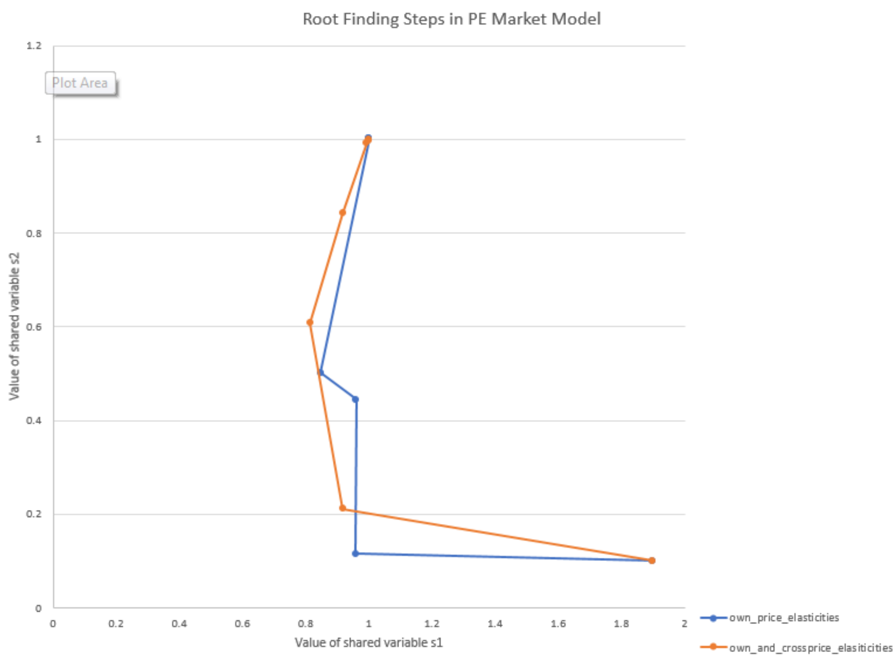

Figure 3.4.4: Block Decomposition with Different Elasticities

do diagonalization, we take prices and quantities in one block as fixed and solve for the other block using the original MCP with equations (3.4.9), (3.5.19), (3.4.10) and (3.4.11). And Newton steps of this approach is drawn here in the following figure. With the same reference demand and supply as before, We find that value of $(s_1, s_2)^T$ jumps back and forth between $(0.143, 37.65)^T$ and $(1, 0.102)^T$ permanently if we decompose the model using diagonalization.

Next, we try block decomposition on Böhringer and Rutherford (2009)'s *top-down/bottom-up* example to see if `LSA` could do some improvement the original decomposition approach.

3.5 General Equilibrium Model with Top-down/Bottom-up Energy Model

In this section we revisit the stylized example reported by Böhringer and Rutherford (2009). We choose this particular application because:

(a) we can build on the careful analysis of Böhringer and Rutherford (2009);

(b) This example illustrates a small scale *top-down/bottom-up* decomposition model and

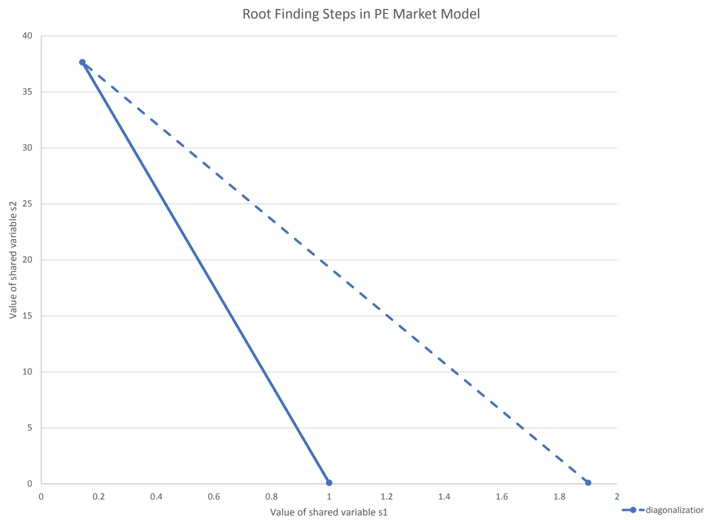

Figure 3.4.5: Block Decomposition with Diagonalization

it leaves space for improvements computationally.

(c) This example illustrates the usefulness of the energy-sector decomposition: welfare differences between the integrated (MCP) model and a *top-down/bottom-up model* are kept low, even when *bottom-up* model of the energy system involves a large number of bounds on decision variables comparing to the totally dimension of the macro model.

Next, we briefly introduce the formulated representative agent model in Böhringer and Rutherford (2009), focusing on its top-down/bottom-up decomposition part. There are two non-energy goods (x and y) and a set of four energy goods (OIL, GAS, COL (coal), and ELE (electricity)) in the example. Böhringer and Rutherford (2009) extend the top-down representation of economic market equilibria as a mixed complementarity problem by an explicit *bottom-up* linear programming model of the energy sector. The complementarity format offers a flexible alternative to non-linear optimization as a means of representing economic equilibrium models. They also lay out the decomposition of the integrated *top-down/bottom-up* model, describing how such a model can be calibrated to base-year social accounts.

3.5.1 The Integrated Model as an MCP

In the MCP framework of Böhringer and Rutherford (2009), by use of Shepard's Lemma they write the equilibrium as the following mixed complementarity problem:

- Zero-profit conditions:

$$\bar{z}_{it} \geq z_{it} \perp \mu_{it} \geq 0 \tag{3.5.1}$$

$$-\Pi^E_{it} \geq 0 \perp z_{it} \geq 0 \tag{3.5.2}$$

$$\Pi_c = 0 \tag{3.5.3}$$

$$\Pi_x = 0 \tag{3.5.4}$$

$$\Pi_y = 0 \tag{3.5.5}$$

- Market-clearence conditions:

$$x = \sum_{it} a^x_{it} z_{it} + c\frac{\partial \Pi_c}{\partial p_x} \tag{3.5.6}$$

$$y = \sum_{it} a^y_{it} z_{it} + c\frac{\partial \Pi_c}{\partial p_y} \tag{3.5.7}$$

$$\bar{L} = x\frac{\partial \Pi_x}{\partial w} + y\frac{\partial \Pi_y}{\partial w} \tag{3.5.8}$$

$$\bar{K}_x = x\frac{\partial \Pi_x}{\partial r_x} \tag{3.5.9}$$

$$\bar{K}_y = y\frac{\partial \Pi_y}{\partial r_y} \tag{3.5.10}$$

$$\sum_t z_{it} - \sum_{i't} b_{ii't} z_{i't} = c\frac{\partial \Pi_c}{\partial p^E_g} + x\frac{\partial \Pi_x}{\partial p^E_g} + y\frac{\partial \Pi_y}{\partial p^E_g} \tag{3.5.11}$$

$$c = \frac{M}{p_c} \tag{3.5.12}$$

- Income balance:

$$M = r_x \bar{K}_x + r_y \bar{K}_y + w\bar{L} + \sum_{it} \mu_{it} \bar{z}_{it} \tag{3.5.13}$$

Table (3.1) provides a summary of the variables appearing in the integrated model.

Activity variables	
c	Aggregate consumption
x, y	Production of goods x and y
z_{it}	Production by technology t for energy good i
Price variables	
p_c	Price index of final consumption
p_x, p_y	Non-energy goods x and y
p_g^E	Energy prices for $i = \{\text{OIL}, \text{GAS}, \text{COL}, \text{ELE}\}$
w	Wage rate
r_x, r_y	Returns to non-energy capital
μ_{it}	Energy sector rents
Income variable	
M	Income of representative agent

Table 3.1: Variables and parameters in the integrated model

3.5.2 BR Decomposition

In decomposition Böhringer and Rutherford (2009) separate the integrated model into a *top-down* model of the overall economy and a *bottom-up* model of the energy supply system. Within the *top-down* model, they treat the energy system *netputs* as exogenous. Energy supply activities are no longer endogenous and they drop equations (3.5.1) and (3.5.2). Net energy supplies and inputs of non-energy goods to the energy system enter the *top-down* model as parameters. Parameterized energy-sector netputs $\tilde{S}_i$ and inputs $\tilde{x}_E$ and $\tilde{y}_E$ are valued at market prices which implicitly include rents on specific energy resources, hence these could be dropped from the income constraint. The adjusted market-clearance condition for energy goods within the *top-down* model is:

$$\tilde{S}_i = c\frac{\partial \Pi_c}{\partial p_i^E} + x\frac{\partial \Pi_x}{\partial p_i^E} + y\frac{\partial \Pi_y}{\partial p_i^E} \tag{3.5.14}$$

and the revised market-clearance conditions for non-energy goods are:

$$x = \tilde{x}_E + c\frac{\partial \Pi_c}{\partial p_x} \tag{3.5.15}$$

and

$$y = \tilde{y}_E + c\frac{\partial \Pi_c}{\partial p_y} \tag{3.5.16}$$

The revised income balance (3.5.13) reads:

$$M = r_x\bar{K}_x + r_y\bar{K}_y + w\bar{L} + \sum_i p_i^E\tilde{S}_i - p_x\tilde{x}_E - p_y\tilde{y}_E \tag{3.5.17}$$

The *bottom-up* model can be represented as a quadratic programming problem in which the sum of producer and consumer surplus is maximized subject to supply-demand balances for energy and resource bounds on technologies:

$$\max \sum_i \tilde{p}_i^E(1 + \frac{2\tilde{S}_i - S_i}{2\epsilon_i\tilde{S}_i}) - \tilde{p}_x x_E - \tilde{p}_y y_E \tag{3.5.18}$$

subject to

$$S_i = \sum_t z_{it} - \sum_{i't} b_{ii't} z_{i't}$$

$$x_E = \sum_{it} a_{it}^x z_{it}$$

$$y_E = \sum_{it} a_{it}^y z_{it}$$

$$0 \leq z_{it} \leq \bar{z}_{it}$$

The following table (3.2) summarizes the additional variables and parameters appearing in the decomposed model.

Variables	
S_i	Net supply of energy good i
x_E, y_E	Aggregate demand for x and y as inputs to energy production
$z_{i,t}$	Activity level of technology t producing energy good i
Parameters	
$\tilde{S}_i$	Reference level of demand or supply for energy good i
$\tilde{x}_E, \tilde{y}_E$	Reference demand of non-energy inputs to energy supply x and y
$\tilde{p}_i^E$	Reference price of energy good i
$\tilde{p}_x, \tilde{p}_y$	Reference prices of non-energy goods x and y
b_{ijt}	Base year intermediate inputs of energy good j to energy good i with technology t
ϵ_i	Demand elasticity for energy good i

Table 3.2: Variables and Parameters in the Decomposed Model

Notice that in Böhringer and Rutherford (2009)'s *bottom-up* model, energy demand is calibrated based on demand elasticities for good i, ϵ_i, which is an own-price elasticity. Own-price elasticity in this case measures only own-price impacts on demand of energy good i.

3.5.3 Block Decomposition on BR example

Next, we apply block decomposition method on the *top-down/bottom-up* example. We call the *top-down* model as block G and the *bottom-up* model as block H. In model implementation of Böhringer and Rutherford (2009), the response functions $g(\cdot)$ and $h(\cdot)$ are selected on the basis of the underlying economic model. Note that function $g(\cdot)$ defines how energy goods in block H will be affected by the general economy, especially by the price changes of energy goods block G. It portrays energy demands in the *bottom-up* model (block H) through linear demand schedules in which they ignore cross-price effects, hence $g(x; x^k, y^k)$ is calibrated on the current estimates of energy demand and energy price, together with a fixed energy demand elasticity. And we use `LSA` to capture the potential impacts on energy demand caused by other energy goods. When Böhringer and Rutherford (2009) bring the energy supply responses back into the economic model (block G), they use a straight diagonalization approach in which energy netputs are fixed exogenously and $h(y; x^k, y^k) = x^k$ according to equation (3.3.13). While in this paper, we leave the assumption on energy supply responses alone and focus on improving the calibration of *bottom-up* energy demand. We rely on the calibrated approximation function (3.3.12) to define energy netputs in *bottom-up* energy models. And we calculate and update own- and cross-price elasticities contained in response matrix A in each iteration to make sure: First, demand elasticities are not fixed but endogenously decided in the previous iteration; Second, cross-price demand effects are considered.

Next, we implement several alternative *bottom-up* models, including some using fixed own-price elasticities and others using a full demand system.

On the supply side of energy goods, according to the quadratic programming of the *bottom-up* model, we define supply of energy good i as:

$$S_i = \sum_t z_{it} - \sum_{jt} b_{ijt} z_{jt} \tag{3.5.19}$$

where z_{it} indicates the energy technology t in good i's production, and b_{ijt} are base year intermediate inputs of energy good j to energy good i with technology t.

On the demand side, we set up different form of demand functions based on own-price and cross-price elasticities as follows:

$$D_i = \begin{cases} \bar{E}_i(1 - \epsilon_i(\frac{P_i}{\bar{P}_i} - 1)) & Own - price - elasticities \\ \bar{D}_i - \Sigma_j \frac{\partial E_i}{\partial P_j}\left(P_j - \bar{P}_j\right) & Cross - price - elasticities \end{cases} \tag{3.5.20}$$

in which $\frac{\partial E_i}{\partial P_j}$ measures first order derivative of energy good supply in the *top-down* model with respect to price P_i, ϵ_i is the own-price elasticities.

On market level, we let $D_i = S_i$ for each i. We then have a MCP model for the *bottom-up* energy system. As noted before, we decompose the model by solving variables block G which is defined

$$x = h(y; x^k, y^k) \equiv x^k, \tag{3.5.21}$$

and variables in block H:

$$y = g(x; x^k, y^k) \equiv y^k + A(x - x^k), \tag{3.5.22}$$

through iteration. In each loop when solving the *bottom-up* model (block H), we calculate derivatives of energy demand of good i to all energy prices in the *top-down* model and updated sensitivity function $g(\cdot)$ through updating matrix A. It is easy to see that block decomposition using own-and cross-price elasticities captures more interactions between goods thus might lead to more accurate decomposition over the solving procedure using own-price elasticities. And replacing the exogenous elasticities with elasticities based on calculated gradients itself is valuable.The cost is additional time spent on deriving the whole `Jacobian` matrix of demands with respect to prices instead of just taking the diagonal elements of that matrix. In practice, for this small scale model, *bottom-up* demand using cross-price elasticities converges as fast as before when we calibrate demand using fixed own-price elasticities. Here is a list of iteration loops assuming that a technological breakthrough makes non-fossil electricity 50% cheaper. And we measure the accuracy of the decomposition by measuring the absolute difference between the solutions of decomposed model and the original integrated MCP models. For more details about the GAMS code of this example, please refer to the Appendix.

```
----    Evaluate precision of the bottomup solution


    own-price  cross-price


1       0.021        0.021

2 3.023881E-4 3.025586E-4

3 4.394040E-5 4.394048E-5

4 6.375617E-6 6.375615E-6
```

```
5 9.291437E-7 9.291438E-7
6 1.310007E-7 1.310007E-7
```

3.6 Conclusion

In this paper, We apply Local Sensitivity Analysis (`LSA`) to block decomposition approach and its application in *top-down/bottom-up* fashion energy-economy model. We show block decomposition algorithm through numerical examples, comparing this approach to other methods and we apply this new approach to the *top-down/bottom-up* discussed by Böhringer and Rutherford (2009). One improvement we achieve through block decomposition with `LSA` is that the our approach captures the full demand sensitivity matrix with respect to energy prices in the *bottom-up* partial equilibrium model, which is potentially useful in future energy modeling.

Chapter 4

Appendix

4.1 Appendix for Chapter 1

This appendix describes how we calculate first/second order derivatives in a general equilibrium model using LSA.

4.1.1 Tools for Sensitivity Analysis in GAMS

For CGE models in GAMS, local sensitivity calculation through LSA is currently available for Mixed Complementarity Problems (MCP). We present next how this works in a trade model. Remember that in the model $z \in \mathrm{R}^{\mathrm{n}}$ is a vector of equilibrium variables and $t \in \mathrm{R}^{\mathrm{m}}$ is a vector of strategic policy variables and we are target the dependence of z_j $(j = 1, \dots, n)$ on t_k $(k = 1, \dots, m)$. In order to compute derivatives, we first invoke LSA using the conventional $libinclude syntax. Here is an example in which we solve an MCP model and then retrieve first/second order derivatives. Following the solve statement, the libinclude call LSA requests specific model name, endogenous variables (var) and instrument variables (ins) at the solution point. And we report first order derivatives in the form of <var>_<ins> and <var>_<ins>_<ins> for second order derivatives. However, if users only need first order derivatives, they could use the command $set lsahessian no to turn off the second order derivatives calculation to save computational time.

```
$set lsahessian no
$libinclude lsa <model_name> <var> <ins>
execute_load '%gams.scrdir%lsa.gdx', <var>_<ins>;
```

When second order derivatives are desired, we turn on `lsahessian` before invoking the `LSA` routine:

```
$set lsahessian yes
$libinclude lsa <model_name> <var> <ins>
execute_load '%gams.scrdir%lsa.gdx', <var>_<ins>, <var>_<ins>_<ins>;
```

We thus have tariff impacts on welfare $\frac{\partial c_r}{\partial \tau_{i,s}}$ calculated as c_tau and second order derivatives of welfare with respect to tariff $\frac{\partial^2 c_r}{\partial \tau_{i,s} \partial \tau_{i',s'}}$ as c_tau_tau using `LSA`. Next, we would introduce an example in a trade model combining data and local sensitivity calculation together.

4.1.2 A GAMS Interface to LUSOL

One key element in `LSA` is `LUSOL` and here is some introduction of this. `LUSOL` is a set of `Fortran` 90 routines written by *Michael Saunders* at Stanford University. `LUSOL` maintains LU factors of a square or rectangular sparse matrix. The initial implementation in GAMS only processes square matrices. Here is a GAMS program which uses `LUSOL` to factorize and solve a linear system:

```
$title  Solve a Linear System using LUSOL

set     i       Row indices /1*3/
        j       Column indices /a*c/

table   a(i,j)  Matrix for to be factorized

                a       b       c
        1       10      -1      3
```

```
        2       2       12      -2
        3       11      -3      15;

parameter       b(i)    RHS vector /
        1       10
        2       2
        3       -4/;

parameter       x(j)    Solution vector which solves A x = b;

$batinclude lusolve a b x

parameter       chk(i)  Cross check on the calculation;

chk(i) = sum(j, a(i,j)*x(j)) - b(i);

display a, b, x, chk;
```

In this example, LUSOL computes a factorization

$$A = LU,$$

and it then solves the linear system

$$Ax = b$$

using the factorization. The listing file contains output from the display statement:

```
----     70 PARAMETER a  Matrix for to be factorized

            a           b           c
```

```
1      10.000      -1.000       3.000
2       2.000      12.000      -2.000
3      11.000      -3.000      15.000


----      70 PARAMETER b  RHS vector


1 10.000,    2  2.000,    3 -4.000


----      70 PARAMETER x  Solution vector which solves A x = b


a  1.370,    b -0.283,    c -1.328


----      70 PARAMETER chk  Cross check on the calculation


1 4.50129E-12,    2 4.44089E-16,    3 1.05000E-11
```

The interface permits separation of the factorization and solution phases. In addition, the GAMS interface permits solution of a set of RHS vectors. For example, the previous GAMS program could have been written as:

```
set     k       RHS vectors /k1*k3/;


table   b(i,k)  RHS matrix


                k1      k2      k3
        1       10      2       -10
        2       2       1       100
        3       -4      20      1;


parameter       x(j,k)  Solution vector which solves A x = b;
```

```
$batinclude lufactor a

$batinclude lusolve  b x

set     k       First RHS index /k1*k3/
        m       Second RHS index /m1*m2/;

table   b(i,k,m)        RHS matrix

                k1.m1   k2.m1   k3.m1   k1.m2   k2.m2   k3.m2
        1       10      2       -10     10      2       -4
        2       2       1       100     2       1       10
        3       -4      20      1       -3      20      2;

parameter       x(j,k,m)    Solution vector which solves A x = b;

$batinclude lusolve  a b x
```

Because multiple RHS vectors phases. In addition, the GAMS interface permits solution of a set of RHS vectors. For example, the previous GAMS program could have been written as:

```
table   ident(i,i)      The identity matrix

                1       2       3
        1       1
        2               1
        3                       1;
```

```
parameter       ainv(j,i)       Inverse of A;


$batinclude lusolve  a ident ainv
```

4.2 Appendix for Chapter 3

4.2.1 GAMS code in the Top-down/Bottom-up model

We document modified Top-down/Bottom-up code using LSA based on the small scale model in Böhringer and Rutherford (2009) as follows:

```
$title  Illustration of a Decomposition Method for Top-Down / Bottom-Up Models
option qcp=cplex;


*       Define the number of technologies here:
$if not set ntech $set ntech 100


*        The %intensity% input to the model can be used to change the
*        assumed energy intensity of final demand:
*        Set energy intensity of final demand:
$if not set intensity $set intensity 10


set     t        Technologies /1*%ntech%/,
        i        Energy goods /ele, oil, gas, col/,
        ele(i)        /ele/,
        oil(i)        /oil/,
        gas(i)        /gas/,
        col(i)        /col/;


alias (i,ii,iii);
```

```
*       Here are the social accounts.  The economy is fairly energy-intensive
*       with a value share of 10% in final demand, 33% in energy-intensive
*       manufacturing (X) and 10% in other industrial (Y).

table   sam     Base year social accounts
                X       Y       EN      FD
        X       15                      -15
        Y               100     -24     -76
        L       -5      -40             45
        K       -5      -50             55
*       -----------------------------------
        ELE     -2      -5      10      -3
        OIL     -1      -2      7       -4
        GAS     -1      -1      3       -1
        COL     -1      -2      4       -1

parameters
        ce0(i,t)        Base year energy cost,
        ec0(i)          Base year energy expenditures in final consumption,
        ex0(i)          Base year energy expenditures in sector X,
        ey0(i)          Base year energy expenditures in sector Y,
        c0              Base year final consumption,
        cy0             Base year consumption of y,
        ay0(i,t)        Unit cost of energy technology,
        ae0(ii,i,t)     Base year intermediate inputs to energy technology,
        ecap(i,t)       Energy capacity,
        re0(i,t)        Rental price on energy capacity,
```

```
        capscale(i)     Scale factor for energy sector capacity,
        ecost0          Aggregate energy cost;

*        Extract SAM:
ex0(i) = -sam(i,"X");
ey0(i) = -sam(i,"Y");
ec0(i) = -sam(i,"FD") * %intensity% / 10;

*        Randomly generate costs distribution such that around 33% of the
*        technologies are idle at the benchmark price of unity:
ce0(i,t) = uniform(0.8,1.1);

*        Randomly sort electricity technologies into coal, gas and "other":
set     coaltech(t)     Coal technology,
        gastech(t)      Gas technology;

coaltech(t) = yes$(ord(t) < card(t)/2);
gastech(t)  = yes$(ord(t) > 0.8*card(t));

*        Assign some fuel costs to coal and gas power plots.  Assume
*        that capital is a higher value share of coal plants:
ae0("col","ele",coaltech) = uniform(0.3,0.6) * ce0("ele",coaltech);
ae0("gas","ele",gastech)  = uniform(0.8,0.9) * ce0("ele",gastech);

*        Other variable costs are taken from the market for Y:
ay0(i,t) = ce0(i,t) - sum(ii, ae0(ii,i,t));

*        Capital earnings are determine as a residual.  If an activity is
```

```
*        not profitable, the return to that capital stock is zero:
re0(i,t) = max(1 - ce0(i,t), 0);

*        Now we randomly generate some relative capacities:
ecap(i,t) = uniform(0,1);

*        Once we have relative capacities, we need to scale these such that
*        net supply equals macro economic demand.  First do electricity so that
*        we can then infer fuel input to power generation:
loop(ele(i),
        capscale(i) = (ex0(i)+ey0(i)+ec0(i)) / sum(t$(ce0(i,t) <= 1), ecap(i,t));
        ecap(i,t)   = ecap(i,t) * capscale(i);
);

*        Gas and coal inputs to electricity are then added to aggregate demand
*        when supply capacities for gas and coal are scaled:
loop(i$(not ele(i)),
        capscale(i) = (ex0(i)+ey0(i)+ec0(i) + sum(t$re0("ele",t),
                        ae0(i,"ele",t) * ecap("ele",t)))
                        / sum(t$(ce0(i,t) <= 1), ecap(i,t));
        ecap(i,t) = ecap(i,t) * capscale(i);
);

*        Assign some parameters for a more transparent write-down of the model:

ecost0  = sum((i,t)$re0(i,t), ecap(i,t) * ay0(i,t));
cy0     = 100 - ecost0;
c0      = 15 + cy0 + sum(i, ec0(i));
```

```
parameter       peref(i)        Reference energy prices,
                eref(i)         Reference energy supplies;
eref(i) =   sum(t$(ce0(i,t) < 1),
                ecap(i,t)) - sum((ii,t)$(ce0(ii,t) < 1), ae0(i,ii,t)*ecap(ii,t));

*       Top-down formulation in which energy supplies
*       and energy costs are exogenous inputs:

nonnegative variables
*        Activity levels
        C                Final consumption,
        X                Energy-intensive,
        Y                Other goods and services,
        E_C             Composite energy good for final consumption,
        E_X             Composite energy good for energy-intensive sector,
        E_Y             Composite energy good for other goods and services,
        E(i,t)          Energy technologies,
*        Prices
        PC              Price of consumption,
        PX              X sector output,
        PY              Y sector output,
        PE(i)           Energy prices,
        PL              Wage rate,
        RX              Return to X sector capital,
        RY              Return to Y sector capital,
        RE(i,t)         Return to energy sector capacity,
        PE_X            Composite energy good of X sector,
```

```
        PE_Y            Composite energy good of Y sector,
        PE_C            Composite energy good of final consumption,
*        Income level
        RA                Representative agent,
*        Energy supply
        ESUP(i)        Current iteration energy supply for the top-down model;

equations
*       Zero profit conditions for activities linked to activity levels
        zprf_x          Zero profit condition for X (EIS) production sector,
        zprf_y          Zero profit condition for Y (ROI) production sector,
        zprf_c          Zero profit condition for provision of aggregate consumption good,
        zprf_e(i,t)     Zero profit condition for energy production technologies,
        zprf_e_x        Zero profit condition for
                        provision of composite energy good of X sector,
        zprf_e_y        Zero profit condition for
                        provision of composite energy good of y sector,
        zprf_e_c        Zero profit condition for
                        provision of composite energy good in final consumption,

*        Market clearance condiditions for goods linked to prices
        mkt_px          Market clearance condition for X good,
        mkt_py_TD       Market clearance condition for Y good,
        mkt_pe_TD(i)    Market clearance condition for electricity,
        mkt_pc          Market clearance condition for aggregate consumption good,
        mkt_pe_x        Market clearance for X sector energy composite,
        mkt_pe_y        Market clearance for Y sector energy composite,
        mkt_pe_c        Market clearance for final demand energy composite,
```

```
        mkt_pl          Market clearance condition for labor,
        mkt_rx          Market clearance condition for specific capital in X sector,
        mkt_ry          Market clearance condition for specific capital in Y sector,
        mkt_re(i,t)     Market clearance condition for technolgy specific capital,

*        Income balance for representative agent linked to income level
        inc_ra_TD        Budget constraint;

*        Definition of zero profit conditions
zprf_x.. ((10/15)*((PL**0.5)*(RX**0.5))**(1-0.5) +
         (sum(i, ex0(i))/15)*(PE_X**(1-0.5)))**(1/(1-0.5)) =G= PX;

zprf_e_x.. prod(i$ele(i), PE(i)**( ex0(i)/sum(ii, ex0(ii)) ))
        * ([  sum( i$col(i),  ex0(i)/ sum(ii$(not ele(ii)),
                        ex0(ii))*PE(i)**(1-0.5) )
            + sum( i$(oil(i) or gas(i)),
                        ex0(i) )/sum(i$(not ele(i)), ex0(i))
            * prod(i$(oil(i) or gas(i)),
                        PE(i)**(ex0(i)/sum(ii$(oil(ii) or gas(ii)), ex0(ii))))**(1-0.5)
           ]**(1/(1-0.5)))**( sum(i$(not ele(i)),
                        ex0(i))/sum(i, ex0(i)) )
           =G= PE_X;

zprf_y..  ((90/100)*((PL**(40/90))*(RY**(50/90)))**(1-0.5) +
               (10/100)*(PE_Y**(1-0.5)))**(1/(1-0.5)) =G= PY;

zprf_e_y.. prod(i$ele(i),
                        PE(i)**(ey0(i)/sum(ii, ey0(ii))))
```

```
        * ([  sum( i$col(i),
                        ey0(i)/ sum(ii$(not ele(ii)), ey0(ii))*PE(i)**(1-0.5) )
            + sum( i$(oil(i) or gas(i)),
                        ey0(i) )/sum(i$(not ele(i)), ey0(i))
            * prod(i$(oil(i) or gas(i)),
                        PE(i)**(ey0(i)/sum(ii$(oil(ii) or gas(ii)), ey0(ii))))**(1-0.5)
           ]**(1/(1-0.5)))**( sum(i$(not ele(i)),
                        ey0(i))/sum(i, ey0(i)) )
           =G= PE_Y;

zprf_c..  ( sum(i, ec0(i))/c0*PE_C**(1-0.5)
          + (15 + cy0)/c0 * (PX**(15/(15+cy0))*PY**(cy0/(15+cy0)))**(1-0.5)
          )**(1/(1-0.5)) =G= PC;

zprf_e_c.. ( sum(i, ec0(i)/sum(ii,
                        ec0(ii))*PE(i)**(1-0.8)) )**(1/(1-0.8)) =G= PE_C;

zprf_e(i,t)..        ay0(i,t)*PY + sum( ii,
                        ae0(ii,i,t)*PE(ii) ) + RE(i,t) =G= PE(i);

mkt_pe_x..        sum(i, ex0(i))*E_X  =G= sum(i, ex0(i))*X*(PX/PE_X)**0.5;

mkt_pe_y..        sum(i, ey0(i))*E_Y  =G= sum(i, ey0(i))*Y*(PY/PE_Y)**0.5;

mkt_pe_c..        sum(i, ec0(i))*E_C  =G= sum(i, ec0(i))*C*(PC/PE_C)**0.5;

mkt_pc$(PC.LO < PC.UP)..        c0*C*PC =G= RA;
```

```
mkt_px..  15*X =G= 15*C*( PC/(PX**(15/(15+cy0))*
                         PY**(cy0/(15+cy0))) )**0.5
            *( PX**(15/(15+cy0))*PY**(cy0/(15+cy0)) )/PX ;


mkt_pl..  45 =G= 5*X*( PX/((PL**0.5)*(RX**0.5)) )**0.5*
                         ( (PL**0.5)*(RX**0.5) )/PL
           + 40*Y*( PY/((PL**(40/90))*(RY**(50/90))) )**0.5*
                         ( (PL**(40/90))*(RY**(50/90)) )/PL;


mkt_rx..  5 =G= 5*X*( PX/((PL**0.5)*(RX**0.5)) )**0.5*
                         ( (PL**0.5)*(RX**0.5) )/RX;


mkt_ry..  50 =G= 50*Y*( PY/((PL**(40/90))*(RY**(50/90))) )**0.5*
                         ( (PL**(40/90))*(RY**(50/90)) )/RY;


mkt_re(i,t)..        ECAP(i,t) =G= E(i,t);


mkt_py_TD$(PY.LO < PY.UP)..
          100*Y + (-ecost0) =G=
                         cy0*C*( PC/(PX**(15/(15+cy0))*PY**(cy0/(15+cy0))) )**0.5
                         *(PX**(15/(15+cy0))*PY**(cy0/(15+cy0)))/PY;
mkt_pe_TD(i) ..


        ESUP(i) =G=


      ec0(i)*E_C*(PE_C/PE(i))**0.8


  + ( ex0(i)*E_X*(PE_X/PE(i)) + ey0(i)*E_Y*(PE_Y/PE(i)) )$ele(i)
```

```
+ [ ex0(i)*E_X*( PE_X/ [ sum(ii$col(ii),
     ex0(ii)/ sum(iii$(not ele(iii)), ex0(iii))*PE(ii)**(1-0.5) )
+ sum(ii$(oil(ii) or gas(ii)),
     ex0(ii))/sum(ii$(not ele(ii)), ex0(ii))
* prod(ii$(oil(ii) or gas(ii)),
     PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)),
              ex0(iii))))**(1-0.5)
 ]**(1/(1-0.5))
*( [  sum( ii$col(ii),
      ex0(ii)/ sum(iii$(not ele(iii)), ex0(iii))*PE(ii)**(1-0.5) )
+ sum(ii$(oil(ii) or gas(ii)),
      ex0(ii))/sum(ii$(not ele(ii)), ex0(ii))
* prod(ii$(oil(ii) or gas(ii)),
      PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)), ex0(iii))))**(1-0.5)
 ]**(1/(1-0.5))/PE(i))**0.5
  )

+   ey0(i)*E_Y*( PE_Y/ [sum(ii$col(ii),
      ey0(ii)/ sum(iii$(not ele(iii)), ey0(iii))*PE(ii)**(1-0.5) )
+ sum(ii$(oil(ii) or gas(ii)),
      ey0(ii))/sum(ii$(not ele(ii)), ey0(ii))
* prod(ii$(oil(ii) or gas(ii)),
      PE(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)), ey0(iii))))**(1-0.5)
]**(1/(1-0.5))
*([  sum(ii$col(ii),
      ey0(ii)/ sum(iii$(not ele(iii)), ey0(iii))*PE(ii)**(1-0.5) )
+ sum(ii$(oil(ii) or gas(ii)),
```

```
      ey0(ii))/sum(ii$(not ele(ii)), ey0(ii))
* prod(ii$(oil(ii) or gas(ii)),
      PE(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)), ey0(iii))))**(1-0.5)
]**(1/(1-0.5))/PE(i))**0.5
  )]$col(i)

+   [        ex0(i)*E_X*
PE_X / ( sum(ii$col(ii),
      ex0(ii))/sum(ii$(not ele(ii)), ex0(ii))*
      sum(ii$col(ii), PE(ii)**(1-0.5))
+ sum(ii$(oil(ii) or gas(ii)),
      ex0(ii))/sum(ii$(not ele(ii)), ex0(ii))
*(prod( ii$(oil(ii) or gas(ii)),
      PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)),
              ex0(iii))) ))**(1-0.5)
 )**(1/(1-0.5))
* ( (sum(ii$col(ii),
      ex0(ii))/sum(ii$(not ele(ii)),
      ex0(ii))*sum(ii$col(ii), PE(ii)**(1-0.5))
+ sum(ii$(oil(ii) or gas(ii)),
      ex0(ii))/sum(ii$(not ele(ii)), ex0(ii))
*(prod( ii$(oil(ii) or gas(ii)),
      PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ex0(iii))) ))**(1-0.5)
)**(1/(1-0.5))/
prod( ii$(oil(ii) or gas(ii)),
      PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ex0(iii))) )) **0.5
```

```
*prod( ii$(oil(ii) or gas(ii)),
      PE(ii)**(ex0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ex0(iii))) ) /PE(i)


+           ey0(i)* E_Y *
PE_Y / ( sum(ii$col(ii),
      ey0(ii))/sum(ii$(not ele(ii)),
      ey0(ii))*sum(ii$col(ii), PE(ii)**(1-0.5))
+ sum(ii$(oil(ii) or gas(ii)),
      ey0(ii))/sum(ii$(not ele(ii)), ey0(ii))
*(prod(ii$(oil(ii) or gas(ii)),
      pe(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ey0(iii))) ) )**(1-0.5)
)**(1/(1-0.5))
* (( sum(ii$col(ii), ey0(ii))/sum(ii$(not ele(ii)),
      ey0(ii))*sum(ii$col(ii), pe(ii)**(1-0.5))
+ sum(ii$(oil(ii) or gas(ii)),
      ey0(ii))/sum(ii$(not ele(ii)), ey0(ii))
*(prod( ii$(oil(ii) or gas(ii)),
      pe(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ey0(iii))) ))**(1-0.5)
)**(1/(1-0.5)) /
prod( ii$(oil(ii) or gas(ii)),
      pe(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ey0(iii))) )) **0.5
* prod( ii$(oil(ii) or gas(ii)),
      pe(ii)**(ey0(ii)/sum(iii$(oil(iii) or gas(iii)),
      ey0(iii))) )/pe(i)
```

```
  ]$(oil(i) or gas(i));

inc_ra_TD..  45*PL + 5*RX + 50*RY + sum(i, ESUP(i)*PE(i)) - ecost0*PY =G= RA;

*        Initialize activities and prices
C.L    = 1;
X.L    = 1;
Y.L    = 1;

PC.L   = 1;
PX.L   = 1;
PY.L   = 1;

E_C.L  = 1;
E_X.L  = 1;
E_Y.L  = 1;

PE_C.L = 1;
PE_X.L = 1;
PE_Y.L = 1;

PL.L   = 1;
RX.L   = 1;
RY.L   = 1;

PE.L(i)   = 1;
RE.L(i,t) = re0(i,t);
E.L(i,t)  = ecap(i,t)$(ce0(i,t) < 1);
```

```
RA.L          = 45*PL.L + 5*RX.L + 50*RY.L + sum((i,t), ecap(i,t)*RE.L(i,t));

model topdown /zprf_x.X, zprf_y.Y, zprf_c.C,
               zprf_e_x.E_X, zprf_e_y.E_Y, zprf_e_c.E_C,
               mkt_px.PX, mkt_py_td.PY, mkt_pc.PC,
               mkt_pe_x.PE_X,mkt_pe_y.PE_Y, mkt_pe_c.PE_C, mkt_pe_td.PE,
               mkt_pl.PL, mkt_rx.RX, mkt_ry.RY, inc_ra_td.RA/;

*        Define a numeraire:
PY.FX = 1;

*        Supply of energy is fixed in the TD model
ESUP.FX(i) = eref(i);

topdown.iterlim = 10000;
solve topdown using mcp;
abort$round(topdown.objval,4)        "Benchmark calibration fails.";

set        mask(i,ii)        Mask for gradient calculations;
mask(i,ii) = yes;

parameter  dpdq(i,ii)        Local sensitivity of PE on ESUP;
$batinclude ..\inclib\lsa topdown PE ESUP mask
execute_load '%gams.scrdir%lsa.gdx', dpdq=pe_esup;

parameter        peref(i)        Reference price level;
peref(i) = PE.L(i) / PY.L;
```

```
*        Construct the matrix of own and cross-price elasticities
*        even though we don't need to use these in our demand function:

parameter
        dqdp(i,ii)        Derivative of E(i) wrt P(ii),
        ident(i,ii)        Indentity matrix;

*        Find inverse of the dpdq
ident(i,i) = 1;
execute_unload 'lusol_in.gdx',dpdq=a,ident=b;
execute '..\inclib\lusol lusol_in.gdx lusol_out.gdx';
execute_load 'lusol_out.gdx',dqdp=x;

parameter
        epsilon(i,ii)   Own and cross price elasticities of demand,
        eta(i)          Elasticity of energy demand
                                /ele        0.5,
                                 oil        0.75,
                                 gas        0.75,
                                 col        0.5/;
epsilon(i,ii) = dqdp(i,ii) * PE.L(ii) / eref(i);
option epsilon:6:1:1;
display dqdp, dpdq, epsilon;

*       Declare the Quadratic Programming (QP) model representing
*       the bottom-up component of the model:

variable        PI              Profit (sum of producer and consumer surplus),
```

```
                COST            Cost,
                S(i)            Energy supply;

nonnegative
variables       ET(i,t)         Energy technologies;

equations       costdef         Defines cost of production,
                pidef           Defines profit,
                supply(i)       Net energy supply;

parameter       diagonal        Flag for diagonal demand function   /1/,
                isoelastic      Flag for isoelastic demand function /0/;

pidef..         PI  =e=  sum(i, S(i) * peref(i)) - COST +
                             sum(i, sqr(S(i)-eref(i))*dpdq(i,i)/2)$diagonal +
                             sum((i,ii),
                                (S(i)-eref(i)) *
                                (S(ii)-eref(ii)) * dpdq(i,ii)/2)$(not diagonal);

supply(i)..     S(i)  =e=  sum(t, ET(i,t)) - sum((ii,t), ET(ii,t) * ae0(i,ii,t));

costdef..        COST  =e=  sum((i,t), ET(i,t) * ay0(i,t));

model bottomup /pidef, costdef, supply/;
bottomup.iterlim = 10000;

*        Nonzero diagonal means no cross-price effects in demand
*        Nonzero isoelastic means isoelastic demand function
```

```
nonnegative
variable        DE(i)           Energy demand;

equation        mkt_pe_BU        Market clearance condition for electricity,
                dedef                Definition of energy demand in certain form,
                iedef                Definition of energy demand in certain form,
                netput          Net energy supply;

mkt_pe_BU(i)..  S(i) =e=
                (eref(i)*(1-eta(i)*(PE(i)/peref(i)-1)))$diagonal +
                DE(i)$(not diagonal);

dedef(i)$((not diagonal) and (not isoelastic))..
                sum(ii, dpdq(i,ii) * (DE(ii)-eref(ii))) =e= PE(i)-peref(i);

iedef(i)$((not diagonal) and (isoelastic))..
                prod(ii, (DE(ii)/eref(ii))**
                (dpdq(i,ii)*eref(ii)/peref(i)))  =e= PE(i)/peref(i);

netput(i)..     S(i) =e= sum(t, E(i,t)) - sum((ii,t), E(ii,t)*ae0(i,ii,t));

model bottomup_mcp/zprf_e.E, mkt_re.RE,
                   mkt_pe_BU.PE, dedef.DE, iedef.DE, netput.S/;

ET.L(i,t)     = ecap(i,t)$(ce0(i,t) < 1);
ET.UP(i,t)    = ecap(i,t);
```

```
*        Perform an illustrative policy shock to demonstrate
*        the equivalence of the decomposed model with the
*        integrated model:

*        Assume that a technological breakthrough makes non-fossil
*        electricity 50% cheaper:
ay0("ele",t)$(not (coaltech(t) or gastech(t))) = ay0("ele",t)/2;

parameter       decomplog       Decomposition log,
                bottomup_log    Iteration log of bottomup solution,
                precision       Evaluate precision of the bottomup solution,
                pesol(i)        Energy prices in solution to the bottomup model;

set                decompiter      Decomposition iterations /1*6/;

option solvelink = 2;
*        -------------------------------------------------------------------
*        The following code implements several alternative bottom up models,
*        including some which are diagonalized and others which use a full
*        demand system:

*        Initiate at the reference point:
COST.L  = sum((i,t)$re0(i,t), ecap(i,t) * ay0(i,t));
S.L(i)  = sum(t$(ce0(i,t) < 1), ecap(i,t)) -
          sum((ii,t)$(ce0(ii,t) < 1), ae0(i,ii,t)*ecap(ii,t));

loop(decompiter,
*       Begin the iteration with a computation of the top-down model:
```

```
        ecost0           = COST.L;
        ESUP.FX(i) = S.L(i);
        solve topdown using mcp;
*        Save the reference price and quantity at this point:
        peref(i) = PE.L(i) / PY.L;
        eref(i)  = ESUP.L(i);
*        Extract local elasticities:
$batinclude ..\inclib\lsa topdown PE ESUP mask
execute_load '%gams.scrdir%lsa.gdx', dpdq=pe_esup;
*        ------------------------------------------------------------------
*        Solve the bottom up model using exogenous elasticities:
        display eta;
        solve bottomup using qcp maximizing PI;
        bottomup_log(decompiter,i,"eta fixed","PE") = SUPPLY.M(i);
        bottomup_log(decompiter,i,"eta fixed","S")  = S.L(i);
*        Evaluate precision of this bottomup solution:
        pesol(i)        = SUPPLY.M(i);
        ecost0          = COST.L;
        ESUP.FX(i)      = S.L(i);
        solve topdown using mcp;
        precision(decompiter,"eta fixed") =
                sum(i, abs(pesol(i)-PE.L(i)));
*        ------------------------------------------------------------------
*        Do another bottomup calculation using
*        elasticities of demand at the reference point
*        -----------------------------------------------------------------
*        Find inverse of dpdp to get
*       dqdp based on the Inverse Function Theorem:
```

```
        ESUP.FX(i) = eref(i);
        solve topdown using mcp;
$batinclude ..\inclib\lsa topdown PE ESUP mask
execute_load '%gams.scrdir%lsa.gdx', dpdq=pe_esup;
*        Find inverse of dpdq
execute_unload 'lusol_in.gdx',dpdq=a,ident=b;
execute '..\inclib\lusol lusol_in.gdx lusol_out.gdx';
execute_load 'lusol_out.gdx',dqdp=x;
        eta(i) = -dqdp(i,i)*peref(i)/eref(i);
        display eta;
        solve bottomup using qcp maximizing PI;
        bottomup_log(decompiter,i,"eta calc","PE") = SUPPLY.M(i);
        bottomup_log(decompiter,i,"eta calc","S")  = S.L(i);
*        ---------------------------------------------------------------
*        Evaluate precision of the bottomup solution:
        pesol(i)        = SUPPLY.M(i);
        ecost0          = COST.L;
        ESUP.FX(i)      = S.L(i);
        solve topdown using mcp;
        precision(decompiter,"eta calc") = sum(i, abs(pesol(i)-PE.L(i)));
*        ---------------------------------------------------------------
*        Verify that the MCP version of the bottom-up model replicates
*        the NLP model at the benchmark point
        E.L(i,t)        = ET.L(i,t);
        PE.L(i)         = SUPPLY.M(i);
        RE.L(i,t)       = max(0, ET.M(i,t));
        DE.L(i)         = S.L(i);
        diagonal        = 1;
```

```
        bottomup_mcp.iterlim = 0;
*        solve bottomup_mcp using mcp;
*        abort$round(bottomup_mcp.objval,4) "MCP fails to replicate the NLP solution.";
*        ------------------------------------------------------------------
*        Solve the bottomup model with the asymmetric
*        linear demand system:
        diagonal  = 0;
        bottomup_mcp.iterlim = 10000;
        solve bottomup_mcp using mcp;
        bottomup_log(decompiter,i,"asymmetric","PE") = PE.L(i);
        bottomup_log(decompiter,i,"asymmetric","S")  = S.L(i);
        pesol(i)   = PE.L(i);
        ecost0           = COST.L;
        ESUP.FX(i) = S.L(i);
        solve topdown using mcp;
        precision(decompiter,"asymmetric") = sum(i, abs(pesol(i)-PE.L(i)));
*        ------------------------------------------------------------------
*        Solve the bottomup model with the asymmetric isoelastic
*        demand system:
        isoelastic      = 1;
        diagonal        = 0;
        bottomup_mcp.iterlim = 10000;
        solve bottomup_mcp using mcp;
        bottomup_log(decompiter,i,"isoelastic","PE") = PE.L(i);
        bottomup_log(decompiter,i,"isoelstic","S")   = S.L(i);
        pesol(i)   = PE.L(i);
        ecost0     = COST.L;
        ESUP.FX(i) = S.L(i);
```

```
        solve topdown using mcp;
        precision(decompiter,"isoelastic") = sum(i, abs(pesol(i)-PE.L(i)));
        isoelastic = 0;
*        -----------------------------------------------------------------
*        Solve the bottomup model with the symmetric system, first by
*        maximizing the sum of consumer and producer surplus:
        dpdq(i,ii) = (dpdq(i,ii) + dpdq(ii,i))/2;
        diagonal = 0;
        solve bottomup using qcp maximizing PI;
        bottomup_log(decompiter,i,"symmetric","PE") = SUPPLY.M(i);
        bottomup_log(decompiter,i,"symmetric","S")  = S.L(i);
*        -----------------------------------------------------------------
*        Verify consistency between the NLP formulation and the
*        MCP model:
        E.L(i,t)  = ET.L(i,t);
        PY.FX     = PY.L;
        PE.L(i)   = SUPPLY.M(i);
        RE.L(i,t) = max(0, ET.M(i,t));
        DE.L(i)   = S.L(i);
        bottomup_mcp.iterlim = 0;
        solve bottomup_mcp using mcp;
        abort$round(bottomup_mcp.objval,4)
        "MCP symmetric demand model fails to replicate the NLP solution.";
        pesol(i)  = SUPPLY.M(i);
        ecost0    = COST.L;
        ESUP.FX(i)= S.L(i);
        solve topdown using mcp;
        precision(decompiter,"symmetric")
```

```
                = sum(i, abs(pesol(i)-PE.L(i)));
*        -----------------------------------------------------------------
*        Then verify consistency of the NLP
*       (surplus maximization) model and
*        the MCP (economic equilibrium) model:
        E.L(i,t)  = ET.L(i,t);
        PY.FX     = PY.L;
        PE.L(i)   = SUPPLY.M(i);
        RE.L(i,t) = max(0,ET.M(i,t));
        DE.L(i)   = S.L(i);
        bottomup_mcp.iterlim = 0;
        solve bottomup_mcp using mcp;
        abort$round(bottomup_mcp.objval,4)
          "MCP symmetric demand model fails to replicate the NLP solution.";
*        -----------------------------------------------------------------
*       Update quantities from the QP solution which are used in the
*       next solution of the top-down model:
        ecost0          = COST.L;
        eref(i)         = S.L(i);
        ESUP.FX(i)      = S.L(i);
        E.L(i,t)        = 0;
        E.L(i,t)        = ET.L(i,t);
);
option bottomup_log:3:2:2;
display bottomup_log;
display precision;
```

Bibliography

Böhringer, Christoph, "The synthesis of bottom-up and top-down in energy policy modeling," *Energy Economics*, 1998, *20-3*, 233–248.

Böhringer, Christoph and Thomas F. Rutherford, "Carbon Abatement and International Spillovers," *Environmental and Resource Economics*, 2002, *22-3*, 391–417.

_ **and** _ , "Integrated Assessment of Energy Policies: Decomposing Top-down and Bottom-up," *Journal of Economic Dynamics and Control*, 2009, *33*, 1648–1661.

Bohringer, Christoph and Thomas Rutherford, "Combining bottom-up and top-down," *Energy Economics*, 2008, *30*, 574–596.

Bosetti, Valentina, Carlo Carraro, Marzio Galeotti, Emanuele Masesetti, and Massimo Tavoni, "Witch: a world induced technical change hybrid model," *The Energy Journal*, 2006, *27*, 13–38.

Dirkse, Steven P. and Michael C. Ferris, "The PATH Solver: a non-monotone stabilization scheme for mixed complementarity problems," *Optimization Methods and Software*, 1995, *5*, 123–156.

Drouet, Laurent, Alain Haurie, Maryse Labriet, Philippe Thalmann, Marc Vielle, and Laurent Viguier, *Chapter 2: A coupled bottom-up/top-down model for GHG abatement scenarios in the Swiss housing sector, Energy and environment*, Cambridge, U.K.: Cambridge University Press, 2005.

Frei, Christoph W., Pierre-Andre Haldi, and Gerard Sarlos, “Dynamic formulation of a top-down and bottom-up merging energy policy model,” *Energy Policy*, 2003, *31*, 1017–1031.

Friedman, Thomas L., *The World is Flat*, New York, NY: Farrar, Straus and Giroux, 2005.

Fujimori, Shinichiro, Toshihiko Masui, and Yuzuru Matsuoka, “Development of a global computable general equilibrium model coupled with detailed energy end-use technology,” *Applied Energy*, 2014, *128*, 296–306.

Greene, William, “A Statistical Model for Credit Scoring,” *Working paper*, 1992.

_ , *Econometrics Analysis*, New York, NY: Prentice Hall, 2011.

Harrison, W.Jill., J.M.Horridge, and K.R.Pearson, “Decomposing Simulation Results with Respect to Exogenous Shocks,” *Computational Economics*, 2000, *15-3*, 227–249.

IPCC, *Climate Change 2001: Mitigation*, Cambridge, U.K.: Cambridge University Press, 2001.

Jacobsen, Henrik K., “Integrating the bottom-up and top-down approach to energy]economy modelling: the case of Denmark,” *Energy Economics*, 1998, *20*, 443–461.

Kalvelagen, Erwin, “A Nonlinear Regression Solver for GAMS,” Working Paper, Amsterdam Optimization CCP 2007.

_ , “Least Squares Calculations with GAMS,” Working Paper, Amsterdam Optimization CCP 2007.

Manne, Alan S., “ETA-MACRO:a model of energy economy interactions,” Technical Report, ElectricPowerResearchInstitute 1977.

_ **, Robert Mendelsohn, and Richard Richels**, “MERGE: a model for evaluating regional and global effects of GHG reduction policies,” *Energy Policy*, 1995, *23*, 17–34.

McFadden, Daniel, "Conditional logit analysis of qualitative choice behavior," in P. Zaremiibka, ed., *Frontiers in Econometrics*, Vol. 4, John Wiley and sons, 1973, pp. 105–135.

Messner, Sabine and Leo Schrattenholzer, "Message-macro: linking an energy supply model with a macroeconomic model and solving iteratively," *Energy*, 2000, *25*, 267–282.

Mizon, Grayham E., "Inferential Procedures in Nonlinear Models: An Application in a UK Industrial Cross Section Study of Factor Substitution and Returns to Scale," *Econometrica*, 1977, *45-5*, 1221–1242.

Nordhaus, William and Joseph Boyer, *Warming the World: Models of Global Warming*, Cambridge, MA: MIT Press, 1999.

Paltsev, Sergey, John M. Reilly, Henry D. Jacoby, Richard S. Eckaus, James McFarland, Marcus Sarofim, Malcolm Asadoorian, and Mustafa Babiker, "The MIT Emissions Prediction and Policy Analysis (EPPA) Model: Version 4," Discussion Paper 125, MIT Joint Program on the Science and Policy of Global Change 2005.

Rutherford, Thomas, "GTAP7inGAMS, The Model," *Working paper*, 2010.

Rutherford, Thomas F., "Extensions of GAMS for complementarity problems arising in applied economics," *Journal of Economic Dynamics and Control*, 1995, *19*, 1299–1324.

_ , "Calibrated CES Utility Functions: A Worked Example," *Working paper*, 2008.

_ **and Huiming Wang**, "Maximum Likelihood and Least Squares Estimation: New Tools for Solving Econometrics Problems with GAMS," *Working paper*, 2014.

Strachan, Neil and Ramachandran Kannan, "Hybrid modelling of long-term carbon reduction scenarios for the UK," *Energy Economics*, 2008, *30*, 2947–2963.

Wang, Huiming, "Local Sensitivity Analysis and Central Path Decomposition," *Working paper*, 2017.

Wing, Ian Sue, "The synthesis of bottom-up and top-down approaches to climate policy modeling: electric power technologies and the cost of limiting US CO_2 emissions," *Energy Policy*, 2006, *34*, 3847–3869.

_ , "The synthesis of bottom-up and top-down approaches to climate policy modeling: electric power technology detail in a social accounting framework," *Energy Economics*, 2008, *30*, 547–573.